ÍNDICE

PRÓLOGO ..... **XI**

PREFACIO ..... **XIII**

CAPÍTULO 1: DISEÑO CONCEPTUAL: MODELO E/R **1**

INTRODUCCIÓN 1
PROBLEMA 1: HABITANTES Y MUNICIPIOS 12
PROBLEMA 2: CURSOS DE FORMACIÓN 17
PROBLEMA 3: CAMPEONATO DE AJEDREZ 23
PROBLEMA 4: ENERGÍA ELÉCTRICA 31
PROBLEMA 5: CONFLICTOS BÉLICOS 40
PROBLEMA 6: GESTIÓN DE NÓMINAS 49
PROBLEMA 7: ENTORNO DE EJECUCIÓN 59
PROBLEMA 8: ORGANIZACIONES NO GUBERNAMENTALES 66
PROBLEMA 9: ADMINISTRACIÓN DE FINCAS 72
PROBLEMA 10: VENTA DE COCINAS 82
PROBLEMA 11: ALQUILER DE VEHÍCULOS 90
PROBLEMA 12: EMPRESA NUMISMÁTICA 96

PROBLEMA 13: PARQUES NATURALES 103
PROBLEMA 14: GESTIÓN DE HOSPITALES 111
PROBLEMA 15: OLIMPIADAS DE INVIERNO 117

CAPÍTULO 2: DISEÑO LÓGICO: MODELO RELACIONAL 125

INTRODUCCIÓN 125

2.1. MODELADO RELACIONAL 143

PROBLEMA 1: ALOJAMIENTOS RURALES 143
PROBLEMA 2: AVES 151
PROBLEMA 3: PINACOTECAS 159
PROBLEMA 4: CASOS DE CORRUPCIÓN 168
PROBLEMA 5: COMISARÍA 179
PROBLEMA 6: CAMINO DE SANTIAGO 190
PROBLEMA 7: PROYECTOS DE INVESTIGACIÓN 201
PROBLEMA 8: NAVES ESPACIALES RELACIONAL 212
PROBLEMA 9: EMPRESA DE TRABAJO TEMPORAL 223
PROBLEMA 10: FONDOS BIBLIOGRÁFICOS 232

2.2. TRANSFORMACIÓN DE ESQUEMAS E/R A ESQUEMAS RELACIONALES 247

PROBLEMA 1: CURSOS DE FORMACIÓN 247
PROBLEMA 2: CAMPEONATO DE AJEDREZ 252
PROBLEMA 3: ENERGÍA ELÉCTRICA 258
PROBLEMA 4: CONFLICTOS BÉLICOS 266
PROBLEMA 5: GESTIÓN DE NÓMINAS 274
PROBLEMA 6: ENTORNO DE EJECUCIÓN 281
PROBLEMA 7: ORGANIZACIONES NO GUBERNAMENTALES 287
PROBLEMA 8: ADMINISTRACIÓN DE FINCAS 292
PROBLEMA 9: GESTIÓN DE HOSPITALES 298
PROBLEMA 10: OLIMPIADAS DE INVIERNO 305

CAPÍTULO 3: NORMALIZACIÓN DE RELACIONES 313

INTRODUCCIÓN 313
PROBLEMA 1 325
PROBLEMA 2 338
PROBLEMA 3 345
PROBLEMA 4 351
PROBLEMA 5 367
PROBLEMA 6 373
PROBLEMA 7 383
PROBLEMA 8 391

PROBLEMA 9 405
PROBLEMA 10 418

CAPÍTULO 4: DISEÑO DE BASES DE DATOS RELACIONALES DISTRIBUIDAS 429

BASES DE DATOS DISTRIBUIDAS 429
PROBLEMA 1: BDD SOCIEDADES MÉDICAS 440
PROBLEMA 2: BDD UNIVERSIDAD CARLOS III 443
PROBLEMA 3: BDD SERVICIOS INFORMÁTICOS 447
PROBLEMA 4: BDD OFICINAS DEL INEM 452

CAPÍTULO 5: DISEÑO CON HERRAMIENTAS CASE 459

INTRODUCCIÓN 459
CASO PRÁCTICO 1: BASURA ESPACIAL 460
CASO PRÁCTICO 2: SEGUROS DE AUTOMÓVIL 469

APÉNDICE: LISTA DE ACRÓNIMOS Y ABREVIATURAS 483

BIBLIOGRAFÍA 485

ÍNDICE ALFABÉTICO 487

PRÓLOGO

Diseño de Bases de Datos: Problemas Resueltos se presenta en el panorama editorial como un instrumento de valiosa ayuda tanto para los estudiantes de Informática como para los profesionales del desarrollo de software que pretendan dominar las diversas técnicas requeridas para el Diseño de Bases de Datos. Sus más de 50 problemas con su respectiva solución comentada lo convierten en un auténtico manual de auto-aprendizaje de esta disciplina y que complementan a otras obras dedicadas a los aspectos más teóricos del diseño de Bases de Datos en mayor medida. En definitiva, es éste un libro para trabajar con él en el que se esclarece la praxis, de modo que el lector puede ir adquiriendo un mayor grado de dominio en cada una de sus técnicas de una forma amena y sencilla.

Por otra parte, es muy grato para mí tener la oportunidad de prologar este libro realizado por un grupo de discípulos de mi gran amiga, la profesora doña Adoración de Miguel, que tan inesperadamente nos dejó. Con esta obra queda patente que su equipo sigue vivo, continuando su dedicación y esfuerzo y en la dirección que ella les marcó.

Asimismo, me siento orgulloso de mi antigua alumna de la Facultad de Informática de la Universidad Politécnica de Madrid y ahora compañera, Paloma Martínez, por su capacidad para abordar la coordinación de esta obra compaginándola

con las nuevas responsabilidades del área de Lenguajes y Sistemas Informáticos del Departamento de Informática de la Universidad Carlos III de Madrid, así como otro tipo de responsabilidades recientemente adquiridas que se desprenden de su nueva condición de madre. Enhorabuena.

Por último, no quisiera finalizar estas líneas sin hacer una pequeña mención al profesor don Isidro Ramos, quien desde el Mediterráneo siempre ha tendido la mano generosa, confiando y apoyando a este grupo que se siente en deuda con él.

Antonio de Amescua

Madrid, 20 de septiembre de 2000

PREFACIO

La obra se centra en el diseño de bases de datos desde un punto de vista eminentemente práctico. Nuestra experiencia docente en la universidad ha puesto de manifiesto la dificultad a la que se enfrentan los alumnos cuando realizan tareas de análisis y diseño de bases de datos. Por ello, los objetivos que nos hemos propuesto al escribir este libro son:

- Proporcionar una colección de problemas que sirva al lector para poner en práctica los conceptos teóricos de análisis, diseño e implementación de bases de datos.

- Abordar la resolución de los problemas de una forma comprensible justificando en todo momento las decisiones de diseño.

- Emplear unos principios metodológicos que ayuden a los analistas y diseñadores de bases de datos a elaborar esquemas conceptuales y lógicos intentando perder la mínima semántica.

- Dar a conocer el soporte que pueden ofrecer las herramientas CASE en el desarrollo de bases de datos.

Algunos de los ejercicios propuestos provienen de prácticas y exámenes propuestos en nuestras asignaturas de Bases de Datos de las titulaciones de Ingeniería Técnica en Informática de Gestión e Ingeniería Informática de la Universidad Carlos III de Madrid, en la que los autores son profesores. La idea fue elaborar un texto que fuera el complemento práctico a los libros *Diseño de Bases de Datos Relacionales* de Adoración de Miguel, Mario Piattini y Esperanza Marcos y *Fundamentos y Modelos de Bases de Datos* de Adoración de Miguel y Mario Piattini, ambos publicados en esta misma editorial.

CONTENIDOS

El libro está estructurado en cinco grandes bloques: Diseño conceptual, Diseño Lógico, Teoría de la Normalización, Bases de Datos distribuidas y Casos Prácticos completos desarrollados con herramientas CASE. La razón de esta división viene dada porque creemos que esta estructura es la que mejor se ajusta a las fases metodológicas y aspectos cruciales del Diseño de Bases de Datos.

El capítulo de Diseño Conceptual contiene una parte introductoria en la que se recuerdan brevemente los conceptos teóricos y notación del modelo E/R, así como algunas guías metodológicas para la elaboración de las soluciones a los casos propuestos. El resto de este capítulo lo forman 15 ejercicios desglosados en varios pasos; en cada uno de los cuales se estudiará un conjunto de supuestos semánticos que darán lugar a un subesquema E/R de forma que se irán añadiendo elementos al subesquema obtenido en el paso anterior y así sucesivamente hasta completar el estudio de todos los supuestos semánticos contemplados en el enunciado del problema. Supondremos que el enunciado constituye una descripción correcta (y casi siempre completa) del Universo del Discurso.

Dos de los ejercicios de este capítulo, exactamente los problemas 2 y 14, ya fueron propuestos en los apéndices del libro *Diseño de Bases de Datos Relacionales*, pero no se desarrollaron las soluciones.

El capítulo de Diseño Lógico contiene una introducción a modo de recordatorio del modelo relacional y su notación, reglas de transformación de un esquema E/R a un esquema relacional y álgebra relacional. A continuación, existen dos bloques de problemas. El primero engloba 10 ejercicios prácticos de modelado relacional en los que se obtiene el esquema lógico estándar de la BD (grafo relacional) a partir de un conjunto de supuestos semánticos que describen el problema junto con el script de creación de la BD en SQL-92. Además, se muestran varias consultas en SQL-92 y álgebra relacional. El segundo bloque lo forman 10 problemas (cuyos enunciados se resolvieron en el capítulo de Diseño Conceptual) en los que se obtiene un esquema relacional a partir de un esquema E/R.

El capítulo dedicado a la Normalización de relaciones se estructura en una introducción teórica en la que se muestran muy brevemente aspectos del cálculo de las

dependencias, cálculo de formas normales y algoritmos de análisis y síntesis, y en un conjunto de 10 ejercicios. Los ejercicios 3, 4, 5 y 6 también fueron propuestos en el libro *Diseño de Bases de Datos Relacionales* y aunque se indicaban el resultado final de la solución no se desarrollaron por completo. Por ello, aprovechamos aquí para presentar las soluciones completas.

El capítulo dedicado a las Bases de Datos distribuidas contiene una introducción en la que se describen las características y arquitecturas de las Bases de Datos distribuidas así como los algoritmos de diseño de este tipo de Bases de Datos. Se muestran 4 ejercicios de diseño en los que se construyen los esquemas de fragmentación y asignación a partir de las especificaciones del problema

Finalmente, el último capítulo incluye dos casos prácticos completos resueltos con dos herramientas CASE comerciales. Para el primero de ellos se ha utilizado la herramienta CASE ERwin y para el segundo la herramienta Designer 2000 de Oracle. En ambos casos se realiza el desarrollo completo de una base de datos desde la fase de análisis hasta la implementación en un Sistema de Gestión de Bases de Datos comercial. El objetivo de estos casos prácticos es mostrar la potencialidad de las herramientas CASE para ayudar en el desarrollo de Bases de Datos.

ORIENTACIÓN A LOS LECTORES

La audiencia a la que va dirigida el libro es muy amplia: alumnos de las asignaturas de Diseño de Bases de Datos en facultades, escuelas universitarias y escuelas de formación profesional, así como profesionales informáticos que trabajen en el área de bases de datos (analistas, programadores, etc.).

OTRAS OBRAS RELACIONADAS

De Miguel, A. y Piattini, M. (1999). *Fundamentos y modelos de bases de datos*. 2ª ed. Ed. Ra-Ma, Madrid.

> *Este libro constituye el complemento teórico al capítulo 2 de la presente obra, ya que presenta los conceptos fundamentales de la tecnología de bases de datos, analizando en profundidad el modelo relacional y el lenguaje SQL.*

De Miguel, A., Piattini, M. y Marcos, E. (1999). *Diseño de bases de datos relacionales*. Ed. Ra-Ma, Madrid.

> *Este libro constituye el complemento teórico a los capítulos 1, 3, 4 y 5 de la presente obra, ya que presenta con detalle el concepto de modelo de datos, una metodología de desarrollo de bases de datos, el modelado conceptual utilizando el modelo E/R, las reglas de transformación de esquemas conceptuales a esquemas relacionales, la*

teoría de la normalización y herramientas CASE para el desarrollo de bases de datos.

TESTIMONIO DE RECONOCIMIENTO

Este libro se comenzó unos meses antes de que falleciera Adoración de Miguel, catedrática y responsable del grupo de Bases de Datos Avanzadas del Departamento de Informática de la Universidad Carlos III de Madrid. Durante todos los años que hemos trabajado juntos, Adoración no sólo nos ha transmitido conocimientos teóricos, inquietudes científicas e interés por la enseñanza, sino también unos valores humanos de gran significación para nosotros. Su gran perseverancia tanto en la labor docente como en la investigadora y la coherencia demostrada a lo largo de su actividad profesional han sido para todos nosotros un estímulo que esperamos no desaparezca nunca de nuestras vidas.

También deseamos agradecer a Antonio de Amescua el apoyo recibido para llevar a término este trabajo así como a nuestros compañeros del Departamento de Informática de la Universidad Carlos III, y a Juan de Dios García y Carmen Sánchez por su colaboración en la preparación inicial del libro.

Parte del material de esta obra se ha desarrollado en el marco del proyecto TIC1999-0215 (PANDORA: Plataforma CASE para el Aprendizaje y Desarrollo de Bases de Datos y su Enseñanza vía Internet) subvencionado por la CICYT. Por ello, deseamos mostrar nuestro agradecimiento a la CICYT por la subvención concedida.

Deseamos expresar nuestro agradecimiento a nuestras familias y amigos que han soportado pacientemente nuestras horas extras de trabajo y que nos han ofrecido en todo momento su apoyo incondicional.

Por último, nos resta agradecer a la editorial Ra-Ma, en especial a José Luis Ramírez, su inestimable ayuda.

Los autores

Madrid, septiembre de 2000

CAPÍTULO 1

DISEÑO CONCEPTUAL: MODELO E/R

INTRODUCCIÓN

Guías metodológicas para abordar la resolución de los problemas

La elaboración de un esquema E/R que recoja la semántica de un determinado Universo del Discurso es un proceso creativo para el que no existe un procedimiento definido. Sin embargo, sí es posible seguir una serie de recomendaciones o heurísticas que nos ayuden en el diseño. Estas recomendaciones no son reglas que siempre funcionen sino que en algunos casos son adecuadas y en otros no.

A. ALGUNAS HEURÍSTICAS PARA LA IDENTIFICACIÓN DE LOS DISTINTOS CONSTRUCTORES

En la primera propuesta del Modelo E/R, Chen (1976), se distinguen tan sólo tres clases de objetos: entidades, atributos e interrelaciones. El modelo que se utilizará en la fase de modelado conceptual en los ejercicios de este libro es el modelo E/R extendido que añade al modelo básico un conjunto de constructores que ayudan a recoger mayor semántica del universo del discurso (cardinalidades mínimas y máximas en las interrelaciones, generalizaciones, dependencia en existencia y en

identificación, etc.). Realmente no se trata de un único modelo sino de una familia de modelos, dado que cada autor lo especializa y le añade distintos constructores. La descripción completa de este modelo se encuentra en De Miguel *et al.* (1999). En este apartado se expondrán brevemente cuáles son los constructores de los que consta el modelo E/R extendido con el fin de facilitar la comprensión de los problemas.

Las **entidades** son los objetos principales sobre los que debe recogerse información y generalmente denotan personas, lugares, cosas o eventos de interés. Las entidades aparecerán reflejadas en el enunciado habitualmente como nombres. A cada una de las posibles ocurrencias (cada persona, lugar, cosa o evento concreto) de la entidad se le denomina *ejemplar*.

Los **atributos** se utilizan para detallar las entidades asignándoles propiedades descriptivas tales como nombre, color y peso. Existen dos tipos de atributos: *identificadores* y *descriptores*. Los primeros se utilizan para distinguir de manera única cada una de las ocurrencias de una entidad (distinguiéndose entre *Identificadores Principales* e *Identificadores Alternativos*), mientras que los descriptores se utilizan para describir una ocurrencia de entidad. No sólo es posible especificar atributos en la entidades sino también en las interrelaciones (en este caso sólo tiene sentido hablar de atributos descriptores y no de identificadores). Los atributos también aparecerán reflejados en el enunciado, generalmente, como nombres.

En relación con los atributos también existe el concepto de *dominio* (conjunto de valores sobre los que se define el atributo). Aunque se pueden representar explícitamente en los esquemas E/R (como se muestra en el siguiente apartado donde se reflejan las notaciones) en los problemas de este libro se considerará que el dominio toma el mismo nombre que el del atributo. Por ejemplo, la entidad *EMPLEADO* puede tener el atributo "Estado Civil" definido sobre el dominio "Estados Civiles" (compuesto por los valores Soltero, Casado, Viudo, Divorciado) y cuyas dos posibles representaciones en el esquema E/R se muestran en la figura 1.1.

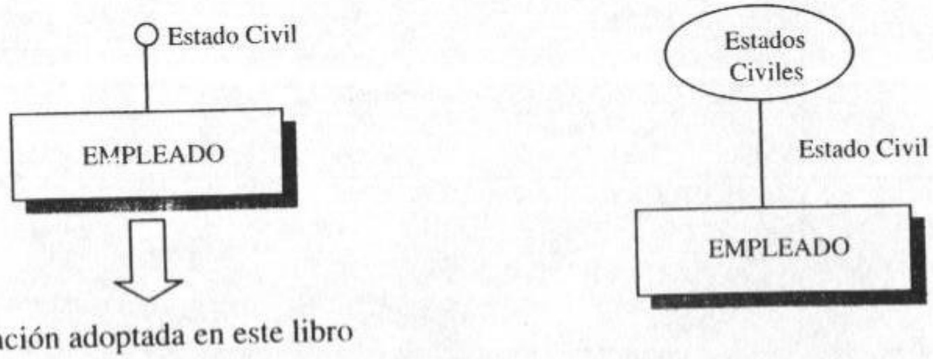


Figura 1.1. Representación de dominios

También es posible recoger otras restricciones semánticas sobre los atributos aparte de las ya mencionadas de atributos Identificadores Principales y Alternativos. Así, hablamos de atributos *obligatorios/opcionales* (si un atributo debe tomar o no un valor), atributos *univaluados/multivaluados* (si un atributo toma un único valor o varios), atributos *derivados* (si su valor se obtiene a partir de otros elementos del esquema E/R), atributos *compuestos/simples* (dependiendo de si un atributo es o no un agregado de otros atributos). A su vez, estas restricciones se pueden combinar entre sí (pueden existir en un esquema E/R atributos multivaluados simples opcionales, univaluados compuestos opcionales, multivaluados obligatorios, multivaluados compuestos, etc.)

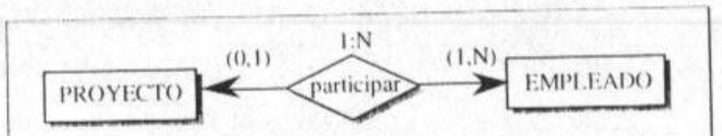
Las entidades pueden clasificarse por la fuerza de sus atributos identificadores, es decir, por su dependencia o no dependencia respecto a otras entidades. Las entidades *fuertes* tienen existencia propia, es decir, poseen identificadores internos que determinan de manera única la existencia de sus ocurrencias. Las entidades *débiles* pueden serlo por dos motivos: bien porque su existencia en la BD depende de una entidad fuerte, bien porque requieran para su identificación de los atributos identificadores (algunas veces llamados atributos externos) de otra entidad, por ejemplo, no poseen atributos identificadores internos que permitan la identificación de cada una de sus ocurrencias y requieren la presencia de atributos externos. En el primero de los casos se habla de *Dependencia en Existencia* y en el segundo *de Dependencia en Identificación*[1].

Finalmente, las **interrelaciones** representan asociaciones del mundo real entre una o más entidades. Las interrelaciones se caracterizan por su nombre, el grado (número de entidades que participan en la interrelación), el tipo de correspondencia (número máximo de ejemplares de una entidad asociados a una combinación de ejemplares de las otras entidades en la interrelación, que puede ser 1 ó N). Así, en el ejemplo de la figura 1.2 se observa que el tipo de correspondencia de la interrelación **Participar** es 1:N, es decir, un empleado participa como máximo en un proyecto y en un proyecto participan como máximo N empleados. Al igual que en las entidades, se denomina ejemplar de la interrelación a cada combinación de ejemplares de las entidades interrelacionadas que constituyen una ocurrencia en la interrelación.

Un constructor que amplía la semántica recogida en una interrelación es la *restricción de cardinalidad*. Se definen las cardinalidades máximas y mínimas de las entidades que participan en una interrelación como el número máximo y mínimo de ejemplares de una entidad que puede relacionarse con un único ejemplar de la otra, u otras entidades que participan en la interrelación, De Miguel *et al.* (1999). Gráficamente, las restricciones de cardinalidad se representan por una etiqueta, (0,1),

[1] Que, obviamente, siempre implica una Dependencia en Existencia.

(1,1), (0,N) o (1,N)[2], situada en la línea que conecta la entidad con el rombo que representa el tipo de interrelación (ver figura 1.2).



(1,N) en *EMPLEADO* indica que un ejemplar de la entidad *PROYECTO* está relacionado con uno o más ejemplares de la entidad *EMPLEADO* en la interrelación *participar*.

(0,1) en *PROYECTO* indica que un ejemplar de la entidad *EMPLEADO* está relacionado con cero o un ejemplar de la entidad *PROYECTO* en la interrelación *participar*.

Figura 1.2. Ejemplo de cardinalidades mínimas y máximas

En los enunciados, la aparición de verbos podrá indicarnos, en algunos casos, la existencia de una interrelación en el esquema E/R.

En cuanto a las **generalizaciones,** nos proporcionan un mecanismo de abstracción que permite especializar una entidad (que se denominará *supertipo*) en *subtipos*, o lo que es igual, generalizar los subtipos en el supertipo. De esta forma vemos un conjunto de ocurrencias de una entidad como ocurrencias de otra entidad (como sucede también en las jerarquías "es_un" de las redes semánticas). Por ejemplo, una "Persona" es un "Animal" y un "Reptil" es un "Animal"; en este caso, "Animal" puede considerarse el supertipo y "Persona" y "Reptil" son subtipos de "Animal". Las ocurrencias o ejemplares de "Persona" lo son también de "Animal" e igual sucede con las de "Reptil".

Podremos identificar generalizaciones si encontramos una serie de atributos comunes a un conjunto de entidades. Esta situación se da con frecuencia si se está realizando un proceso de integración de vistas, De Miguel *et al.* (1999). Estos atributos comunes describirán al supertipo y los atributos particulares permanecerán en los subtipos. Puede ocurrir que los subtipos no tengan atributos propios; en ese caso, sólo existirán subtipos (aun cuando por recoger mejor la semántica, éstos puedan reflejarse) si éstos van a participar en interrelaciones (aparte de las interrelaciones en las que participe el supertipo).

Por ejemplo, supongamos las siguientes entidades identificadas en un universo del discurso de una empresa: *EMPLEADO* (con identificador "Nº-Emp" y los descriptores "Nombre-Emp", "Dirección-Familiar", "Fecha-Nacimiento", "Descripción-Puesto", "Salario", "Experiencia"), *INGENIERO* (con identificador "Nº-Emp" y descriptores "Nombre-Emp", "Dirección-Familiar", "Especialidad"), *SECRETARIO*

[2] Otras posibles restricciones de cardinalidad son (0, Num_1), (1, Num_1), (Num_1, N), (Num_1, Num_2)

(con idenficador "Nº-Emp" y descriptores "Nombre-Emp", "Fecha-Nacimiento", "Salario", "Pulsaciones"), *TÉCNICO* (con identificador "Nº-Emp" y descriptores "Nombre-Emp", "Experiencia", "Años-de-Experiencia").

Identificamos que *EMPLEADO* es una generalización de *INGENIERO*, *SECRETARIO* y *TÉCNICO*. Entonces redistribuimos los atributos entre las entidades. Situamos el identificador "Nº-Emp" y los descriptores genéricos "Nombre-Emp", "Dirección-Familiar", "Fecha-Nacimiento", "Descripción-Puesto" y "Salario" en la entidad supertipo *EMPLEADO*, ponemos el descriptor específico "Especialidad" en la entidad *INGENIERO*; ponemos el descriptor específico "Pulsaciones" en la entidad *SECRETARIO* y, por último, ponemos los descriptores específicos "Experiencia" y "Años-Experiencia" en la entidad *TÉCNICO*. La figura 1.3 muestra el esquema E/R resultante.

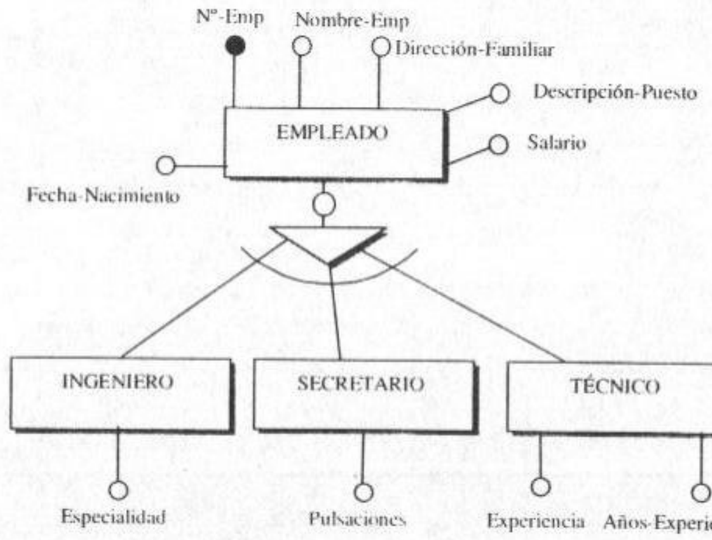


Figura 1.3. Ejemplo de generalización

Existen otras restricciones semánticas relacionadas con las generalizaciones como son la *totalidad/parcialidad* y la *exclusividad/solapamiento*. Si las ocurrencias de los subtipos de una generalización cubren al supertipo (es decir, no hay ocurrencias en el supertipo que no pertenezcan a ninguno de los subtipos) entonces se trata de una generalización total y en caso contrario parcial. Por otro lado, si puede haber ocurrencias que pertenezcan a más de uno de los subtipos entonces se trata de generalizaciones con solapamiento; en caso de que los subtipos sean disjuntos se habla de generalizaciones exclusivas.

En el ejemplo de la Figura 1.3 se muestra una generalización total, es decir, no existen *EMPLEADOS* que no sean ni *INGENIEROS*, ni *SECRETARIOS* ni *TÉCNICOS*; además, es una generalización exclusiva, es decir, un *EMPLEADO* o es *INGENIERO* o *SECRETARIO* o *TÉCNICO*, pero no puede ser varias cosas a la vez.

B. ALGUNAS HEURÍSTICAS PARA LA ELECCIÓN ENTRE VARIOS CONSTRUCTORES

Aunque es sencillo definir los constructores de entidad, atributo e interrelación, no es tan sencillo distinguir su papel en el modelado de BD. ¿Qué es lo que hace que un concepto sea un atributo, una entidad o una interrelación?; en este caso únicamente se podrá recurrir también a una serie de heurísticas que nos ayuden a decidir entre uno de los constructores que pueden reflejar la semántica de un determinado concepto. Algunas de ellas se describen a continuación.

Entidades vs. atributos

Los atributos no tienen existencia por sí mismos sino que tienen sentido en cuanto a que pertenecen a una determinada entidad o interrelación. Una entidad debe estar caracterizada por algo más que su Identificador Principal. Si existe información descriptiva sobre un concepto u objeto, entonces debería clasificarse como entidad. Si sólo se necesita un identificador para un objeto, el objeto debería clasificarse como un atributo. Así,

las entidades poseen información descriptiva y los atributos no

Por ejemplo, en el supuesto "los Almacenes se localizan en Ciudades". Si existe alguna información descriptiva sobre el Estado y la Población de las Ciudades, entonces Ciudad debería clasificarse como entidad. Si sólo se necesita el atributo "Nombre Ciudad" para identificar una ciudad, entonces debería clasificarse como atributo.

Por otro lado, podría ocurrir que aún teniendo un concepto para el que sólo existe un Identificador Principal, éste se relacione con más de una entidad. En ese caso podría aparecer como una entidad en el esquema E/R. En el ejemplo anterior, si existiera otro supuesto "los proveedores tienen asignados varias ciudades", entonces podría dejarse Ciudad como entidad relacionada con las entidades Almacén y Proveedor o bien dejarse como atributos en ambas entidades.

Entidades vs. atributos multivaluados

En este aspecto existen discrepancias. Hay propuestas que prefieren incorporar en los esquemas E/R un atributo multivaluado como una entidad y otras prefieren representarlo como un atributo. En nuestro caso, con independencia de que el atributo sea simple o compuesto, si se sabe que tendrá un número limitado y no muy elevado de ocurrencias, entonces formará parte de la entidad que describe (siempre y cuando el concepto que representa no esté relacionado con otras entidades del esquema E/R). Un ejemplo puede ser "De un empleado interesa almacenar su DNI, nombre, dirección y teléfonos"; en este caso el atributo "Teléfono" es un atributo multivaluado de la

entidad *EMPLEADO*. En el ejemplo "Un profesor se caracteriza por su nombre, DNI, dirección y los campus en los que imparte docencia"; en este caso el atributo "Campus" también podría ser un atributo multivaluado de la entidad *PROFESOR* pero si existen supuestos adicionales en el universo del discurso que nos indican información adicional para describir un campus y además se relaciona con otras entidades (por ejemplo, *DEPARTAMENTO*, *AULARIO*, etc.) entonces debe reflejarse como una entidad.

Entidades vs. interrelaciones

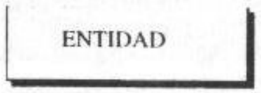
Lo habitual es no tener problemas en la diferenciación entre entidades e interrelaciones. Las interrelaciones asocian una o varias entidades, mientras que las entidades no. Sin embargo, en cualquier interrelación puede realizarse un proceso de nominalización. Por ejemplo, mediante una nominalización puede convertirse "los hombres se casan con mujeres" en "los hombres y mujeres forman matrimonios". Así, se ha sustantivado una interrelación y al introducir un nuevo concepto, se ha convertido en una entidad.

Aun cuando la nominalización puede resultar útil en un proceso de diseño complejo, en especial para tratar de reducir el grado de una interrelación muy compleja o para encontrar elementos de interés para el sistema que inicialmente no se habían tenido en cuenta, en nuestro caso, evitaremos las nominalizaciones que no se encuentran presentes en los enunciados o que no resultan evidentes en el Universo del Discurso.

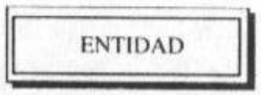
Notaciones

A continuación se muestran las convenciones seguidas en los problemas para la representación gráfica de los distintos constructores de un diagrama E/R.

- Entidad (fuerte)



- Entidad débil



- Identificador Principal (IP)[3]

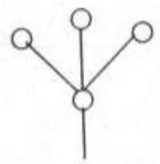
[3] En el Modelo Relacional, Clave Primaria

- Identificador Alternativo (IA)[4]

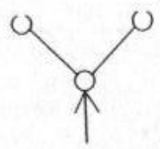
- Atributo

- Atributo Multivaluado

- Atributo Compuesto



- Atributo Multivaluado Compuesto



- Atributo Opcional

- Atributo Derivado[5]

[4] En el Modelo Relacional, Clave Alternativa

[5] En el Modelo Relacional, atributo procedimiento de derivación

- Interrelación

- Interrelación con Dependencia en Existencia

- Interrelación con dependencia en Identificación

- Dominio

- Jerarquía solapada y parcial (sin ninguna restricción)

- Jerarquía solapada y total

- Jerarquía exclusiva y parcial

- Jerarquía exclusiva y total

Cómo se estructuran los problemas

A fin de facilitar la comprensión de las soluciones propuestas a los problemas de modelado, cada problema se desglosará arbitrariamente en varios pasos. En cada paso se estudiará un conjunto de supuestos semánticos que darán lugar a un subesquema E/R. En cada paso se irán añadiendo elementos al subesquema obtenido en el paso anterior y así sucesivamente hasta completar el estudio de todos los supuestos semánticos contemplados en el enunciado del problema. Supondremos que el enunciado constituye una descripción correcta (y casi siempre completa) del Universo del Discurso.

Una aproximación utilizada habitualmente en la construcción de esquemas E/R es identificar primero las entidades, luego las interrelaciones y por último los atributos de las entidades e interrelaciones.

En nuestro caso se identificarán todos los elementos (entidades, atributos, interrelaciones, cardinalidades, etc.) por cada conjunto de supuestos semánticos analizados. Ello puede implicar que algunos conceptos se representen en los primeros pasos con determinados constructores y que posteriormente, en sucesivos pasos, los nuevos supuestos semánticos analizados nos proporcionen información adicional que modifique alguno de los constructores seleccionados. Por ejemplo, podría ocurrir que un determinado concepto se representara en una primera aproximación como un atributo y que según se avanza en el análisis de los supuestos del enunciado se descubra que debe representarse como una entidad.

A veces también se suele recurrir a otro tipo de herramientas que nos ayudan a detectar información que no aparece explícitamente representada en el enunciado y que resultan de gran utilidad a los diseñadores inexpertos. Así, una propuesta de

metodología de realización de un esquema conceptual que tiene en cuenta estos aspectos constaría de los siguientes pasos[6]:

1. Estudiar el enunciado que describe el Universo del Discurso y elaborar dos listas; una con los candidatos a ser entidades y otra con las posibles interrelaciones junto con su tipo de correspondencia (1:1, 1:N, N:M). Además, se especificarán aquellos conceptos dudosos que no se sabe cómo representar (si como entidad o como interrelación).

2. Construir una Matriz de Entidades en la que las filas y las columnas son nombres de entidades y cada celda puede contener o no nombres de interrelaciones. Esta matriz tiene el siguiente aspecto:

	E1	**E2**	**E3**	**...**	**EN**
E1	I1	I2	--	...	I3
E2	X	I4	I5	...	--
E3	X	X	I6	...	--
...	X	X	X	...	IN
EN	X	X	X	X	X

Las entidades son *E1, E2,..., EN* y las interrelaciones son **I1**, **I2**, ..., **IN**. Como la matriz es simétrica, las celdas que aparecen con una cruz se corresponden con interrelaciones que ya están especificadas en la otra mitad de la matriz. El símbolo -- en una celda indica que no existe interrelación entre las dos entidades referenciadas. Además, habría que indicar los tipos de correspondencia de cada interrelación. Por ejemplo, **I1** podría ser una interrelación 1:N. Es importante destacar que esta matriz no recoge las interrelaciones de grado superior a dos.

En la elaboración de esta matriz es posible detectar interrelaciones que no aparecen explícitamente representadas en el enunciado y que, sin embargo, podría resultar interesante que se recogieran en el esquema E/R. Este tipo de interrelaciones se detectan, generalmente, por sentido común, aunque sería necesario siempre validarlas con el usuario.

3. Utilizando la matriz de entidades se construye un primer esquema E/R con las entidades, atributos, interrelaciones y sus tipos de correspondencia. A este esquema se le añaden las cardinalidades mínimas y máximas.

4. En este último paso se refina el esquema E/R del paso anterior estudiando las posibles redundancias siempre y cuando existan ciclos con interrelaciones

[6] Esta aproximación se utilizará en el primer ejercicio de este tema.

semánticamente equivalentes. Existe redundancia en un esquema E/R cuando la misma semántica se recoge de manera duplicada, por lo que ese esquema podría representarse manteniendo la misma semántica con menos elementos. En general, puede haber redundancia cuando existen ciclos en un esquema E/R (varias entidades unidas por varias interrelaciones relacionadas semánticamente formando un ciclo). En este caso, habría que comprobar si eliminando una interrelación, la semántica representada en ella puede obtenerse mediante las interrelaciones restantes. Para ello, hay que estudiar detalladamente las cardinalidades de las interrelaciones y hacer la comprobación tanto en un sentido como en el otro. Este proceso se describe en De Miguel *et al.* (1999). Se mostrará algún ejemplo en los ejercicios de este capítulo.

PROBLEMA 1: HABITANTES Y MUNICIPIOS

Enunciado

Supongamos el siguiente universo del discurso sobre municipios, viviendas y personas. Cada persona sólo puede habitar en una vivienda y estar empadronada en un municipio, pero puede ser propietaria de varias viviendas. Nos interesa también conocer las personas que dependen del Cabeza de Familia (C.F.). Se indicarán los supuestos semánticos que se consideren oportunos para justificar todas las decisiones de diseño.

Discusión del enunciado

Para la resolución de este ejercicio se seguirán los pasos indicados al final de la sección 3 de la introducción, en la que se exponía una guía metodológica para diseñadores inexpertos.

- **1er paso**: Elaborar las listas de conceptos candidatos a ser entidades e interrelaciones e indicar también los conceptos que no se sabe cómo catalogar. Las listas obtenidas son:

Entidades:	Interrelaciones:
MUNICIPIO	**Habita** entre *PERSONA* y *VIVIENDA*
VIVIENDA	**Empadronada** entre *PERSONA* y *MUNICIPIO*
PERSONA	**Propiedad** entre *PERSONA* y *VIVIENDA*

¿CABEZA DE FAMILIA?

Las entidades e interrelaciones anteriores están explícitamente representadas en el enunciado. En principio, no sabemos cómo representar el concepto Cabeza de Familia, pues en realidad es también una Persona. Dejaremos la clasificación de este concepto para el siguiente paso.

- **2º paso**: Construir una matriz Entidades/Entidades para representar todas las interrelaciones junto con su tipo de correspondencia. Para ello, iremos analizando los supuestos semánticos explícitamente representados en el enunciado, así como los que están implícitos o son de sentido común.

A) Supuestos dados en el enunciado:

- Cada PERSONA sólo puede HABITAR en una VIVIENDA

  (interrelación **Habita** (1:?) entre *PERSONA* y *VIVIENDA*)

- Cada PERSONA puede ser PROPIETARIA de más de una VIVIENDA

  (interrelación **Propiedad** (?:N) entre *PERSONA* y *VIVIENDA*)

- Las PERSONAS dependen del cabeza de familia

  (interrelación **C.F.** (?:?) entre *PERSONA* Y *PERSONA*)

- Una PERSONA está empadronada en un único MUNICIPIO.

  interrelación **Empadronada** (1:N) entre *PERSONA* y *MUNICIPIO*

B) Supuestos no dados en el enunciado:

- En una VIVIENDA pueden HABITAR muchas PERSONAS (supuesto lógico del mundo real)

  interrelación **Habita** (1:N) entre *PERSONA* y *VIVIENDA*

- Una VIVIENDA puede ser PROPIEDAD de muchas PERSONAS (supuesto legal)

  interrelación **Propiedad** (M:N) entre *PERSONA* y *VIVIENDA*

- Una PERSONA sólo puede tener un cabeza de familia y un cabeza de familia puede serlo de varias PERSONAS

  interrelación **C.F.** (1:N) entre *PERSONA* y *PERSONA*

- Un MUNICIPIO puede tener muchas VIVIENDAS y una VIVIENDA pertenece a un solo MUNICIPIO.

  interrelación **Está_En** (N:1) entre *MUNICIPIO* y *VIVIENDA*

La matriz obtenida se muestra en la figura 1.4.

	PERSONA	*MUNICIPIO*	*VIVIENDA*
PERSONA	**C.F.** (1:N)	**Empadronada** (1:N)	**Habita** (1:N) **Propiedad** (N:M)
MUNICIPIO		--	**Está_En** (N:1)
VIVIENDA			--

Figura 1.4. Matriz de entidades/entidades

- **3er paso**: Obtener una versión preliminar del esquema E/R. En la figura 1.5 se muestra una primera versión del Esquema E/R correspondiente a los supuestos mencionados.

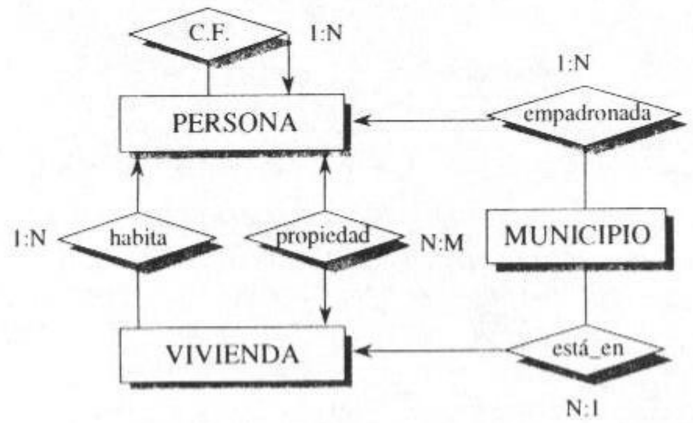


Figura 1.5. Versión preliminar del esquema E/R

- **4º paso**: Análisis de las cardinalidades mínimas. Hasta ahora se han estudiado sólo las cardinalidades máximas de las interrelaciones. A continuación, se estudiarán las cardinalidades mínimas.

  - Interrelación **C.F.**: Una *PERSONA* tiene obligatoriamente como mínimo una *PERSONA* que es Cabeza de Familia y una *PERSONA* que es Cabeza de Familia puede que no tenga ninguna persona a su cargo.

  - Interrelación **Habita**: Una *PERSONA* habita como mínimo en una *VIVIENDA* y en una *VIVIENDA* puede que no habite ninguna *PERSONA*.

  - Interrelación **Propiedad**: Una *PERSONA* puede que no sea propietaria de ninguna *VIVIENDA* y una *VIVIENDA* puede que no sea propiedad de ninguna *PERSONA* (una vivienda podría ser propiedad de una empresa, por ejemplo).

  - Interrelación **Empadronada**: Una PERSONA está empadronada como mínimo en un MUNICIPIO (y como máximo también) y en un MUNICIPIO como mínimo está empadronada una PERSONA.

  - Interrelación **Está_En**: Una *VIVIENDA* está en un único *MUNICIPIO* y en un *MUNICIPIO* hay, como mínimo, una *VIVIENDA*

La figura 1.6 muestra el diagrama E/R con las cardinalidades máximas y mínimas.

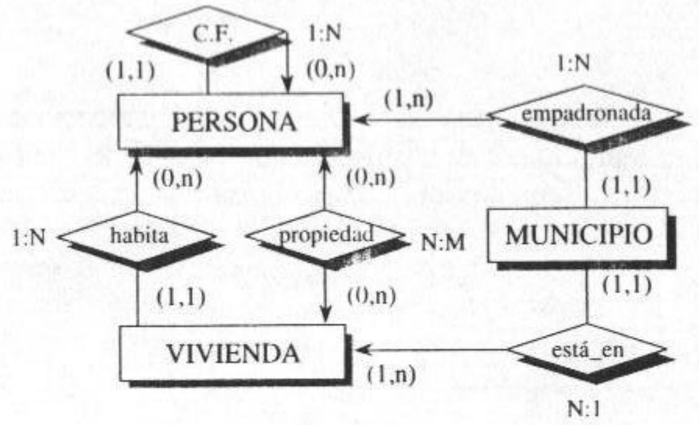


Figura 1.6. Esquema E/R con restricciones de cardinalidad

- **5º paso**: Análisis de redundancias. Como existen dos ciclos en el esquema E/R hay que estudiar si existe alguna interrelación redundante, es decir, si hay alguna interrelación cuya semántica pueda obtenerse a partir de las otras interrelaciones.

  El primer ciclo lo constituyen las interrelaciones **Propiedad**, **Está_En** y **Empadronada**. La primera condición para saber si tenemos un ciclo en el que haya alguna interrelación susceptible de ser redundante es que las tres interrelaciones estén semánticamente relacionas. En este caso la interrelación **Propiedad** no es semánticamente equivalente a **Está_En** y **Empadronada**, puesto que el poseer o no una vivienda no influye en si la persona reside en el municipio en el que se encuentra la vivienda.

  El segundo ciclo lo constituyen las interrelaciones **Habita**, **Está_En** y **Empadronada**. En este caso las tres interrelaciones están semánticamente relacionadas[7]. Veamos si alguna de estas interrelaciones es redundante:

  Interrelación **Habita**: Si intentamos eliminar la interrelación **Habita** debe ser posible obtener su semántica a partir de **Está_En** y **Empadronada**; así, si queremos obtener las personas que habitan en una determinada vivienda, a partir de **Está_En** se obtiene el municipio en el que se encuentra la vivienda y con la interrelación **Empadronada** se obtienen las personas que habitan en ese municipio, pero no sabemos las personas que habitan en la vivienda sino las que habitan en todas las viviendas del municipio. Por ello, la interrelación **Habita** no se puede eliminar.

[7] Suponemos que las personas habitan en los municipios en los que están empadronadas.

Interrelación **Está_En**: Si intentamos suprimir la interrelación **Está_En** debe ser posible obtener su semántica a partir de las interrelaciones **Habita** y **Empadronada**. Para conocer las viviendas que se encuentran en un determinado municipio, a partir de **Empadronada** obtenemos todas las personas empadronadas en ese municipio y mediante la interrelación **Habita** obtenemos las viviendas en las que habitan esas personas (pues una persona obligatoriamente debe habitar en una vivienda); de esta forma, sabremos las viviendas de ese municipio. En el otro sentido de la interrelación **Está_En**, para conocer en qué municipio está una determinada vivienda, a partir de **Habita** obtenemos las personas que habitan en ella; sin embargo, puede ocurrir que en una determinada vivienda no habite nadie (cardinalidad mínima 0), por lo que no podemos alcanzar la interrelación **Empadronada** entre persona y municipio. Así, la interrelación **Está_En** no es redundante.

Interrelación **Empadronada**: Si eliminamos la interrelación **Empadronada**, debería ser posible obtener su semántica a partir de **Habita** y **Está_En**. Para conocer el municipio en que está empadronada una persona, mediante **Habita** obtenemos la vivienda en la que habita esa persona y con la interrelación **Está_En** obtenemos el municipio en que se encuentra la vivienda; Por ello, conocemos el municipio en que está empadronada esa persona. En el otro sentido de la interrelación **Empadronada**, debe ser posible conocer las personas empadronadas en un determinado municipio; mediante la interrelación **Está_En** conocemos las viviendas de ese municipio y a partir de **Habita** sabemos todas las personas que viven en esas viviendas, conociendo así todas las personas empadronadas en el municipio. Consecuentemente, la interrelación **Empadronada** se puede eliminar del esquema E/R sin perder semántica.

El esquema E/R definitivo se muestra en la figura 1.7.

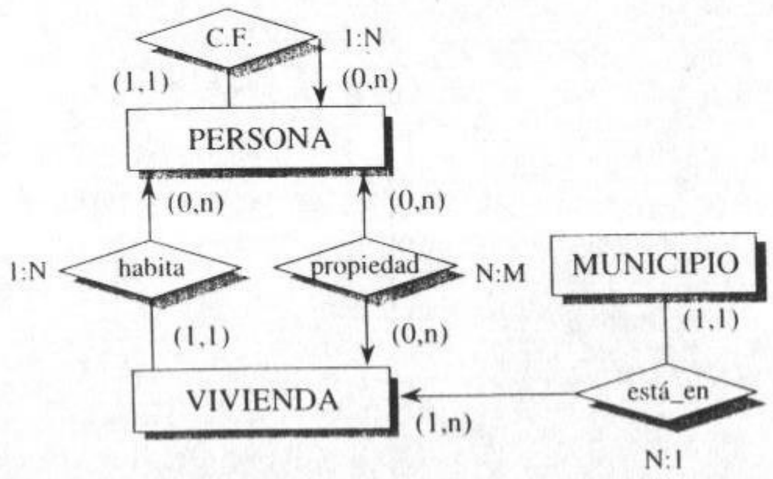


Figura 1.7. Esquema E/R sin redundancias

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

Toda la semántica del problema ha sido reflejada, y no ha sido necesario realizar supuestos adicionales.

PROBLEMA 2: CURSOS DE FORMACIÓN

Enunciado

El departamento de formación de una empresa desea construir una base de datos para planificar y gestionar la formación de sus empleados.

La empresa organiza cursos internos de formación de los que se desea conocer el código de curso, el nombre, una descripción, el número de horas de duración y el coste del curso.

Un curso puede tener como prerrequisito haber realizado otro(s) previamente, y, a su vez la realización de un curso puede ser prerrequisito de otros. Un curso que es un prerrequisito de otro puede serlo de forma obligatoria o sólo recomendable.

Un mismo curso tiene diferentes ediciones, es decir, se imparte en diferentes lugares, fechas y con diferentes horarios (intensivo, de mañana o de tarde). En una misma fecha de inicio sólo puede impartirse una edición de un curso.

Los cursos se imparten por personal de la propia empresa.

De los empleados se desea almacenar su código de empleado, nombre y apellidos, dirección, teléfono, NIF (Número de Identificación Fiscal), fecha de nacimiento, nacionalidad, sexo, firma y salario, así como si está o no capacitado para impartir cursos.

Un mismo empleado puede ser docente en una edición de un curso y alumno en otra edición, pero nunca puede ser ambas cosas a la vez (en una misma edición de curso o lo imparte o lo recibe).

Discusión del enunciado

PARTE 1

> *"La empresa organiza cursos internos de formación de los que se desea conocer el código de curso, el nombre, una descripción, el número de horas de duración y el coste del curso."*

"Un curso puede tener como prerrequisito haber realizado otro(s) previamente, y, a su vez la realización de un curso puede ser prerrequisito de otros. Un curso que es un prerrequisito de otro puede serlo de forma obligatoria o sólo recomendable."

Parece evidente a partir de los primeros párrafos que cada curso de formación será una entidad: un *CURSO* tiene un "Código" que lo identifica, y tendrá como atributos el "Nombre" (el cual consideramos que además es único, es decir, es un identificador alternativo), una "Descripción", el número de horas de "Duración" y su "Coste". Para modelar que un curso puede ser prerrequisito de otros no podemos utilizar un atributo, ya que un curso puede ser prerrequisito de varios. La forma de representarlo es mediante una interrelación reflexiva en la entidad *CURSO*. Las cardinalidades mínimas y máximas en ambos sentidos de la interrelación son (0,n), lo cual significa que:

- Un curso puede ser prerrequisito de ninguno o de varios otros cursos.
- Un curso puede tener como prerrequisito haber cursado previamente ninguno o varios cursos.

Por otra parte, el atributo que indica la obligatoriedad ha de figurar en la interrelación, y no en la entidad, ya que no es propio del curso el ser obligatorio o recomendable, sino de un curso con respecto a otro del que es prerrequisito. Las consideraciones anteriores aparecen en el esquema de la figura 1.8.

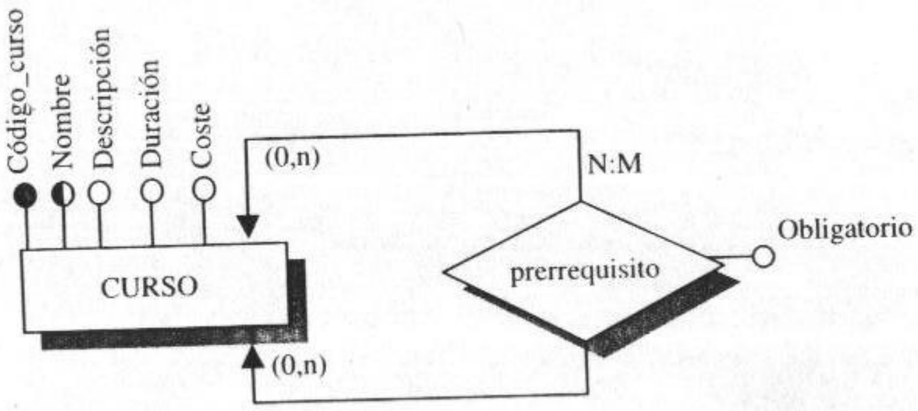


Figura 1.8

PARTE 2

"Un mismo curso tiene diferentes ediciones, es decir, se imparte en diferentes lugares, fechas y con diferentes horarios (intensivo, de mañana o de tarde). En una misma fecha de inicio sólo puede impartirse una edición de un curso."

Puesto que en una misma fecha sólo puede comenzar una única edición del mismo curso, será suficiente con la fecha (junto con el código del Curso) para poder identificarla. Por lo tanto, una *EDICIÓN* será una entidad débil que tendrá una dependencia en identificación con respecto a *CURSO*. Las cardinalidades con respecto a *CURSO* serán (1,1), ya que toda edición lo es de uno y sólo un curso (además, toda dependencia en identificación ha de tener, evidentemente, cardinalidades (1,1) con respecto a la entidad regular). La cardinalidad máxima de una edición con respecto a un curso será *n* (puesto que puede haber distintas ediciones de un mismo curso). En cuanto a la mínima, será 0 si pueden existir cursos de los que no se hayan impartido ediciones, y 1 si todo curso ha de tener al menos una edición. Hemos elegido la cardinalidad menos restrictiva, esto es, 0, para permitir que la empresa pueda tener planificados distintos cursos de los que todavía no se haya celebrado ninguna edición.

La semántica anterior se refleja en el esquema de la figura 1.9.

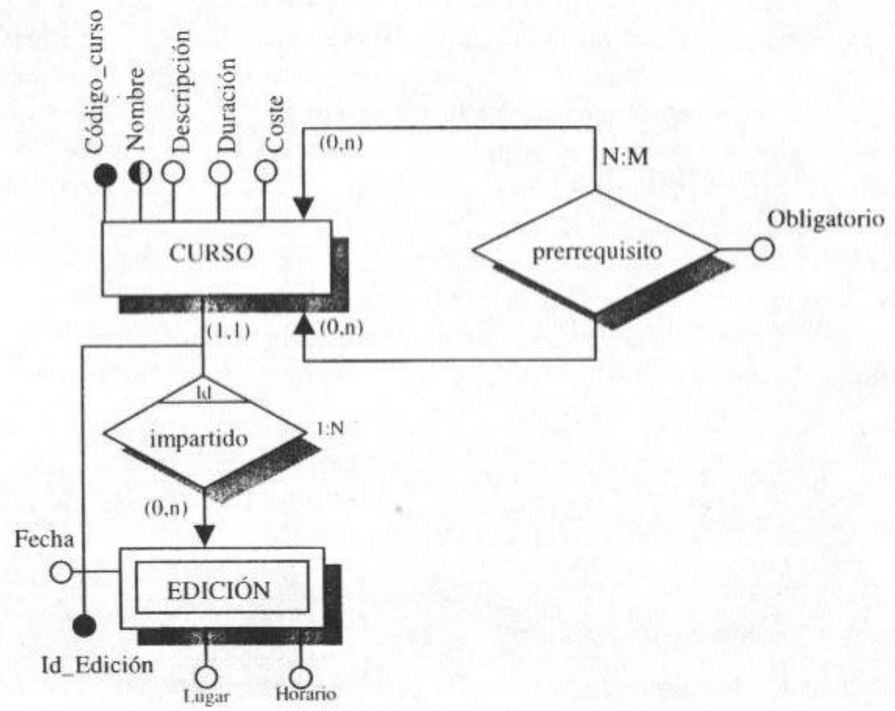


Figura 1.9

PARTE 3

"Los cursos se imparten por personal de la propia empresa."

"De los empleados se desea almacenar su código de empleado, nombre y apellidos, dirección, teléfono, NIF (Número de Identificación Fiscal), fecha de nacimiento, nacionalidad, sexo, firma y salario, así como si está o no capacitado para impartir cursos."

"Un mismo empleado puede ser docente en una edición de un curso y alumno en otra edición, pero nunca puede ser ambas cosas a la vez (en una misma edición de curso o lo imparte o lo recibe)."

El esquema modelado hasta ahora ha sido bastante sencillo. Sin embargo, para modelar toda la semántica asociada al hecho de que los empleados participen (impartan o reciban) en las ediciones de los cursos, tenemos varias posibilidades. En cualquier caso, tendremos una entidad *EMPLEADO*, con un conjunto de atributos, de los cuales "Código de Empleado" será identificador principal, y "NIF" identificador alternativo. Además de otros atributos, como el "Teléfono", "Fecha de Nacimiento", "Nacionalidad", etc., existirá un atributo que especificará si un empleado está o no "Capacitado" para impartir cursos[8].

A continuación ofrecemos distintas posibilidades para modelar la participación de los empleados en las ediciones de los cursos, y veremos las ventajas y desventajas de cada una de ellas (por claridad, en estas figuras no se incluyen los atributos de las entidades):

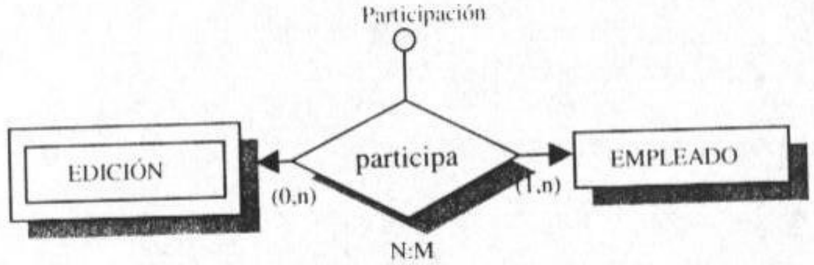


Figura 1.10

La solución más sencilla (figura 1.10) sería modelar la participación de los empleados en las ediciones de los cursos mediante una interrelación **Participa** entre *EMPLEADO* y *EDICIÓN*. Dicha interrelación tendría un atributo "Participación", cuyos posibles valores serían "Docente" y "Alumno". Esta solución recoge que en un curso un empleado sólo puede participar como docente o como alumno, pero no como ambas cosas a la vez. Sin embargo esta solución no recoge las siguientes características:

- Que un empleado no capacitado para impartir cursos no pueda participar en una edición como docente

[8] Esto es así porque en los requisitos se nos indica que la capacitación depende únicamente del empleado. Podría plantearse una ampliación del problema permitiendo que dicha capacitación dependiera no sólo del empleado sino también de cada curso (es decir, que un empleado estuviera capacitado para

- Que en una edición deba haber al menos un alumno y un docente (la solución sólo exige que haya un participante)

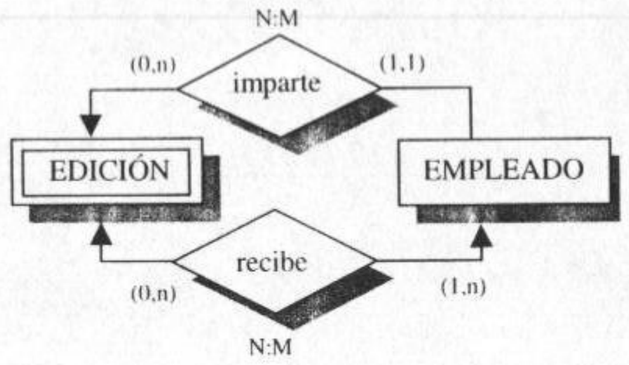


Figura 1.11

Podemos ampliar la solución anterior, dividiendo la interrelación **Participa** en dos interrelaciones, **Imparte** y **Recibe**, como muestra la figura 1.11. De esta manera resolvemos que en una edición deba haber un docente y, al menos, un alumno. La cardinalidad máxima de **Imparte** sería 1 si un curso lo imparte una única persona o N si suponemos que lo pueden impartir más. Sin embargo, seguimos sin resolver el hecho de impedir que empleados no capacitados puedan impartir cursos, y además hemos añadido un problema que con la solución de la figura 1.10 teníamos resuelto: que en una misma edición un mismo empleado participe como docente y como alumno. Esto no se resolvería mediante una restricción de exclusividad (en *EMPLEADO*), ya que significaría que un empleado o siempre es alumno, o siempre es profesor, pero no que pueda ser unas veces alumno y otras profesor, pero siempre en distintas ediciones de cursos.

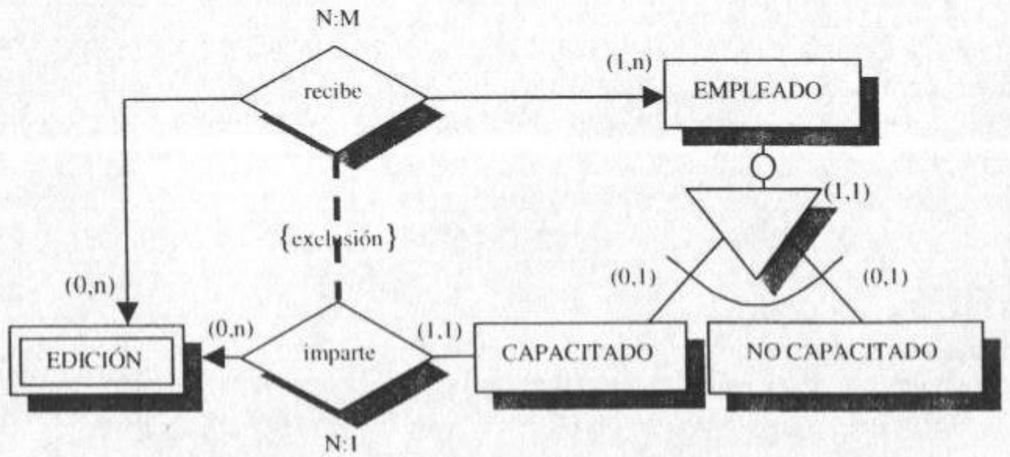


Figura 1.12

Una mejora a la solución de la figura 1.11 sería añadir una jerarquía de empleados (*CAPACITADOS* y *NO_CAPACITADOS*, o únicamente *CAPACITADOS*) en la que el atributo discriminante fuera "Capacitado", y dos interrelaciones, **Imparte** y **Recibe**. Con esta solución, que se muestra en la figura 1.12, resolveríamos los dos aspectos que no se recogían con la solución de la figura 1.10, pero seguimos sin resolver el impedir que un mismo empleado participe en la misma edición como alumno y como docente, ya que en la interrelación **Recibe** participan tanto los empleados capacitados como los no capacitados. Para resolverlo deberíamos exigir que una ocurrencia de la interrelación **Recibe** no apareciera en la interrelación **Imparte** y viceversa, es decir, una restricción de exclusión. Esta notación procede de los modelos de objetos, y puede consultarse en De Miguel *et al.* (1999).

Propuesta de solución

Hemos elegido la opción de la figura 1.12 porque, como hemos visto, es la que más semántica recoge. La figura 1.13 muestra el esquema definitivo.

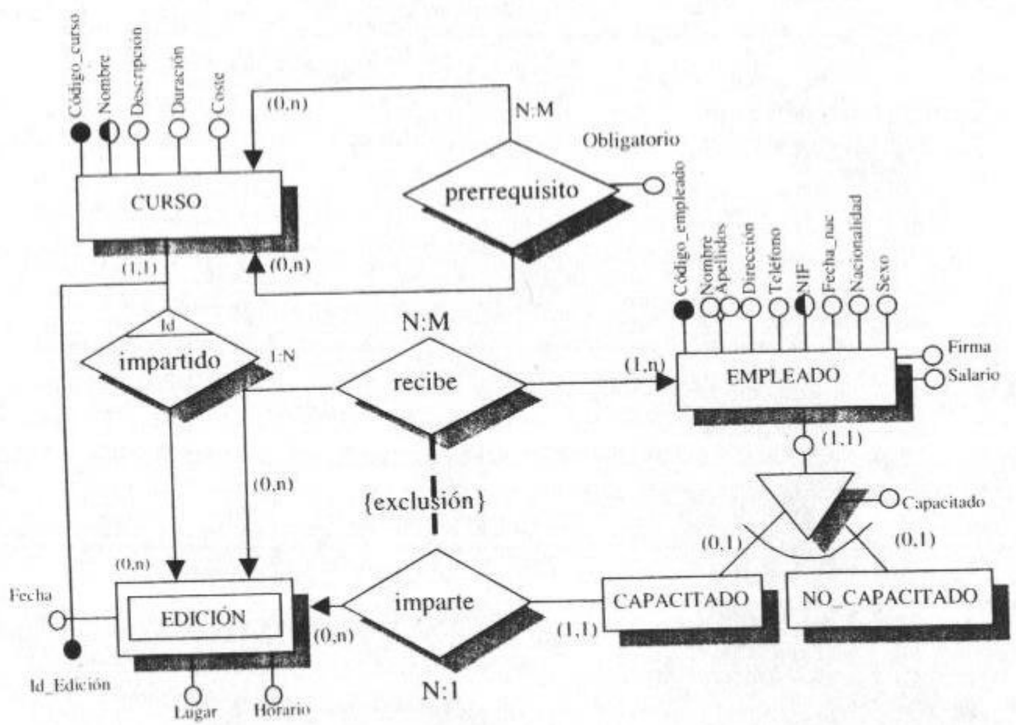


Figura 1.13

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha considerado el nombre del curso como un identificador alternativo.

PROBLEMA 3: CAMPEONATO DE AJEDREZ

Enunciado

El club de Ajedrez de Villatortas de Arriba, ha sido encargado por la Federación Internacional de Ajedrez de la organización de los próximos campeonatos mundiales que se celebrarán en la mencionada localidad. Por este motivo, desea llevar a una base de datos toda la gestión relativa a participantes, alojamientos y partidas. Teniendo en cuenta que:

En el campeonato participan jugadores y árbitros; de ambos se requiere conocer el número de asociado, nombre, dirección, teléfono de contacto y campeonatos en los que han participado (como jugador o como árbitro). De los jugadores se precisa además el nivel de juego en una escala de 1 a 10.

Ningún árbitro puede participar como jugador.

Los países envían al campeonato un conjunto de jugadores y árbitros, aunque no todos los países envían participantes. Todo jugador y árbitro es enviado por un único país. Un país puede ser representado por otro país.

Cada país se identifica por un número correlativo según su orden alfabético e interesa conocer además de su nombre, el número de clubes de ajedrez existentes en el mismo.

Cada partida se identifica por un número correlativo (Cód_P), la juegan dos jugadores y la arbitra un árbitro. Interesa registrar las partidas que juega cada jugador y el color (blancas o negras) con el que juega. Ha de tenerse en cuenta que un árbitro no puede arbitrar a jugadores enviados por el mismo país que le ha enviado a él.

Todo participante participa en al menos una partida.

Tanto jugadores como árbitros se alojan en uno de los hoteles en los que se desarrollan las partidas, se desea conocer en qué hotel y en qué fechas se ha alojado cada uno de los participantes. Los participantes pueden no permanecer en Villatortas durante todo el campeonato, sino acudir cuando tienen que jugar alguna partida alojándose en el mismo o distinto hotel. De cada hotel, se desea conocer el nombre, la dirección y el número de teléfono.

El campeonato se desarrolla a lo largo de una serie de jornadas (año, mes, día) y cada partida tiene lugar en una de las jornadas aunque no tengan lugar partidas todas las jornadas.

Cada partida se celebra en una de las salas de las que pueden disponer los hoteles, se desea conocer el número de entradas vendidas en la sala para cada partida. De cada sala, se desea conocer la capacidad y medios de que dispone (radio, televisión, vídeo...) para facilitar la retransmisión de los encuentros. Una sala puede disponer de varios medios distintos.

De cada partida se pretende registrar todos los movimientos que la componen, la identificación de movimiento se establece en base a un número de orden dentro de cada partida: para cada movimiento se guardan la jugada (5 posiciones) y un breve comentario realizado por un experto.

Discusión del Enunciado

PARTE 1

> *"En el campeonato participan jugadores y árbitros; de ambos se requiere conocer el número de asociado, nombre, dirección, teléfono de contacto y campeonatos en los que han participado (como jugador o como árbitro). De los jugadores se precisa además el nivel de juego en una escala de 1 a 10."*

En este primer supuesto aparecen dos entidades: *ARBITRO* y *JUGADOR*; que comparten muchos de sus atributos, por lo que puede pensarse en abstraer ambas entidades a un supertipo *PARTICIPANTE* del cual éstas serían subtipos. Los atributos comunes son: número de asociado ("N_Socio"), que actúa como identificador principal, "Nombre", "Dirección" y "Teléfono", que se consideran atributos obligatorios y, por ultimo, para recoger los distintos campeonatos en los que han participado tanto jugadores como árbitros, se crea un atributo "Campeonato", que será multivaluado, opcional y compuesto por "Nombre" del campeonato y "Tipo" (árbitro o jugador). Se podría haber considerado "Campeonato" como un tipo de entidad, de esta forma se podría guardar mas información acerca de éstos pero al leer todo el enunciado nos damos cuenta de que no precisamos mayor detalle. El subtipo *JUGADOR* tendrá un atributo propio denominado "Nivel" de juego.

> *"Ningún árbitro puede participar como jugador."*

El haber reflejado el anterior supuesto como una generalización nos permite reflejar esta nueva especificación en la cual se aclaran cuáles son las propiedades de la jerarquía anterior. Las personas que participan dentro del mismo campeonato son o jugadores o árbitros y sólo pueden actuar como tal, por lo que la generalización es exclusiva, puede suponerse además que es total por conocimiento del universo de discurso.

> *"Los países envían al campeonato un conjunto de jugadores y árbitros, aunque no todos los países envían participantes. Todo jugador y árbitro es enviado por un único país. Un país puede ser representado por otro país."*

> *"Cada país se identifica por un número correlativo según su orden alfabético e interesa conocer además de su nombre, el número de clubes de ajedrez existentes en el mismo."*

Se crea un nuevo tipo de entidad *PAÍS* cuyo identificador principal es un número correlativo según un orden alfabético que no podemos representar en el esquema pero que debemos tener en cuenta a efectos de implementación. *PAÍS* tiene otros dos atributos más, "Nombre", que además de obligatorio se considera identificador alternativo y "N_Club".

La entidad *PAÍS* se relaciona con la entidad *PARTICIPANTE*, por lo que se creará una interrelación **Envía** que tiene como cardinalidades asociadas a *PAÍS* (0,n), ya que en el texto se indica que puede haber países que no envíen participantes. Las cardinalidades asociadas a *PARTICIPANTE* son (1,1) porque todos los participantes son enviados por un y sólo un país. Habría sido más complejo representar esta interrelación si no se hubiera introducido la jerarquía de participantes, pues se habría tenido que establecer al nivel de los subtipos, duplicándose esta interrelación.

Como un país puede ser representando por otro país se creará una interrelación reflexiva **Representa** cuyas cardinalidades son (0,1) para reflejar el hecho de que un país puede o no ser representado por otro, y (0,n) pues un país podría representar a varios países.

La figura 1.14 refleja las consideraciones anteriores.

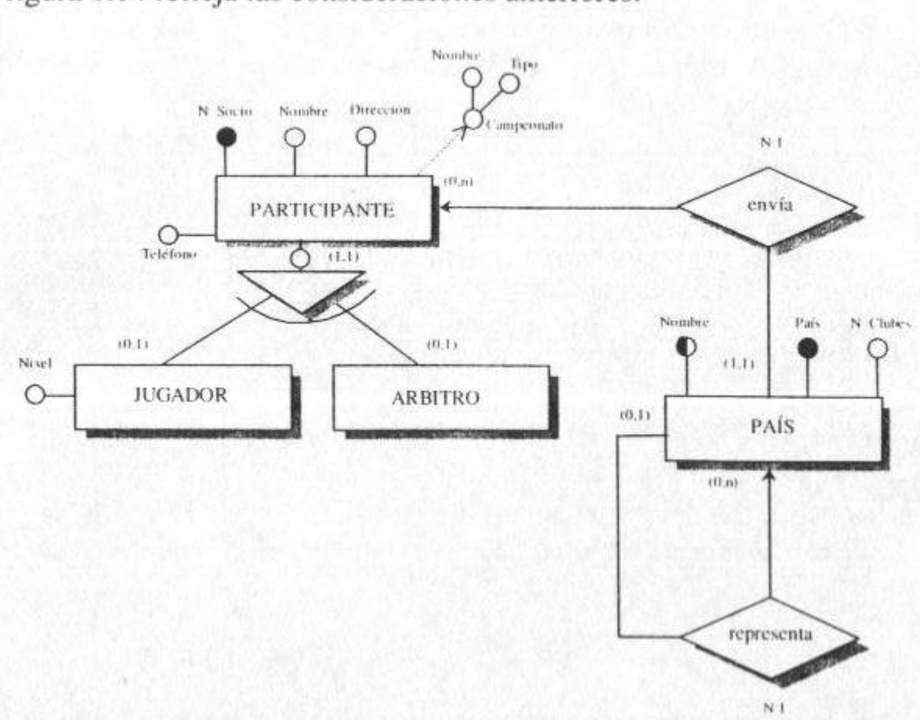


Figura 1.14

"Cada partida se identifica por un número correlativo (Cod_P), la juegan dos jugadores y la arbitra un árbitro. Interesa registrar las partidas que juega cada jugador y el color (blancas o negras) con el que juega. Ha de tenerse en cuenta que un árbitro no puede arbitrar a jugadores enviados por el mismo país que le ha enviado a él."

"Todo participante participa en al menos una partida."

Se añade la entidad *PARTIDA* con identificador principal "Cód_P", número elativo, propiedad que no se puede reflejar en el esquema E/R pero que se debe en cuenta cuando se implemente la BD.

Una partida de ajedrez la juegan dos jugadores y la arbitra un árbitro, hecho que epresenta mediante dos interrelaciones binarias, ya que existen dos asociaciones ntas, **Juega** y **Arbitra**. La interrelación **Juega** tendrá como cardinalidades aso-as a *PARTIDA* (2,2) y las asociadas a *JUGADOR* serán (1,n). Además, para saber qué color ha participado cada jugador en las distintas partidas disputadas, ten-nos un atributo obligatorio "Color" en la interrelación, cuyo dominio estará com-to de dos valores, blanco y negro. La interrelación **Arbitra** tendrá como cardinali-es (1,1) para la entidad *PARTIDA* y (1,n) para la entidad *ÁRBITRO*.

Se ha de tener en cuenta que el modelo E/R no controla la restricción de que un ro no puede arbitrar a jugadores de su mismo país y tampoco que los valores del uto "Color" sean distintos para jugadores que desarrollen la misma partida, pero or este hecho debemos de ignorarlo sino que la guardaremos para contemplarlo en mplementación, pues se trata de supuestos semánticos que el modelo E/R no es z de recoger.

Una posibilidad alternativa para recoger el anterior supuesto habría sido tratar de siderar una interrelación ternaria, pero la semántica, en especial la referente a inalidades, se refleja con mayor precisión, en este caso mediante interrelaciones rias.

"Tanto jugadores como árbitros se alojan en uno de los hoteles en los que se desarrollan las partidas, se desea conocer en qué hotel y en qué fechas se ha alojado cada uno de los participantes. Los participantes pueden no permanecer en Villatortas durante todo el campeonato, sino acudir cuando tienen que jugar alguna partida alojándose en el mismo o distinto hotel. De cada hotel, se desea conocer el nombre, la dirección y el número de teléfono."

Como los participantes se alojan en uno o varios hoteles durante el campeonato, que no es obligatorio que permanezcan en él durante todo el campeonato, se creará entidad *HOTEL* cuyo identificador principal será "Nombre" y como atributos rnativos se presentarán "Dirección" y "Teléfono". Y se relacionarán *HOTEL* y *RTICIPANTE* mediante una interrelación **Aloja** cuyas cardinalidades asociadas n (0,n) y (1,n), respectivamente. Con la cardinalidad mínima 0 para *HOTEL* se

contemplará que en un momento dado puede que no existan participantes alojados en el mismo. Además, las estancias de un participante pueden ser intermitentes, con lo cual la interrelación posee dos atributos multivaluados, "F_Entrada" y "F_Salida", que recogerán este supuesto. Para reflejar las instancias acutales, en las cuales, evidentemente no habrá una fecha de salida, el atributo "f_salida", será opcional.

Como restricción adicional deducida de la interpretación del Universo de Discurso se debería asegurar que la fecha de salida sea mayor que la de entrada y que los periodos de tiempo no se solapen entre sí.

La semántica anterior aparece reflejada en el esquema de la figura 1.15.

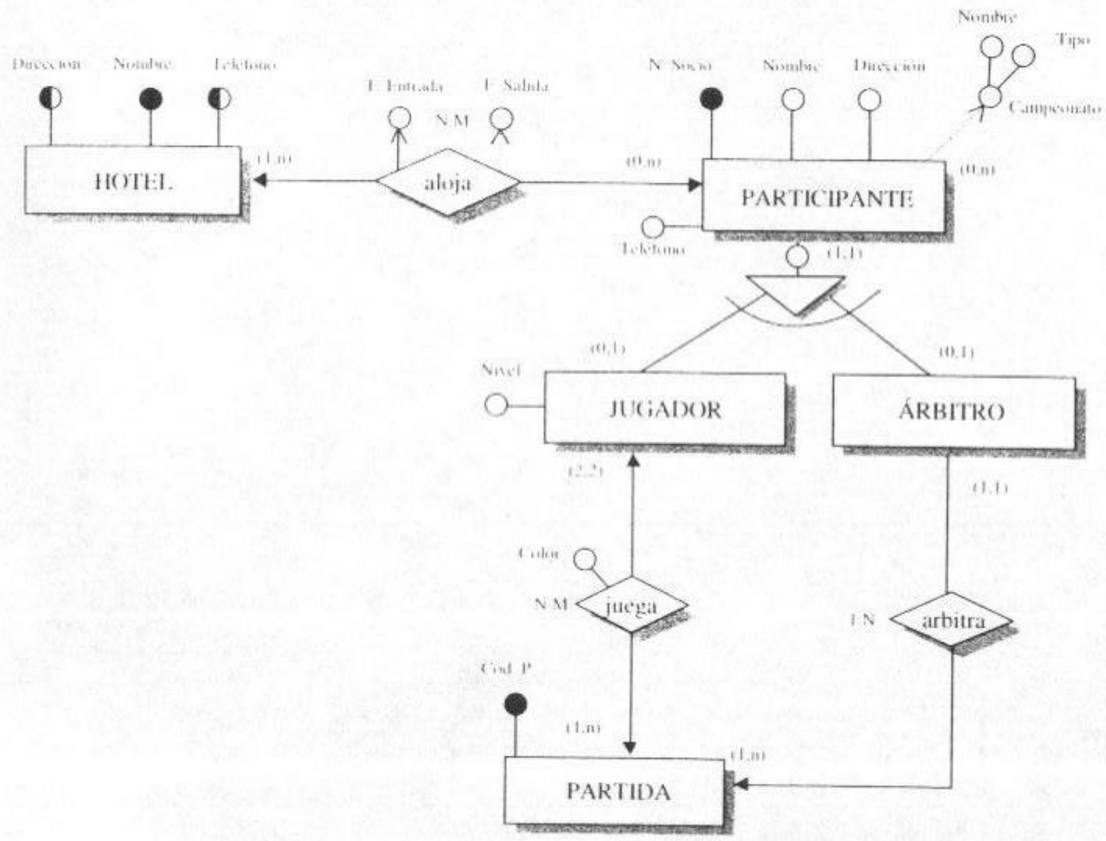


Figura 1.15

"El campeonato se desarrolla a lo largo de una serie de jornadas (año, mes, día) y cada partida tiene lugar en una de las jornadas aunque no tengan lugar partidas todas las jornadas."

Dependiendo de la información que se quiera recoger acerca de las jornadas del campeonato tendremos un nuevo tipo de entidad *JORNADAS* que reflejará los acontecimientos del campeonato haya o no partidas, o un nuevo atributo "Jornada"

estaría compuesto por tres atributos, "Año", "Mes" y "Día", que recogerá sólo información acerca de los días en que se han celebrado partidas. Se tomará esta última ión, ya que en el enunciado no se referencia a las jornadas en que no hay partidas que el sistema de información contempla sólo las jornadas donde éstas existen.

> *"Cada partida se celebra en una de las salas de las que pueden disponer los hoteles, se desea conocer el número de entradas vendidas en la sala para cada partida. De cada sala, se desea conocer la capacidad y medios de que dispone (radio, televisión, vídeo...) para facilitar la retransmisión de los encuentros. Una sala puede disponer de varios medios distintos."*

Aparece en este párrafo una nueva entidad *SALA*, cuya existencia depende del el al que pertenezca, por lo que será una entidad débil y que se relacionará con *OTEL* mediante una interrelación en existencia. Como en las especificaciones del to no aparece ninguna característica de cada sala que pueda ejercer las labores de ntificador principal, se añade a los atributos de ésta, que son "Capacidad" y ledios", un "Cód_Sala". El atributo "Medios" es multivaluado y opcional, ya que en a sala pueden existir, o no, distintos medios audiovisuales.

La interrelación **Dispone** que asocia *HOTEL* y *SALA*, comentada anteriormente r ser una interrelación en existencia, tendrá como cardinalidades (0,n) y (1,1), spectivamente. Recogiendo con la cardinalidad mínima 0 que puede haber hoteles e no dispongan de sala y sólo tengan la función de alojar participantes.

La otra interrelación que aparece en este párrafo asocia la entidad *SALA* con *ARTIDA* y posee un atributo "Entradas" que indica el número de entradas que se han ndido para una determinada partida. Una restricción que se deberá tener en cuenta steriormente para la implementación de este esquema es que el valor de este atributo puede exceder al del aforo o capacidad de la sala. Las cardinalidades asociadas son ,1) para la entidad *PARTIDA* y (1,n) para *SALA*. Esta ultima cardinalidad mínima bliga a que sólo tengamos información de aquellas salas que se utilizan en el mpeonato aunque el hotel pudiera tener más.

> *"De cada partida se pretende registrar todos los movimientos que la componen, la identificación de movimiento se establece en base a un número de orden dentro de cada partida: para cada movimiento se guardan la jugada (5 posiciones) y un breve comentario realizado por un experto."*

Por ultimo, se representará la entidad *MOVIMIENTO* dependiente de cada *PARIDA* que detallará el desarrollo de la misma. Según se muestra en el enunciado, para identificación de cada movimiento será necesario el identificador principal de *PARIDA*, por lo que tendremos una interrelación en identificación denominada **Consta**.

El esquema de la figura 1.16 refleja los comentarios anteriores.

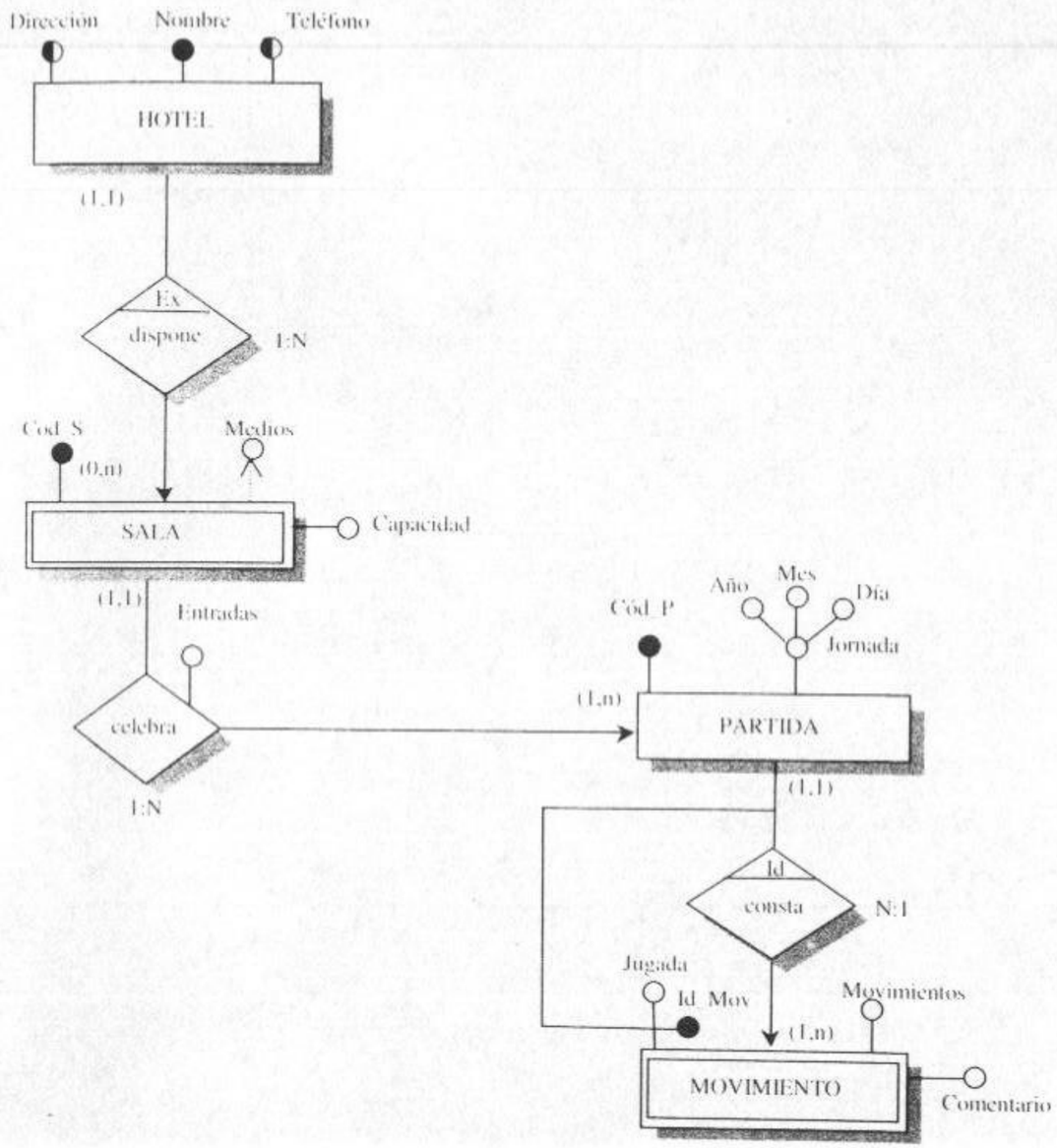


Figura 1.16

Propuesta de solución

Agrupamos las distintas soluciones parciales obtenidas en la figura 1.17.

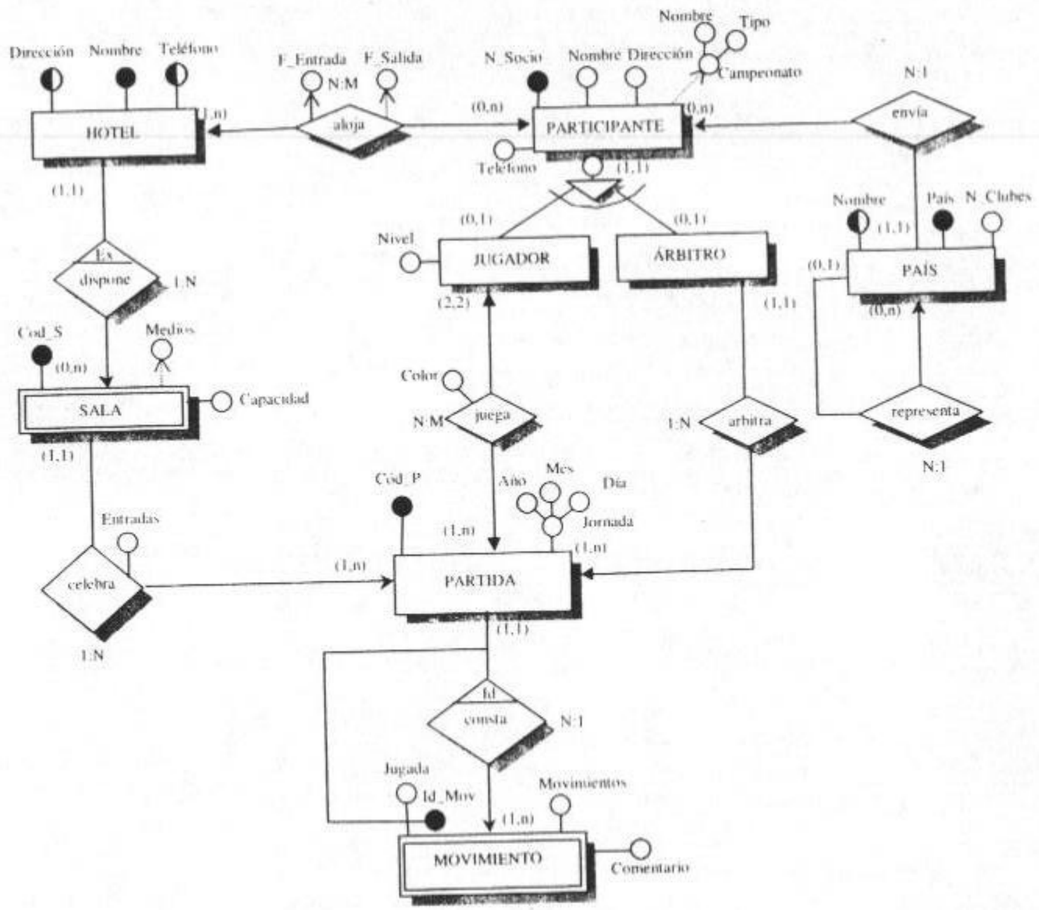


Figura 1.17

En la solución obtenida, hay ciclos, pero no son semánticamente equivalentes.

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha supuesto que no hay más participantes que árbitros y jugadores.

No se ha podido reflejar:

- Un árbitro no puede arbitrar partidas en las que participen jugadores de su mismo país.
- En cada partida uno de los jugadores juega con blancas y el otro con negras.
- Secuencialidad del código de país.
- Secuencialidad del código de partida.
- Secuencialidad del número de movimiento dentro de cada partida.

PROBLEMA 4: ENERGÍA ELÉCTRICA

Enunciado

Se pretende llevar a cabo un control sobre la energía eléctrica que se produce y consume en un determinado país. Se parte de las siguientes hipótesis.

Existen productores básicos de electricidad que se identifican por un nombre, de los cuales interesa su producción media, producción máxima y fecha de entrada en funcionamiento. Estos productores básicos lo son de una de las siguientes categorías: Hidroeléctrica, Solar, Nuclear o Térmica. De una central hidroeléctrica o presa nos interesa saber su ocupación, capacidad máxima y número de turbinas. De una central solar nos interesa saber la superficie total de paneles solares, la media anual de horas de sol y tipo (fotovoltaica o termodinámica). De una central nuclear, nos interesa saber el número de reactores que posee, el volumen de plutonio consumido y el de residuos nucleares que produce. De una central térmica, nos interesa saber el número de hornos que posee, el volumen de carbón consumido y el volumen de su emisión de gases.

Por motivos de seguridad nacional interesa controlar el plutonio de que se provee una central nuclear, este control se refiere a la cantidad de plutonio que compra a cada uno de sus posibles suministradores, (nombre y país), y que porta un determinado transportista (nombre y matrícula), ha de tenerse en cuenta que el mismo suministrador puede vender plutonio a distintas centrales nucleares y que cada porte, (un único porte por compra), puede realizarlo un transportista diferente.

Cada día, los productores entregan la energía producida a una o varias estaciones primarias, las cuales pueden recibir diariamente una cantidad distinta de energía de cada uno de estos productores. Los productores entregan siempre el total de su producción. Las estaciones primarias se identifican por su nombre y tienen un número de transformadores de baja a alta tensión y son cabecera de una o varias redes de distribución

Una red de distribución se identifica por un número de red y sólo puede tener una estación primaria como cabecera. La propiedad de una red puede ser compartida por varias compañías eléctricas, a cada compañía eléctrica se le identifica por su nombre.

La energía sobrante en una de las redes puede enviarse a otra red. Se registra el volumen total de energía intercambiada entre dos redes.

Una red está compuesta por una serie de líneas, cada línea se identifica por un número secuencial dentro del número de red y tiene una determinada longitud. La menor de las líneas posibles abastecerá al menos a dos subestaciones.

Una subestación es abastecida sólo por una línea y distribuye a una o varias zonas de servicio, a tales efectos, las provincias (código y nombre), se encuentran divididas en tales zonas de servicio, aunque no puede haber zonas de servicio que pertenezcan a más de una provincia. Cada zona de servicio puede ser atendida por más de una subestación.

En cada zona de servicio se desea registrar el consumo medio y el número de consumidores finales de cada una de las siguientes categorías: particulares, empresas e instituciones.

Discusión del enunciado

PARTE 1

> *"Existen productores básicos de electricidad que se identifican por un nombre, de los cuales interesa su producción media, producción máxima y fecha de entrada en funcionamiento. Estos productores básicos lo son de una de las siguientes categorías: Hidroeléctrica, Solar, Nuclear o Térmica. De una central hidroeléctrica o presa nos interesa saber su ocupación, capacidad máxima y número de turbinas. De una central solar nos interesa saber la superficie total de paneles solares, la media anual de horas de sol y tipo (fotovoltaica o termodinámica). De una central nuclear, nos interesa saber el número de reactores que posee, el volumen de plutonio consumido y el de residuos nucleares que produce. De una central térmica, nos interesa saber el número de hornos que posee, el volumen de carbón consumido y el volumen o su emisión de gases."*

En este primer párrafo se identifica inicialmente el tipo de entidad *PRODUCTOR*, cuyos atributos son "Nombre", "Producción Media", "Producción Máxima" y "Fecha Inicio". El atributo "Nombre" identifica a cada productor según el enunciado, por lo que lo consideramos como identificador principal. Los productores pueden ser de distintas categorías, esto sugiere introducir una jerarquía donde el supertipo será *PRODUCTOR* y los subtipos *PRESA*, *SOLAR*, *NUCLEAR* y *TÉRMICA*. De todos los atributos de cada uno de los subtipos es de interés mencionar el atributo "Tipo" del subtipo *SOLAR*, este atributo estará definido sobre un dominio compuesto por los valores <fotovoltaica, termodinámica>, de manera que no podrá tomar ningún otro valor, no tiene la suficiente entidad para considerarle como atributo discriminante de una jerarquía.

> *"Por motivos de seguridad nacional interesa controlar el plutonio de que se provee una central nuclear, este control se refiere a la cantidad de plutonio que compra a cada uno de sus posibles suministradores, (nombre y país), y que porta un determinado transportista (nombre y matrícula), ha de tenerse en cuenta que el mismo suministrador puede vender plutonio a distintas centrales nucleares y que cada porte, (un único porte por compra), puede realizarlo un transportista diferente."*

La lectura de este párrafo nos obliga a introducir dos nuevas entidades *SUMINISTRADOR* y *TRANSPORTISTA*. De los atributos de cualquiera de las dos entidades se puede considerar el "Nombre" como identificador principal, pues como no se nos dice nada en contra podemos suponer que éste es único.

Las centrales nucleares compran plutonio a suministradores, el cual es transportado por distintos transportistas. Esta frase nos induce a pensar en la posibilidad de introducir una interrelación ternaria **Compra** entre las entidades *NUCLEAR*, *SUMINISTRADOR* y *TRANSPORTISTA*. Pasemos a analizar las cardinalidades de los tipos de entidad que intervienen:

El plutonio que una determinada central nuclear compra a un suministrador es obvio que puede ser transportado como mínimo por un transportista y como máximo por *n*. Un suministrador y un mismo transportista pueden suministrar plutonio a varias centrales nucleares (cardinalidad máxima *n*) y como mínimo a una, pues si no suministraran a ninguna esa ocurrencia no intervendría en la interrelación. Por último, una central nuclear que recibe el plutonio transportado por un transportista, puede habérselo comprado (en la misma o en diferentes ocasiones) a uno o a varios suministradores, por lo que las cardinalidades mínima y máxima respectivamente de esta interrelación en el extremo de la entidad *SUMINISTRADOR* son 1 y *n*.

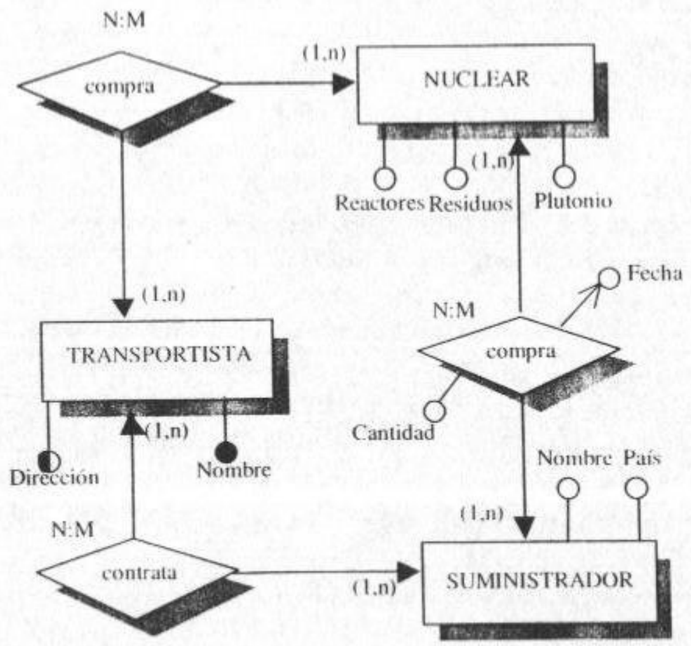


Figura 1.18

El enunciado nos dice que debemos controlar la cantidad de plutonio comprada por una central nuclear, por tanto esta cantidad debe ser un atributo propio de la interrelación ternaria. Además parece claro, aunque no lo diga explícitamente el enunciado, que cada central nuclear puede comprar cantidades distintas o iguales en

distintas fechas, luego se debe introducir un atributo multivaluado "Fecha" en la misma interrelación.

Como siempre que nos enfrentamos ante interrelaciones ternarias, debemos hacer un análisis para ver si podemos descomponer la interrelación ternaria en binarias como se muestra en la figura 1.18 (ver figura en la página anterior), pero es evidente que no se recoge la misma semántica (no podría por ejemplo determinarse qué transportista ha realizado el porte de una determinada compra). No obstante las cardinalidades máximas de los tipos de entidad involucrados en la interrelación ternaria, así como los atributos presentes en la interrelación, indican que no puede descomponerse.

El resultado de todas las consideraciones hechas en esta primera parte queda reflejado en la figura 1.19.

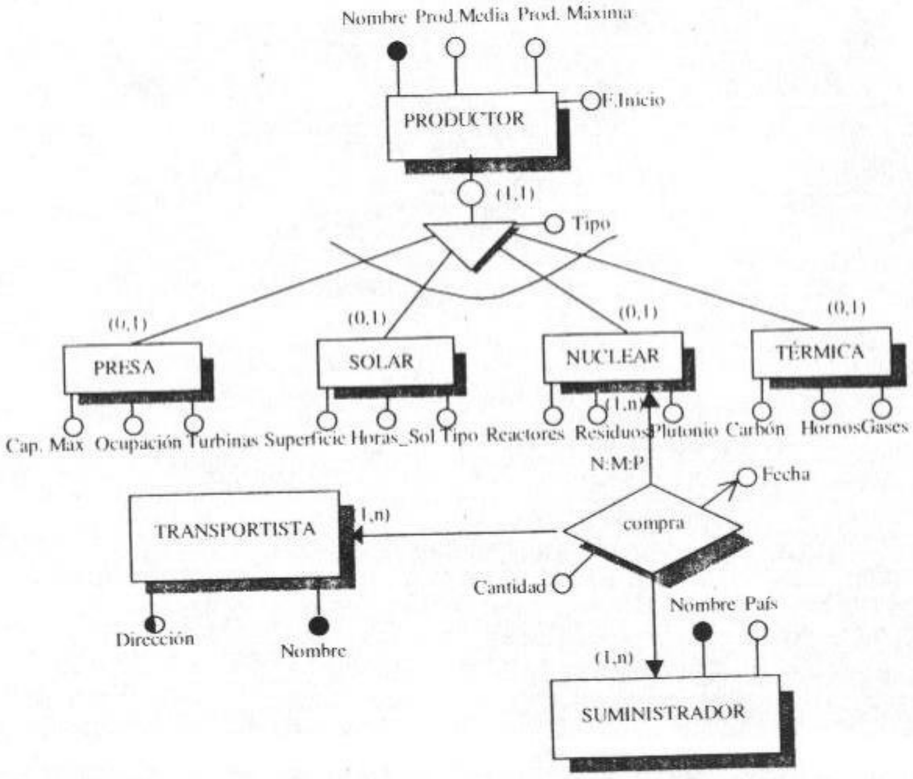


Figura 1.19

PARTE 2

> *"Cada día, los productores entregan la energía producida a una o varias estaciones primarias, las cuales pueden recibir diariamente una cantidad distinta de cada uno de estos productores. Los productores entregan siempre el total de su producción. Las estaciones primarias se identifican por su nombre y tienen un número de transformadores de baja a alta tensión y son cabecera de una o varias redes de distribución."*

En este párrafo se introduce una nueva entidad *ESTACIÓN PRIMARIA* que se relaciona con *PRODUCTOR* a través de la interrelación **Entrega**, puesto que los productores deben entregar la energía producida a las estaciones primarias. Además las cantidades se entregan diariamente y pueden ser distintas, por lo que la interrelación **Entrega** debe tener dos atributos "Cantidad" y "Fecha", este último multivaluado, pues cada productor puede entregar cantidades distintas o iguales a una misma estación primaria en distintas fechas.

En cuanto a las cardinalidades mínima y máxima de esta interrelación en el extremo de la entidad *ESTACIÓN PRIMARIA* el enunciado nos dice que deben ser 1 y *n* respectivamente. Cada estación puede recibir energía de varios productores, lo cual nos indica que la cardinalidad máxima en el extremo de *PRODUCTOR* es *n*, en cambio, en cuanto a la cardinalidad mínima, si suponemos que toda estación debe recibir la producción de al menos un productor, la cardinalidad mínima será 1, si por el contrario suponemos que puede darse el caso de que en una estación primaria no reciba energía de ningún productor en una fecha determinada, la cardinalidad mínima sería 0. Nosotros hemos optado por esta última suposición en el intento de reflejar el caso más general.

En cuanto a los atributos del tipo de entidad *ESTACIÓN PRIMARIA*, "Nombre" y "Transformadores", el primero es el identificador principal de la entidad, pues se supone único, el segundo debe almacenar el número de transformadores en el rango de baja a alta tensión.

> *"Una red de distribución se identifica por un número de red y sólo puede tener una estación primaria como cabecera. La propiedad de una red puede ser compartida por varias compañías eléctricas, a cada compañía eléctrica, se le identifica por su nombre."*

Cada estación primaria es cabecera de una o varias redes de distribución, por lo que debemos introducir un nuevo tipo de entidad *RED* que se interrelaciona con *ESTACIÓN PRIMARIA* mediante la interrelación **Cabecera**. Cada estación primaria es cabecera de una o varias redes de distribución, por lo que es claro que las cardinalidades mínima y máxima de la interrelación en el extremo del tipo de entidad *RED* serán 1 y *n* respectivamente. Por otro lado cada red sólo puede tener una estación primaria como cabecera, lo que nos indica que la cardinalidad máxima de la interrelación en el extremo de este tipo de entidad es 1, y por tanto la correspondencia será 1:N. Si suponemos que una red puede no tener ninguna cabecera, consideraríamos la cardinalidad mínima del tipo de entidad *ESTACIÓN PRIMARIA* 0, si por el contrario suponemos que toda red debe tener como cabecera a al menos una estación primaria, la cardinalidad mínima sería 1. Nosotros hemos optado por esto último.

Por último, debemos introducir un nuevo tipo de entidad *COMPAÑÍA ELÉCTRICA*, para reflejar que toda red debe ser propiedad de al menos una compañía eléctrica y lo reflejaremos a través de la interrelación **Pertenece**, cuyo tipo de

correspondencia es N:M, pues una compañía eléctrica puede compartir la propiedad de varias redes. Además suponemos que al menos cualquier compañía es propietaria de al menos una red.

> *"La energía sobrante en una de las redes puede enviarse a otra red. Se registra el volumen total de energía intercambiada entre dos redes."*

Este párrafo nos indica la necesidad de introducir una interrelación reflexiva **Envía** en el tipo de entidad *RED*. En principio, si una red dispone de excedente de energía se la puede enviar a otra red (cardinalidad mínima 0) o a otra u otras redes (cardinalidad máxima *n*). En el sentido contrario las cardinalidades son también (0,n), ya que una red puede no recibir energía o bien recibirla de cualquier número de redes. Además se debe registrar la cantidad de energía intercambiada entre dos redes, por lo que debemos introducir un atributo "Volumen" en la interrelación **Envía**.

A diferencia de lo que sucedía en la interrelación **Entrega** en la que era preciso introducir el atributo multivaluado "Fecha" para diferenciar entre diferentes entregas, en este caso, el atributo "Volumen" tiene carácter acumulativo, por lo que no es preciso diferenciar los diferentes envíos de energía entre redes.

Todo lo expuesto en esta parte queda reflejado en la figura 1.20.

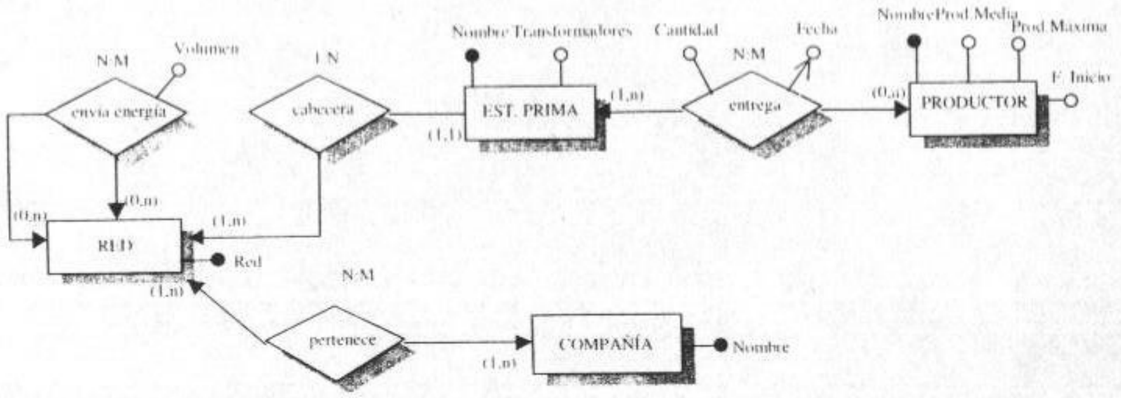


Figura 1.20

PARTE 3

> *"Una red está compuesta por una serie de líneas, cada línea se identifica por un número secuencial dentro del número de red y tiene una determinada longitud. La menor de las líneas posibles abastecerá al menos a dos subestaciones."*
>
> *"Una subestación es abastecida sólo por una línea y distribuye a una o varias zonas de servicio, a tales efectos, las provincias (código y nombre), se encuentran divididas en tales zonas de servicio, aunque no puede haber zonas de servicio que pertenezcan a más de una provincia. Cada zona de servicio puede ser atendida por más de una subestación."*
>
> *"En cada zona de servicio se desea registrar el consumo medio y el número de consumidores finales de cada una de las siguientes categorías: particulares, empresas e instituciones."*

En esta última parte se pueden distinguir claramente cuatro tipos de entidades más *LÍNEA*, *SUBESTACIÓN*, *ZONA* y *PROVINCIA*. *LÍNEA* será un tipo de entidad débil con respecto a *RED*, pues una línea existe en función de que exista la red que la incluye. La interrelación **Compone** entre estos dos tipos de entidades será de dependencia en identificación, pues parece claro que el identificador principal de *LÍNEA* está constituido por un número más el número de red en el que se incluye la línea; el tipo de correspondencia será 1:N pues una red se compone de varias líneas (cardinalidad máxima *n*) y como mínimo podemos suponer que de una (cardinalidad mínima 1).

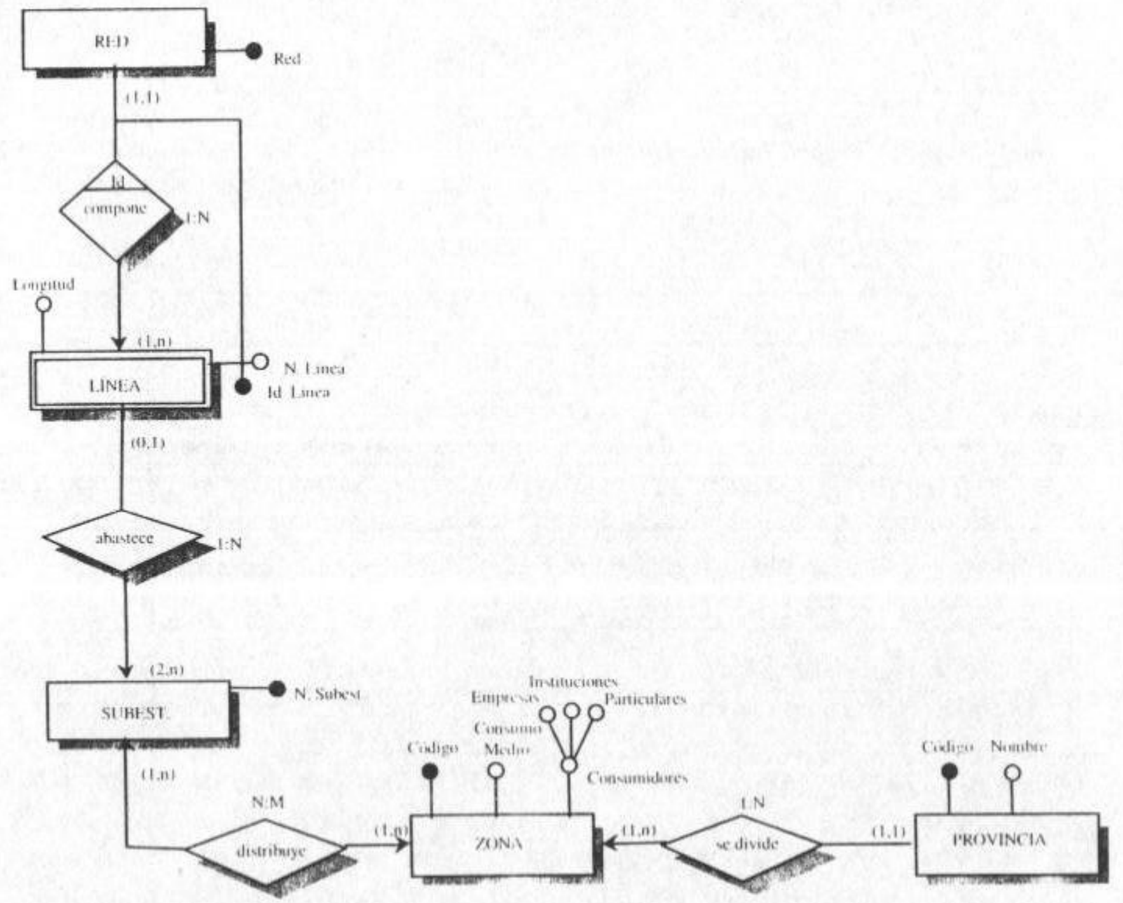


Figura 1.21

Existe otra interrelación **Abastece** entre *LÍNEA* y *SUBESTACIÓN*. Cada línea al menos abastece a dos subestaciones, carecerían de sentido líneas que no tuvieran al menos dos subestaciones (principio y fin), por tanto las cardinalidades mínima y máxima de esta interrelación en el extremo de la entidad *SUBESTACIÓN* serán 2 y *n* respectivamente. Además cada subestación es abastecida por una única línea, lo que indica que las cardinalidades de la interrelación en el sentido contrario son siempre 1, y por lo tanto el tipo de correspondencia será 1:N.

Aunque no se nos dice nada al respecto, debemos suponer un atributo "Número de Subestación" como identificador principal de la entidad *SUBESTACIÓN*.

Las subestaciones distribuyen a una o varias zonas de servicio, lo que nos indica que debemos crear una interrelación **Distribuye** entre *SUBESTACIÓN* y *ZONA*, con tipo de correspondencia N:M, pues cada zona de servicio puede ser atendida por más de una subestación. Las cardinalidades mínimas y máximas de esta interrelación en cada uno de sus sentidos son respectivamente 1 y *n*.

De las zonas se desea almacenar el consumo medio y el número de consumidores finales, que pueden ser particulares, empresas e instituciones, por tanto este atributo tomará varios valores (17 instituciones, 400.000 particulares y 35 empresas, por ejemplo), por lo que lo se podría definir como multivaluado, sin embargo, dado que siempre tomará tres valores, una alternativa más simple es considerarlo como un atributo compuesto. Además se debe introducir un nuevo atributo "Cód_Zona" como identificador principal de la entidad *ZONA*.

Como último punto, el enunciado nos dice que las provincias están divididas en zonas de servicio, esto significa que debemos introducir una nueva interrelación **Se Divide** entre las entidades *ZONA* y *PROVINCIA*. Cada zona de servicio corresponde a una única provincia, pero cada provincia será dividida en varias zonas, por lo que el tipo de correspondencia será 1:N.

Todo lo analizado en esta parte aparece representado en la figura 1.21 (ver figura en la página anterior).

Propuesta de solución

La figura 1.22 agrupa las soluciones parciales anteriores.

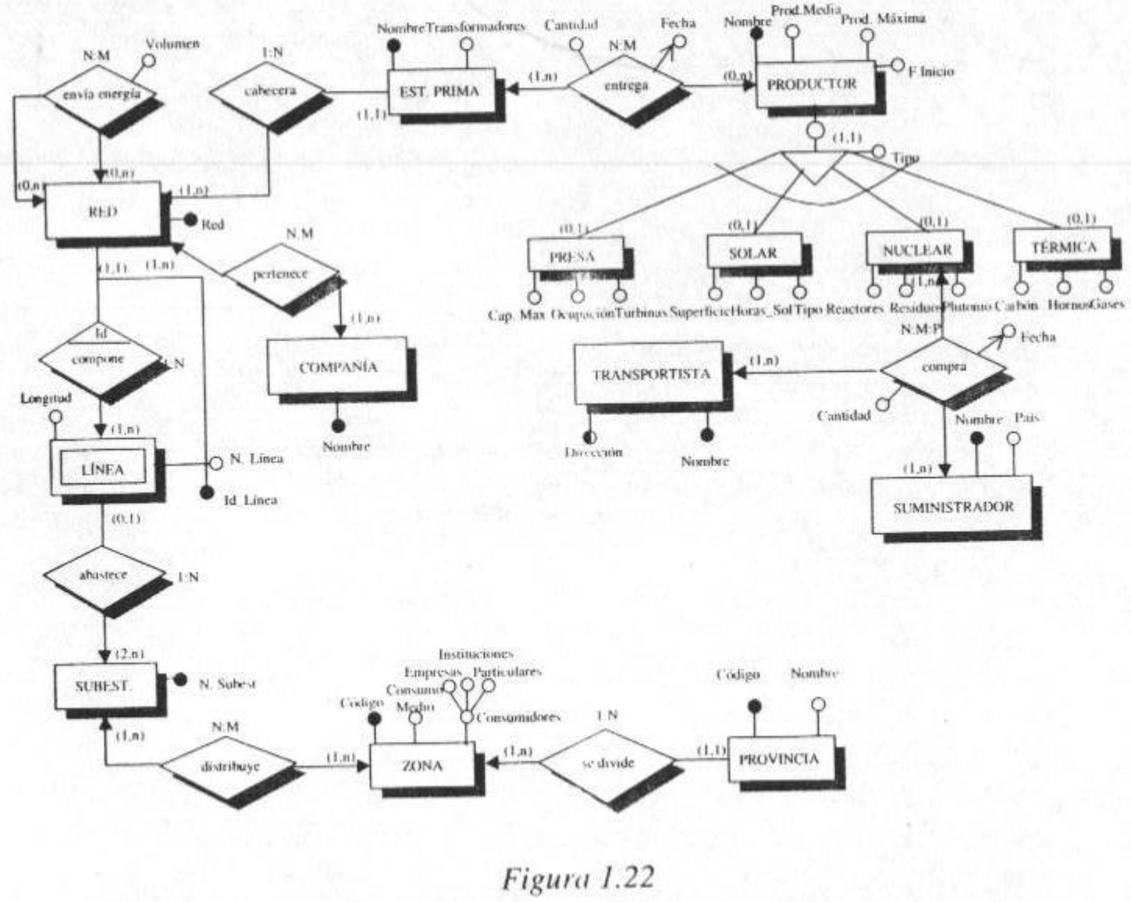


Figura 1.22

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha supuesto que una estación primaria puede no recibir energía de ningún productor.
- Se ha supuesto que el atributo "Transformadores" del tipo de entidad *ESTACIÓN PRIMARIA* es un atributo multivaluado que recoge todos los posibles tipos de transformadores de cada estación.
- Se ha supuesto que toda red debe tener como cabecera al menos a una estación primaria.
- Se ha supuesto que toda compañía eléctrica debe ser propietaria de al menos una red.
- Se ha supuesto que cada red debe estar compuesta de al menos una línea.

No se puede recoger el siguiente supuesto semántico:

- Que la energía sobrante de una red no se envíe a sí misma.

PROBLEMA 5: CONFLICTOS BÉLICOS

Enunciado

Una organización internacional pretende realizar un seguimiento de los conflictos bélicos que se producen en todo el mundo. Para ello creará una BD que responderá al siguiente análisis:

Se entiende por conflicto cualquier lucha armada que afecte a uno o varios países y en el cual se produzcan muertos y/o heridos. Todo conflicto se identificará por un nombre que habitualmente hará referencia a la zona o causa que provoca el conflicto, aunque dado que este nombre puede cambiar con el paso del tiempo, dentro de la BD cada conflicto se identificará mediante un código numérico sin significado alguno. Para cada conflicto se desea recoger los países a que afecta, así como el número de muertos y heridos contabilizados hasta el momento.

Los conflictos pueden ser de distintos tipos según la causa que lo ha originado, clasificándose, a lo sumo, en cuatro grupos: territoriales, religiosos, económicos o raciales, en cada uno de estos grupos se recogerán diversos datos. En los conflictos territoriales se recogerán las regiones afectadas, en los religiosos las religiones afectadas, en los económicos las materias primas disputadas y en los raciales las etnias enfrentadas.

En los conflictos intervienen diversos grupos armados (al menos dos) y diversas organizaciones mediadoras (podría no haber ninguna). Los mismos grupos armados y organizaciones mediadoras pueden intervenir en diferentes conflictos. Tanto los grupos armados como las organizaciones mediadoras podrán entrar y salir del conflicto, en ambos casos se recogerá tanto la fecha de incorporación como la fecha de salida. Temporalmente, tanto un grupo armado como una organización mediadora podrían no intervenir en conflicto alguno.

De cada grupo armado se recoge el código que se le asigna y un nombre. Cada grupo armado dispone de al menos una división y es liderado por al menos un líder político. Las divisiones de que dispone un grupo armado se numeran consecutivamente y se registra el número de barcos, tanques, aviones y hombres de que dispone, asimismo se recoge el número de bajas que ha tenido. Para los grupos armados se recoge el número de bajas como suma de las bajas producidas en todas sus divisiones.

Los traficantes de armas suministran diferentes tipos de arma a los grupos armados. De cada tipo de armas se recoge un nombre y un indicador de su capacidad destructiva. De cada traficante se recoge un nombre, los diferentes tipos de arma que puede suministrar y cantidad de armas de cada uno de los tipos de arma que podría suministrar. Se mantiene el número total de armas de cada uno de los diferentes tipos de armas suministrado por cada traficante a cada grupo armado.

Los líderes políticos se identifican por su nombre y por el código de grupo armado que lideran. Además se recoge una descripción textual de los apoyos que éste posee.

Cada división la pueden dirigir conjuntamente un máximo de tres jefes militares, aunque cada jefe militar no dirige más de una división. A cada jefe militar se le identifica por un código, además se recoge el rango que éste posee, y dado que un jefe militar no actúa por iniciativa propia sino que siempre obedece las órdenes de un único líder político de entre aquellos que lideran al grupo armado al que el jefe pertenece, se registrará el líder político al que obedece.

De las organizaciones mediadoras se recogerá su código, su nombre, su tipo (gubernamental, no gubernamental o internacional), la organización de que depende (una como máximo), el número de personas que mantiene desplegadas en cada conflicto y el tipo de ayuda que presta en cada conflicto que será de uno y sólo uno de los tres tipos siguientes, médica, diplomática o presencial.

Con diversos fines, los líderes políticos dialogan con las organizaciones; se desea recoger explícitamente esta información. Así para cada líder se recogerán aquellas organizaciones con que dialoga y viceversa.

Discusión del enunciado

PARTE 1

> *"Se entiende por conflicto cualquier lucha armada que afecte a uno o varios países y en el cual se produzcan muertos y/o heridos. Todo conflicto se identificará por un nombre que habitualmente hará referencia a la zona o causa que provoca el conflicto, aunque dado que este nombre puede cambiar con el paso del tiempo, dentro de la BD cada conflicto se identificará mediante un código numérico sin significado alguno. Para cada conflicto se desea recoger los países a que afecta, así como el número de muertos y heridos contabilizados hasta el momento."*

En el primer párrafo se define qué es un conflicto bélico, cómo se identifica y qué información se necesita guardar. Es bastante claro que *CONFLICTO* es una entidad que se identifica por un código numérico que llamaremos "Código", aunque en un principio podríamos pensar en el atributo "Nombre" como AIP. En el enunciado se nos aclara que se prefiere un código numérico para identificar cada conflicto, ya que así se asegura su invariabilidad, por lo que el "Nombre" sería identificador alternativo.

Como un conflicto puede afectar a varios países se podrían considerar dos posibilidades de modelado:

- Pensar en *PAÍS* como un tipo de entidad que se relaciona con *CONFLICTO* mediante la interrelación denominada **Afectar**.
- Representar "País" como un atributo multivaluado.

Si se siguen leyendo las especificaciones del problema se observará que no se vuelve a mencionar ninguna característica de país y por lo tanto se puede concluir que no es importante como entidad. Por lo que elegimos la segunda opción mencionada. Los números de "Muertos" y "Heridos" serán atributos obligatorios del tipo de entidad *CONFLICTO*, cuyo valor por defecto sería cero.

> *"Los conflictos pueden ser de distintos tipos según la causa que lo ha originado, clasificándose, a lo sumo, en cuatro grupos: territoriales, religiosos, económicos o raciales, en cada uno de estos grupos se recogerán diversos datos. En los conflictos territoriales se recogerán las regiones afectadas, en los religiosos las religiones afectadas, en los económicos las materias primas disputadas y en los raciales las etnias enfrentadas."*

En el segundo párrafo se presenta una clasificación de los conflictos en varios tipos, consideraremos esta clasificación como una jerarquía total y exclusiva. El supertipo será la entidad *CONFLICTO*, mientras que los subtipos serán las entidades *TERRITORIAL, RELIGIOSO, ECONÓMICO* y *RACIAL*. Estos subtipos tendrán los atributos o características de *CONFLICTO*, pero además, "Región", "Religión", "M.Prima" y "Etnia", respectivamente, que serán atributos multivaluados e indicarán en cada caso las regiones, religiones, materias primas o etnias afectadas en dicho conflicto.

El esquema de la figura 1.23 refleja las consideraciones anteriores.

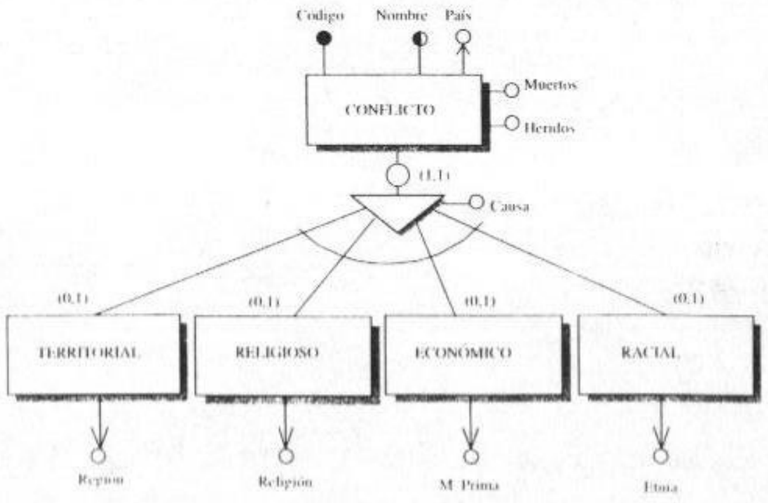


Figura 1.23

PARTE 2

> *"En los conflictos intervienen diversos grupos armados (al menos dos) y diversas organizaciones mediadoras (podría no haber ninguna). Los mismos grupos armados y organizaciones mediadoras pueden intervenir en diferentes conflictos. Tanto los grupos armados como las organizaciones mediadoras podrán entrar y salir del conflicto, en ambos caso se recogerá tanto la fecha de incorporación como la fecha de salida. Temporalmente, tanto un grupo armado como una organización mediadora podrían no intervenir en conflicto alguno."*

Como se indica en el texto, en los conflictos pueden intervenir, además de los grupos armados, organizaciones mediadoras, los cuales se considerarán como tipos de entidad que se interrelacionan con "Conflicto".

Cada una de las interrelaciones poseerá dos atributos multivaluados, "F.I.Cont", "F.F.Cont", "F.I.Media", "F.F.Media", que nos indican las fechas de entrada y de salida en los conflictos de los grupos armados y las organizaciones mediadoras, respectivamente. Se puede observar que las fechas de fin de las intervenciones, tanto de los grupos armados como de las organizaciones, son atributos opcionales, ya que puede darse el caso de no haber acabado con la intervención.

Para contemplar el hecho de que tanto un *GRUPO ARMADO* como una *ORGANIZACIÓN* pueden no participar en un conflicto de forma temporal, pondremos la cardinalidad mínima de ambas a cero. La cardinalidad máxima será *n* en las dos entidades.

También se indica en el texto que en un conflicto participan al menos dos grupos armados, con lo que la cardinalidad mínima de *CONFLICTO* en la interrelación **Interviene** es de 2 y la máxima de *n*.

Las cardinalidades de *CONFLICTO* en la interrelación **Media** son 0 para la mínima, ya que en las especificaciones se expone que puede haber conflictos en los que no participe ninguna organización, y *n* porque no existe limitación en el número de organizaciones que intervienen en un conflicto.

> *"De cada grupo armado se recoge el código que se le asigna y un nombre. Cada grupo armado dispone de al menos una división y es liderado por al menos un líder político. Las divisiones de que dispone un grupo armado se numeran consecutivamente y se registra el número de barcos, tanques, aviones y hombres de que dispone, asimismo se recoge el número de bajas que ha tenido. Para los grupos armados se recoge el número de bajas como suma de las bajas producidas en todas sus divisiones."*

El tipo de entidad *GRUPO ARMADO* se identificará por un "Código", el "Nombre", suponiendo que este es único, será un atributo alternativo y los líderes políticos se podrían considerar como un atributo multivaluado, pero si leemos el problema hasta

el final se comprobará que se necesita como tipo de entidad, ya que se relaciona con más entidades.

La cardinalidad mínima de *GRUPO ARMADO* en esta interrelación es 1, al tener al menos un líder político que lo lidera y su cardinalidad máxima es *n*, ya que no se tiene ninguna restricción acerca del número de lideres políticos asociados a un grupo armado en concreto. Las cardinalidades asociadas a *LÍDER POLÍTICO* podrían ser 1 la mínima y *n* la máxima si se considerara que sólo queremos en la BD aquellos líderes que se relacionen con algún grupo armado, como es el caso.

Por otro lado, detectamos la entidad débil *DIVISIÓN* que depende en identificación de *GRUPO ARMADO*, es decir, cada ejemplar de *DIVISIÓN* se identifica por el código del grupo armado al que pertenece y un número correlativo (que en el esquema E/R no podemos representar, ni siquiera con un dominio, pero que debemos especificar a efectos de una posterior implementación). Todos los atributos que posee la entidad *DIVISIÓN* son obligatorios.

Aunque aparezca el número de bajas por cada división, en el enunciado se especifica que también se desea conocer el número de bajas en total para cada grupo armado, y para reflejarlo consideraremos un atributo derivado (que se indica en la figura 1.24 con D1) en la entidad *GRUPO ARMADO*. La figura 1.24 muestra el esquema correspondiente a los supuestos anteriores.

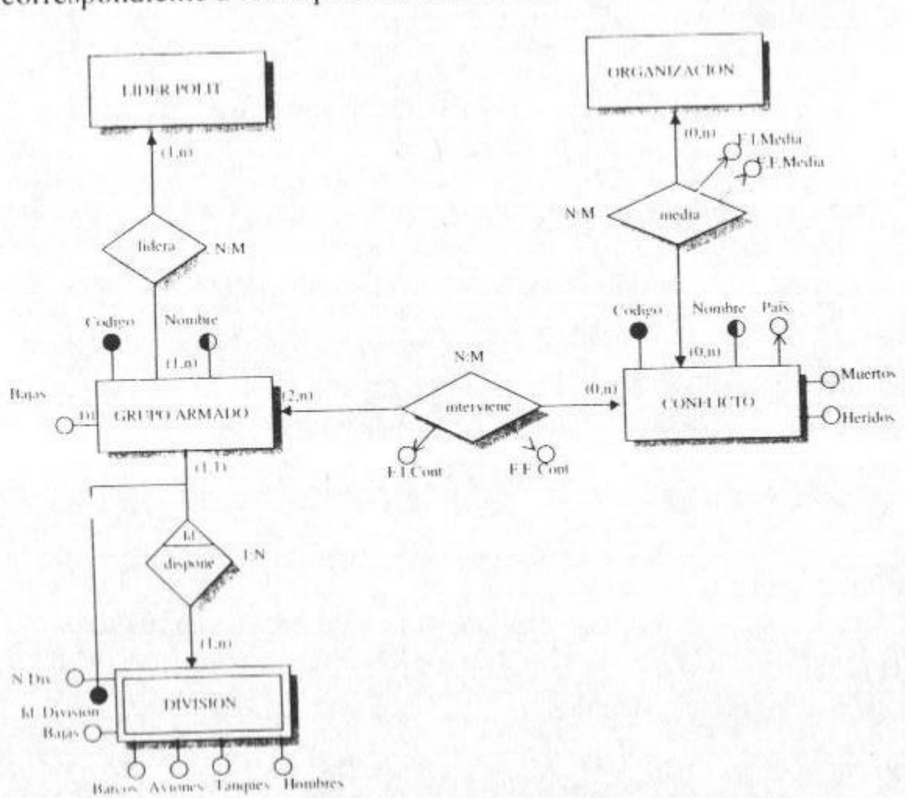


Figura 1.24

PARTE 3

"Los traficantes de armas suministran diferentes tipos de arma a los grupos armados. De cada tipo de armas se recoge un nombre y un indicador de su capacidad destructiva. De cada traficante se recoge un nombre, los diferentes tipos de arma que puede suministrar y cantidad de armas de cada uno de los tipos de arma que podría suministrar. Se mantiene el número total de armas de cada uno de los diferentes tipos de armas suministrado por cada traficante a cada grupo armado."

En la frase "Los traficantes de armas suministran diferentes tipos de arma a los grupos armados" se presenta una relación entre tres tipos de entidad, *TRAFICANTE*, *TIPO ARMA* y *GRUPO ARMADO*, por lo que tenemos una posible interrelación ternaria. Para hallar las cardinalidades fijamos dos entidades y vemos con cuántos ejemplares de la otra entidad está relacionado, por ejemplo, fijamos las entidades *GRUPO ARMADO* y *TIPO DE ARMA*, lo que significa que para un determinado grupo armado que posee un tipo de arma se la han podido suministrar 1 ó *n* traficantes. El resto de las cardinalidades se hallarían de forma análoga y también serían (1, n). Podría haber dudas con la cardinalidad siguiente: un traficante que tiene un tipo de arma puede suministrárselo a varios grupos armados (cardinalidad (0,n) que reflejaría las posibilidades de suministro), sin embargo, la anterior semántica se estará recogiendo con una interrelación binaria explícita **Puede Suministrar** que se analizará posteriormente. Se tiene por tanto que la semántica de la interrelación **Suministra** considera únicamente los suministros efectivos, por lo que su cardinalidad es (1,n).

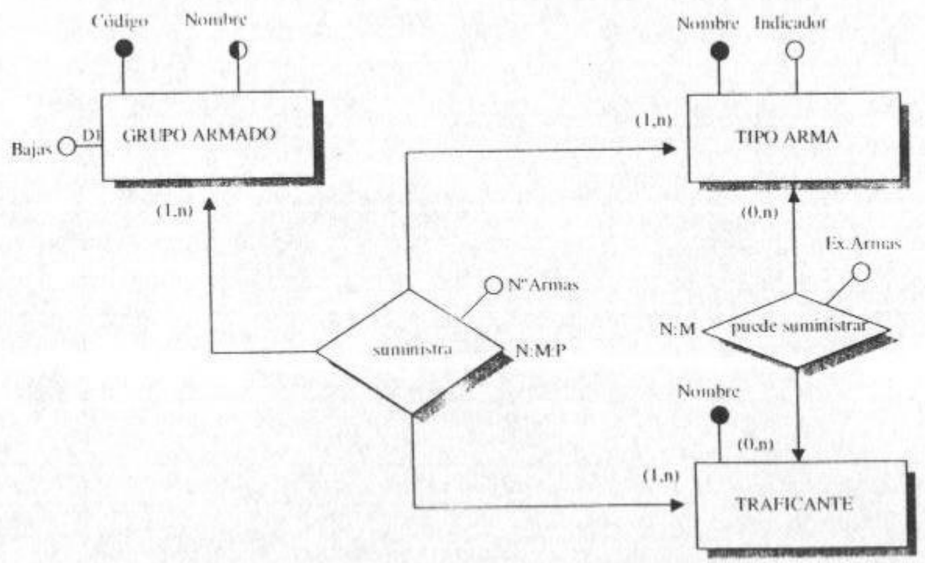


Figura 1.25

Esta interrelación ternaria tendrá un atributo, "Nº Armas", que indicará la cantidad de armas suministradas de cada tipo por un suministrador a los distintos grupos

armados. Este atributo nos da una pista sobre la no utilización de interrelaciones binarias en lugar de la anterior interrelación ternaria para considerar este supuesto, ya que no serían semánticamente equivalentes.

Como identificadores principales de las dos nuevas entidades, *TRAFICANTE* y *TIPO ARMA*, se tendrán los nombres, suponiendo que éstos son únicos. Además, en la entidad *TIPO ARMA* se tendrá otro atributo "Indicador" que informará de la capacidad destructiva de cada ejemplar de dicha entidad, y que se supone un valor obligatorio.

Para reflejar el tipo de armas que posee cada traficante y qué cantidad **Puede Suministrar** (es decir, su stock) se creará una interrelación binaria entre estos dos tipos de entidad y se indicará con un atributo, "Ex. Arma" en dicha interrelación, la información sobre el stock de cada traficante. La semántica anterior se refleja en la figura 1.25 (ver figura en la página actual).

PARTE 4

> *"Los líderes políticos se identifican por su nombre y por el código de grupo armado que lideran. Además se recoge una descripción textual de los apoyos que éste posee."*

Este párrafo es el que confirma la existencia del tipo de entidad *LÍDER POLÍTICO*. Pero además, se añade que este tipo de entidad es débil respecto a la entidad *GRUPO ARMADO*, por lo que se debe añadir al esquema. También se añadirá que la interrelación **Lidera** es en identificación.

> *"Cada división la pueden dirigir conjuntamente un máximo de tres jefes militares, aunque cada jefe militar no dirige más de una división. A cada jefe militar se le identifica por un código, además se recoge el rango que éste posee y dado que un jefe militar no actúa por iniciativa propia sino que siempre obedece las órdenes de un único líder político de entre aquellos que lideran al grupo armado al que el jefe pertenece, se registrará el líder político al que obedece."*

Aparece un nuevo tipo de entidad *JEFE MILITAR* y dos nuevas interrelaciones, una con *DIVISIÓN* para saber qué jefes militares dirigen cada una de las divisiones y otra con *LÍDER POLÍTICO* para conocer de quién recibe órdenes cada jefe militar.

La entidad *JEFE MILITAR* tiene como AIP un "Código" y un atributo obligatorio "Rango".

Las cardinalidades asociadas a *JEFE MILITAR* en la interrelación **Dirige**, que surge con la entidad *DIVISIÓN*, son (0,1) y en el sentido contrario de (1,3) ya que se especifica en el texto que una división la pueden dirigir como máximo tres jefes militares.

Siguiendo con el estudio de cardinalidades la interrelación **Obedece**, que asocia las entidades *JEFE MILITAR* y *LÍDER POLÍTICO*, tendremos (1,1) para *JEFE MILITAR*, ya que siempre obedece órdenes de un líder político y (0,n) para el otro tipo de entidad porque no todos los líderes políticos tienen el poder necesario para dar órdenes. La figura 1.26 muestra el esquema correspondiente a la semántica anterior.

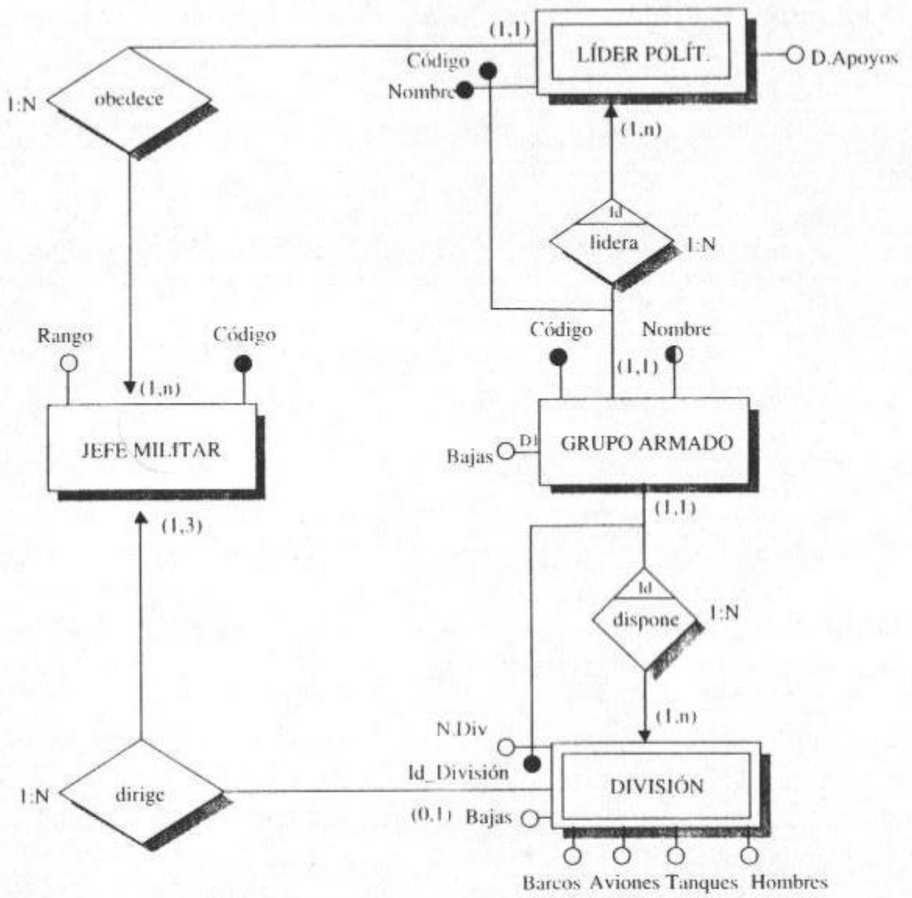


Figura 1.26

PARTE 5

> *"De las organizaciones mediadoras se recogerá su código, su nombre, su tipo (gubernamental, no gubernamental o internacional), la organización de que depende (una como máximo), el número de personas que mantiene desplegadas en cada conflicto y el tipo de ayuda que presta en cada conflicto que será de uno y sólo uno de los tres tipos siguientes, médica, diplomática o presencial."*

En este párrafo se indican los atributos de la entidad *ORGANIZACIÓN*, el AIP será "Código", un atributo alternativo "Nombre" y un atributo "Tipo" cuyos valores pueden ser gubernamental, no gubernamental o internacional. Este último atributo se podría haber pensado como una clasificación de *ORGANIZACIÓN*, es decir una

jerarquía, pero se puede prescindir de ella, ya que los subtipos ni tienen atributos propios ni se relacionan con otras entidades, con lo que se recoge la misma semántica definiendo un dominio para este atributo.

Como aparece en el texto que una organización puede depender de otra, se tendrá en el esquema una interrelación reflexiva, **Depende**, cuyas cardinalidades son (0,1) para representar este hecho y (0,n) para recoger que una organización puede representar a cero o varias organizaciones.

También se desea guardar información sobre las unidades que despliega en cada conflicto, que se representará como atributo de la interrelación **Media** al igual que el tipo de ayuda que se definirá dentro de un dominio cuyos valores serán los tipos de ayuda expuestos en el párrafo.

El que dentro de un conflicto una organización sólo pueda prestar un tipo de ayuda, simplifica la solución. Una hipótesis más general supondría bien tener que introducir un atributo multivaluado, bien añadir una nueva entidad y promocionar la interrelación **Media** a un grado superior a dos.

> *"Con diversos fines, los líderes políticos dialogan con las organizaciones; se desea recoger explícitamente esta información. Así para cada líder se recogerán aquellas organizaciones con que dialoga y viceversa."*

Para recoger que los líderes políticos pueden mantener conversaciones con las organizaciones mediadoras y viceversa, crearemos una nueva interrelación para completar el esquema, que se muestra en la figura 1.27.

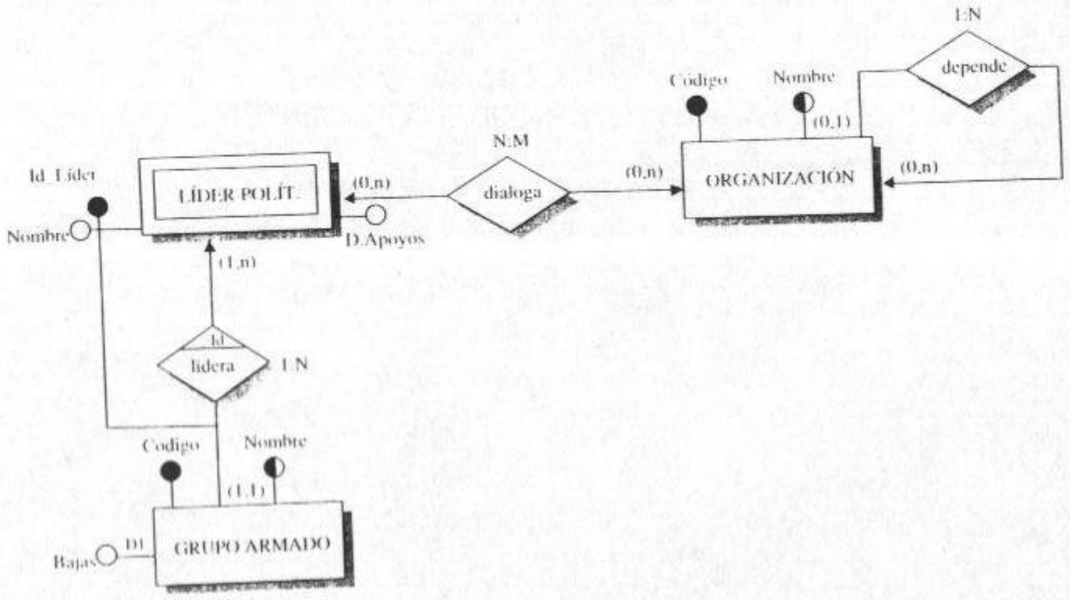


Figura 1.27

Propuesta de solución

Agrupamos las distintas soluciones parciales obtenidas en la figura 1.28.

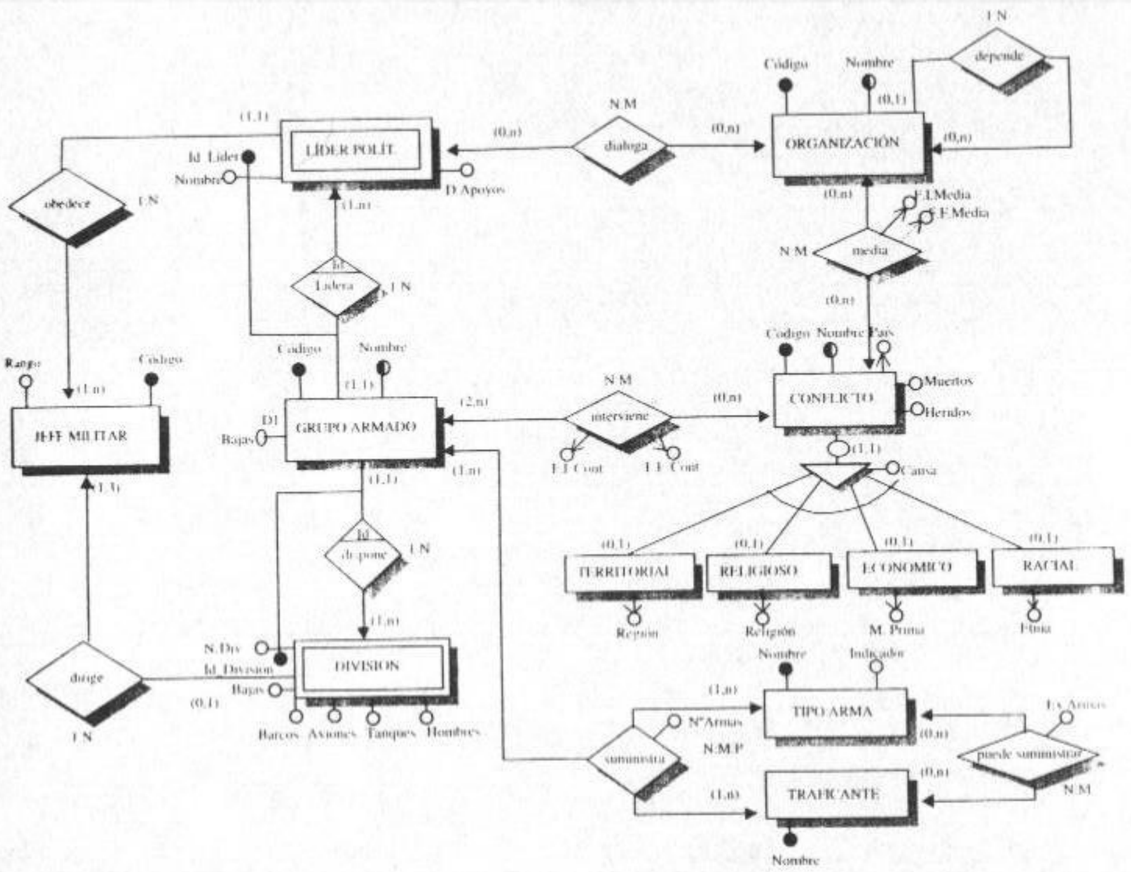


Figura 1.28

No es difícil determinar que los ciclos producidos no son semánticamente equivalentes, por lo que no se tiene redundancia.

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Secuencialidad del número de división de un grupo armado.

PROBLEMA 6: GESTIÓN DE NÓMINAS

Enunciado

Una empresa decide informatizar su nómina. Del resultado del análisis realizado, se obtienen las siguientes informaciones:

mento en que un empleado pierda su condición de perceptor de nómina, se minaría el correspondiente ejemplar de la entidad *EMPLEADO*. El inconveniente es se perdería la información de los empleados que ya no están en activo. Otras rnativas posibles son: incluir un atributo de estado (activo o no), considerar ambos os de empleados como parte de una jerarquía de empleados, o bien mantener una paralela con similar estructura que contenga la información referente a los pleados dados de baja.

La entidad candidata *MES* se ha deducido de la información que dictamina la iodicidad de recepción de los justificantes de nómina. Aún es prematuro establecer tratamiento de la dimensión temporal, pero la inclusión de una entidad para los ses parece una solución compleja y debería evitarse salvo que fuera estrictamente cesario. Aparentemente no aporta nada, y como se verá posteriormente, el mes es a de las propiedades de los *JUSTIFICANTES DE NÓMINA*, por lo que tal vez se te tan sólo de un atributo de esta entidad.

Se reconocen como entidades más firmes tan sólo *EMPLEADO* y *STIFICANTE DE NÓMINA*. De momento pueden considerarse como tales salvo e se oponga alguna información posterior.

En este párrafo también se intuye la interrelación existente entre las dos entidades tectadas que denominaremos **Recibe**. El hecho de que a un empleado se le entregue menos un justificante de nómina podría inducir a pensar que la cardinalidad mínima esta interrelación es uno, sin embargo, por conocimiento del dominio de discurso bemos que un empleado recién contratado no recibirá hasta finales de su primer mes trabajo ningún justificante de nómina, por lo que esta cardinalidad mínima es cero. n cuanto a la cardinalidad máxima, en principio no se afirma nada, pero suponiendo caso más general y apoyando con nuestro conocimiento del dominio de discurso ponemos que se trata de *n*. Las cardinalidades del otro extremo de la interrelación **ecibe** (1,1) se deducen también de nuestro conocimiento acerca del dominio de scurso. No parece lógico que varios empleados reciban el mismo justificante de ómina, ni que haya justificantes de nómina que no sean recibidos por ningún mpleado. Se tiene, por tanto, que un empleado puede recibir varios justificantes de ómina y que todo justificante de nómina lo recibe uno y sólo un empleado.

> *"A cada empleado se le asigna un número de matrícula en el momento de su incorporación a la empresa, y éste es el número usado a efectos internos de identificación. Además, se registran el NIF del empleado, nombre, número de hijos, porcentaje de retención para Hacienda, datos de cuenta corriente en la que se le ingresa el dinero (banco, sucursal y número de cuenta) y departamentos en los que trabaja. Un empleado puede trabajar en varios departamentos y en cada uno de ellos trabajará con una función distinta."*

Este párrafo describe básicamente propiedades acerca de los empleados, aunque ueden detectarse nuevas entidades candidatas, *DEPARTAMENTO*, *DATOS DE UENTA CORRIENTE* y *FUNCIÓN*.

"Datos de Cuenta Corriente" no se trata como entidad, pues aunque posee varios atributos (banco, sucursal y número de cuenta), todos juntos compondrían su único identificador principal. Se trata por tanto de un atributo compuesto a partir de los atributos "Banco", "Sucursal" y "Número de Cuenta".

Podría pensarse que "Departamento" se trata de un atributo multivaluado, sin embargo no es así, dado que mantendría una relación N:M con *EMPLEADO*, lo que no es posible, ya que los atributos multivaluados mantienen una relación 1:N con la entidad a la que pertenecen. Además, en el siguiente párrafo del enunciado, se dan propiedades adicionales de los departamentos, por lo que definitivamente se trata de una entidad con información propia.

DEPARTAMENTO tiene una interrelación N:M (**Trabaja**) con *EMPLEADO*; la "Función" no podría considerarse como una entidad propia sino como un atributo de esta interrelación, pues no posee más propiedades que el propio identificador. Si el mismo empleado pudiera trabajar con distintas funciones en el mismo departamento, habría de emplearse un atributo multivaluado, lo cual tendría implicaciones en la transformación al relacional, pues entonces, el atributo "Función" debería pasar a formar parte de la clave primaria de la correspondiente relación.

Para resolver las cardinalidades de la anterior interrelación precisamos considerar un supuesto semántico complementario. Se supondrá que puede haber departamentos sin empleados; por ejemplo, un departamento recién creado al cual aún no se le haya asignado personal, aunque no puede haber empleados sin departamento.

Son atributos de la entidad *EMPLEADO* el "Nº de Matrícula", el "Nombre", el "NIF", el "Nº de Hijos", los "Datos de Cuenta Corriente" y el "Porcentaje de Retención". El identificador de la entidad *EMPLEADO*, como se establece en el enunciado, es el atributo "Número de Matrícula", no el atributo "NIF", que aun siendo identificador de la entidad no es identificador principal, sino identificador alternativo.

> *"De un departamento se mantiene el nombre y cada una de sus posibles sedes."*

De este supuesto se obtienen los atributos "Nombre" y "Sede" de la entidad *DEPARTAMENTO*. La "Sede" se considera como un atributo multivaluado, pues admite varios valores y del enunciado no se desprende que tenga atributos propios y pueda considerarse por tanto una entidad propia.

> *"Son datos propios de un justificante de nómina el ingreso total percibido por el empleado y el descuento total aplicado. La distinción entre dos justificantes de nómina se hará, además de mediante el número de matrícula del empleado, mediante el ejercicio fiscal y número de mes al que pertenece y con un número de orden en el caso de varios justificantes de nómina recibidos el mismo mes."*

A cada empleado se le entregan múltiples justificantes de nómina a lo largo de su ida laboral en la empresa y al menos uno mensualmente.

A cada empleado se le asigna un número de matrícula en el momento de su ncorporación a la empresa, y éste es el número usado a efectos internos de dentificación. Además, se registran el NIF del empleado, nombre, número de hijos, orcentaje de retención para Hacienda, datos de cuenta corriente en la que se le ngresa el dinero (banco, sucursal y número de cuenta) y departamentos en los que rabaja. Un empleado puede trabajar en varios departamentos y en cada uno de ellos rabajará con una función distinta.

De un departamento se mantiene el nombre y cada una de sus posibles sedes.

Son datos propios de un justificante de nómina el ingreso total percibido por el mpleado y el descuento total aplicado. La distinción entre dos justificantes de nómina e hará, además de mediante el número de matrícula del empleado, mediante el jercicio fiscal y número de mes al que pertenece y con un número de orden en el caso le varios justificantes de nómina recibidos el mismo mes.

Cada justificante de nómina consta de varias líneas (al menos una de ingresos) y ada línea se identifica por un número de línea del correspondiente justificante. Una ínea puede corresponder a un ingreso o a un descuento. En ambos casos, se recoge la antidad que corresponde a la línea (en positivo si se trata de un ingreso o en negativo i se trata de un descuento); en el caso de los descuentos, se recoge la base sobre la ual se aplica y el porcentaje que se aplica para el cálculo de éstos.

Toda línea de ingreso de un justificante de nómina responde a un único concepto etributivo. En un mismo justificante, puede haber varias líneas que respondan al mismo concepto retributivo. De los conceptos retributivos se mantiene un código y una descripción.

De cara a la contabilidad de la empresa, cada línea de un justificante de nómina se imputa al menos a un elemento de coste. Al mismo elemento de coste pueden mputársele varias líneas. Para cada elemento de coste, se recoge un código, una descripción y un saldo.

Entre los elementos de coste se establece una jerarquía, en el sentido de que un elemento de coste puede contener a otros elementos de coste, pero un elemento de coste sólo puede estar contenido en, a lo sumo, otro elemento de coste.

En determinadas fechas, que se deben recoger, cada elemento de coste se liquida con cargo a varios apuntes contables (código y cantidad) y a una o varias transferencias bancarias, de las que se recogen los datos de cuenta corriente (banco, sucursal y número de cuenta) y la cantidad. Por cada apunte contable y transferencia bancaria se pueden liquidar varios elementos de coste.

Discusión del enunciado

PARTE 1

> *"A cada empleado se le entregan múltiples justificantes de nómina a lo largo de su vida laboral en la empresa y al menos uno mensualmente."*

En este primer párrafo pueden detectarse las siguientes entidades candidatas: *EMPLEADO, JUSTIFICANTE DE NÓMINA, VIDA LABORAL, EMPRESA* y *MES*, pero varias de ellas no llegarán a tener la consideración de entidades reales. Así, por diversos motivos que se explican a continuación, se desecharon *EMPRESA, VIDA LABORAL* y *MES*.

EMPRESA puede ser desechada, dado que lo que interesa a este Sistema de Información no es recoger empresas en general, sino recoger tan sólo los datos referentes a una y sólo una empresa (la propia empresa), por lo que en el caso de recoger una entidad *EMPRESA*, ésta contendría tan sólo un ejemplar que estaría describiendo a la propia empresa, lo cual carece de interés y tan sólo complicaría el esquema de datos resultante y las aplicaciones desarrolladas sobre el mismo. Es preferible suponer que toda la información presente en la BD está referida a la propia empresa y omitir esta entidad. Dicho de otra manera, la empresa en que se desarrolla el Sistema de Información es parte del contexto del mismo.

VIDA LABORAL no se trata realmente de una entidad; parece que aunque el propósito de este Sistema de Información sea recoger, al menos en parte, vidas laborales, este concepto no está claramente definido y admite diversas interpretaciones. ¿Qué es una vida laboral? Desde una perspectiva amplia, una vida laboral sería todo aquello que le acaece a un empleado, pero no interesa recoger todo lo que le sucede a un empleado, tan sólo aquello que resulta de interés para los propósitos del Sistema de Información (la gestión de las nóminas). Además de esta consideración como entidad, vida laboral puede interpretarse como información de contexto, de esta forma, el concepto de vida laboral serviría simplemente para delimitar el marco temporal en que un empleado puede recibir justificantes de nómina.

El establecimiento de este marco temporal induciría a restricciones sobre la información admisible en la extensión del esquema. Si nos fijamos en este contexto temporal, ¿qué sucederá cuando un empleado se jubile y finalice su vida laboral?, posiblemente cese su percepción de nóminas, lo cual nos llevaría a replantearnos el significado de la entidad *EMPLEADO*, ¿Qué va a contener esa entidad?, ¿empleados en general o solamente empleados perceptores de nóminas? Todas estas cuestiones y matices semánticos habrían de aclararse revisando la información disponible del universo de discurso y los requerimientos del Sistema de Información, realizando posiblemente un análisis más profundo del presentado en el enunciado de este problema. Por simplificar, en la solución propuesta, se considera que la entidad *EMPLEADO* contiene sólo empleados perceptores de nómina; por tanto, en el

En este apartado se introducen los atributos "Ingreso Total", "Descuento Total", "Mes" y "Número de Orden" de la entidad *JUSTIFICANTE DE NÓMINA*.

Dado que la identificación de un justificante de nómina se realiza a partir del número de matrícula del empleado que la recibe, la interrelación **Recibe** es en identificación y por tanto la entidad *JUSTIFICANTE DE NÓMINA* es débil y su identificador principal engloba a todos los atributos del justificante que se emplean como parte de la identificación: "Ejercicio", "Mes" y "Número de Orden" junto con el "Número de Matrícula" del empleado.

El atributo "Ejercicio" permite la distinción del mes a lo largo de diversos años, en caso de que el enunciado no introdujera este atributo se requeriría establecer una manera de diferenciar los meses a lo largo de los años; entonces, o bien se incluye este atributo o bien se establece algún convenio, como enumerar los meses consecutivamente a partir de un determinado año (por ejemplo, 1900). La inclusión del atributo "Ejercicio" permite, además, diferenciar el número del mes dentro del año.

El resultado de todas las consideraciones realizadas en esta parte se presenta en el esquema mostrado en la figura 1.29.

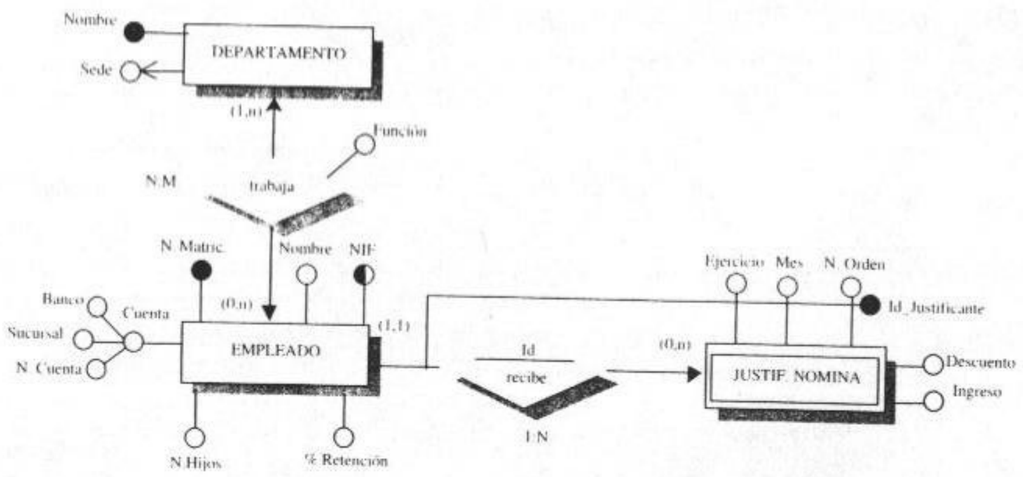


Figura 1.29

PARTE 2

"Cada justificante de nómina consta de varias líneas (al menos una de ingresos) y cada línea se identifica por un número de línea del correspondiente justificante. Una línea puede corresponder a un ingreso o a un descuento. En ambos casos, se recoge la cantidad que corresponde a la línea (en positivo si se trata de un ingreso o en negativo si se trata de un descuento); en el caso de los descuentos, se recoge la base sobre la cual se aplica y el porcentaje que se aplica para el cálculo de éstos."

Se puede distinguir una nueva entidad *LÍNEA DE NÓMINA* con un atributo "Nº de Línea" y otro atributo "Cantidad". Las líneas de nóminas serán de dos clases, ingresos y descuentos, y podría pensarse en diferenciar estas clases mediante un atributo que indicara la clase de línea de que se trata; sin embargo, dado que en el caso de los descuentos se tendrían las propiedades "Base" y "Porcentaje", es preferible introducir una jerarquía y considerar a estas clases como subtipos de *LÍNEA DE NÓMINA*.

La interrelación **Consta** entre las entidades *NÓMINA* y *LÍNEA DE NÓMINA* es similar a la que se da entre una factura y sus líneas de factura o entre un pedido y sus líneas de pedido, por ejemplo, se trata de una agregación todo-parte. En esta interrelación se tiene una dependencia en identificación, por lo que la entidad *LÍNEA DE NÓMINA* es débil respecto a la entidad *JUSTIFICANTE DE NÓMINA* y su identificador principal estará compuesto por el identificador principal de *JUSTIFICANTE DE NÓMINA* junto al atributo "Número de Línea" (que tiene numeración consecutiva dentro del mismo justificante de nómina).

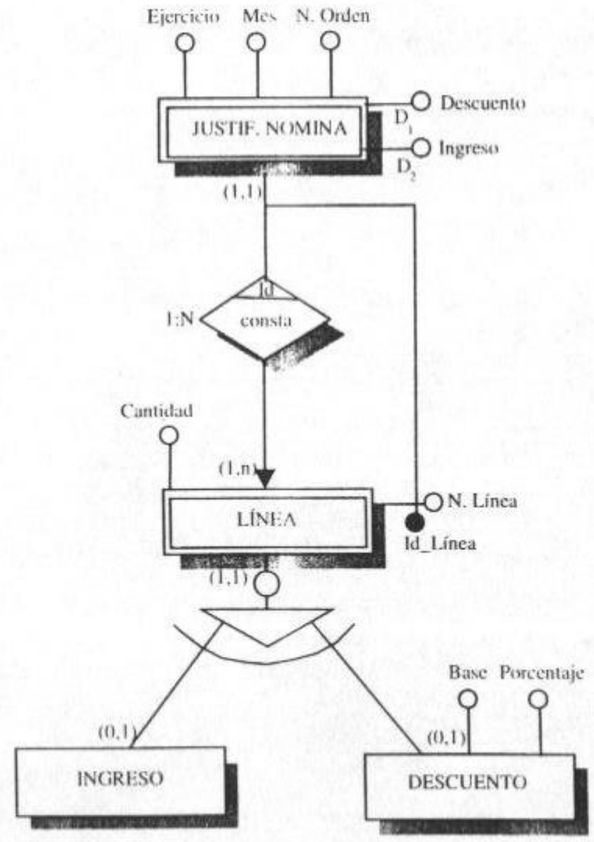


Figura 1.30

Dada la anterior interrelación, puede señalarse ahora que los atributos "Ingreso Total" y "Descuento Total" en la entidad *JUSTIFICANTE DE NÓMINA* son de-

rivados, pues su cálculo puede realizarse a partir de la cantidad que figura en las líneas que componen cada justificante.

La jerarquía existente en la entidad *LÍNEA DE NÓMINA* es total, dado que toda línea es bien una línea de ingreso o bien una de descuento, y exclusiva, dado que sólo puede ser de uno de los dos tipos.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.30 (ver figura en la página anterior).

PARTE 3

> *"Toda línea de ingreso de un justificante de nómina responde a un único concepto retributivo. En un mismo justificante, puede haber varias líneas que respondan al mismo concepto retributivo. De los conceptos retributivos se mantiene un código y una descripción."*

Se distingue una nueva entidad *CONCEPTO RETRIBUTIVO*, con atributos "Código" y "Descripción"; podría haberse considerado como atributo multivaluado si no dispusiera de atributos propios. Se supone el atributo "Código" como el identificador principal de esta entidad.

El subtipo *LÍNEAS DE INGRESO* tiene una interrelación 1:N (**Responde**) con la entidad *CONCEPTO RETRIBUTIVO* que no tiene la entidad *LÍNEAS DE DESCUENTO*, lo que apoya la decisión de reflejar la jerarquía de líneas de manera explícita.

> *"De cara a la contabilidad de la empresa, cada línea de un justificante de nómina se imputa al menos a un elemento de coste. Al mismo elemento de coste pueden imputársele varias líneas. Para cada elemento de coste, se recoge un código, una descripción y un saldo."*

Surge aquí una nueva entidad *ELEMENTO DE COSTE* con atributos "Código", "Descripción" y "Saldo". Se supone que el atributo "Código" es el identificador principal.

Se tiene una interrelación **Imputa** entre las entidades *ELEMENTO DE COSTE* y *LÍNEA* (esta interrelación afecta a ambos subtipos de la jerarquía de líneas, por lo que se establece sobre el supertipo). Se trata de una relación N:M en la que se fuerza una cardinalidad mínima uno en el extremo del elemento de coste, por ejemplo toda línea se imputa al menos a un elemento de coste.

> *"Entre los elementos de coste se establece una jerarquía, en el sentido de que un elemento de coste puede contener a otros elementos de coste, pero un elemento de coste sólo puede estar contenido en, a lo sumo, otro elemento de coste."*

La jerarquía que el enunciado señala que existe entre los elementos de coste, se trata en realidad de una interrelación reflexiva en esta entidad. Refleja el típico caso de

una estructura arborescente propia de las jerarquías implosión-explosión. El tipo de la interrelación es 1:N y las cardinalidades son (0,1) y (0,n), es decir, un hijo sólo puede tener un padre, mientras que un padre puede tener varios hijos. Se supone que un elemento de coste no puede contener a ningún antecesor suyo, pues en otro caso no se tendrá un arbol, sino un grafo.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.31.

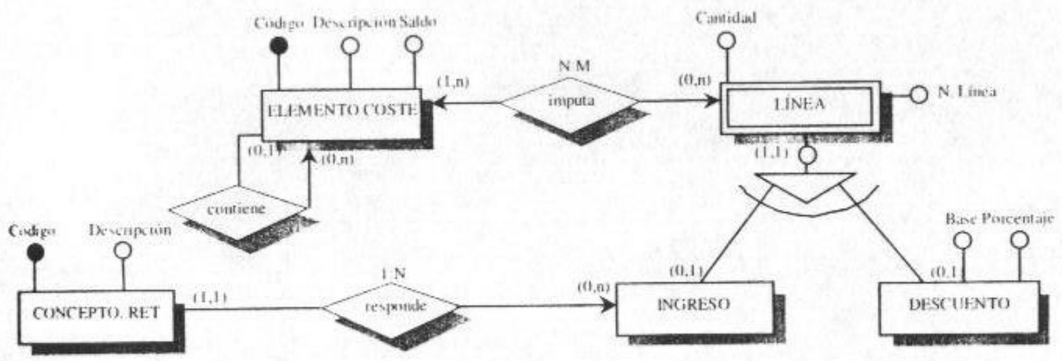


Figura 1.31

PARTE 4

> *"En determinadas fechas, que se deben recoger, cada elemento de coste se liquida con cargo a varios apuntes contables (código y cantidad) y a una o varias transferencias bancarias, de las que se recogen los datos de cuenta corriente (banco, sucursal y número de cuenta) y la cantidad. Por cada apunte contable y transferencia bancaria se pueden liquidar varios elementos de coste."*

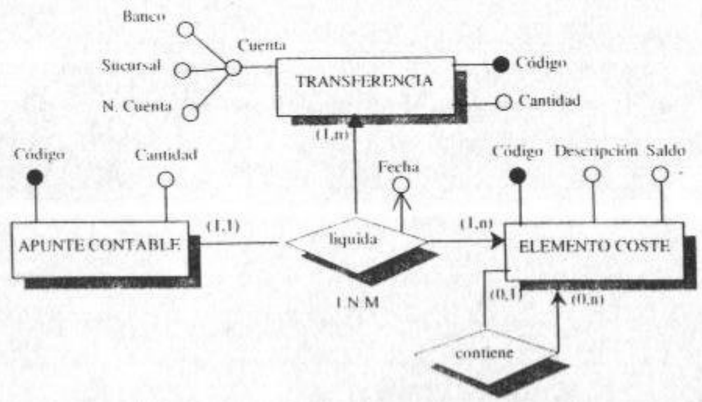


Figura 1.32

Se obtiene una nueva entidad *TRANSFERENCIA*, con atributos "Código" (se precisa para diferenciar las distintas transferencias entre sí, por lo que se considerará como identificador principal), "Cantidad" y "Datos de Cuenta Corriente" que a su vez

es un atributo compuesto de "Banco", "Sucursal" y "Número de Cuenta". Además, existe la entidad *APUNTE CONTABLE* con los atributos "Código" y "Cantidad", en la que se supondrá que el atributo "Código" es el identificador principal.

La relación existente entre las entidades *TRANSFERENCIA, APUNTE CONTABLE* y *ELEMENTO DE COSTE*, se modelará con una interrelación ternaria **Liquida** que posee además un atributo propio "Fecha", el cual es multivaluado y permitirá distinguir las liquidaciones que afecten al mismo elemento de coste, transferencia y apunte contable. Nótese cómo a partir del conocimiento del dominio de discurso, se deduce que la cardinalidad máxima en apunte contable es uno, por ejemplo cada elemento de coste y transferencia que se liquidan se hace con cargo a un único apunte contable.

No sería posible modelar esta interrelación ternaria como relaciones binarias, dada la existencia del atributo "Fecha" que afecta por igual a los ejemplares que se combinan de las distintas entidades.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.32 (ver figura en la página anterior).

Propuesta de solución

Agrupamos las distintas soluciones parciales y se obtiene el esquema E/R de la figura 1.33.

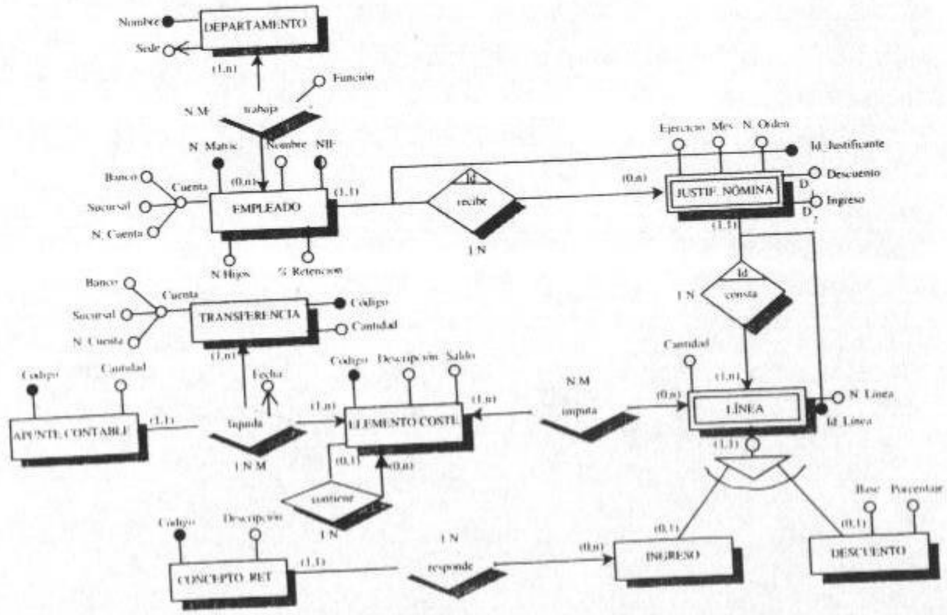


Figura 1.33

En la solución obtenida, no es preciso realizar análisis de redundancias, pues no se han producido ciclos.

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha supuesto que puede haber departamentos sin empleados, por ejemplo, un departamento recién creado al cual aún no se le haya asignado personal.
- Se ha supuesto que un elemento de coste no puede contener ningún antecesor suyo.

No se pueden recoger en el esquema E/R los siguientes supuestos semánticos:

- Un empleado no puede recibir nóminas tras la extinción de su vida laboral.
- El que cada empleado reciba al menos un justificante de nómina cada mes.
- El que haya al menos una línea de ingresos en cada justificante.
- Secuencialidad del número de orden del justificante.
- Secuencialidad del número de línea del justificante.
- La cantidad en las líneas de ingreso se expresa como un número positivo y en las líneas de descuento como un número negativo.

PROBLEMA 7: ENTORNO DE EJECUCIÓN

Enunciado

Una empresa decide crear un único entorno de ejecución que controle la seguridad de acceso para todas sus aplicaciones informáticas. Para ello considera conveniente dividir sus aplicaciones en subsistemas funcionales especializados y establecer el control de acceso al nivel de estos subsistemas. Se desarrollará un motor de ejecución que, tomando como parámetros los contenidos de la BD, controlará la ejecución de los subsistemas y el acceso a los mismos. Este motor se hará cargo también de la navegación dentro de los subsistemas. Profundizando en este enfoque, se establecen los siguientes requisitos:

La unidad básica de acceso a los subsistemas es el denominado perfil de acceso. Un usuario tendrá acceso a todos los subsistemas a los que permiten acceder los distintos perfiles de que disfruta (al menos uno). Un perfil permite el acceso de al menos un subsistema y para cualquier subsistema habrá siempre un perfil que permita acceder al mismo.

De cada usuario se mantiene el DNI, nombre, teléfono y terminales en que trabaja.

De los perfiles de acceso, lo mismo que de los subsistemas, se mantiene un código y una descripción. De los subsistemas, se mantiene, además, la ventana en la que arranca.

Las ventanas están compuestas por controles; toda ventana tendrá un control que permita cerrarla. Todo control ha de emplearse en alguna ventana y el mismo control puede ser empleado en distintas ventanas. De las ventanas y controles se mantiene también un código y una descripción.

Los controles pueden ser de dos tipos: botones o ítems de menú. Para soportar la estructura jerárquica de menús, de un ítem de menú pueden depender otros ítems, pero no puede darse la situación de que el mismo ítem dependa de varios ítems. En los ítems de menú se ha de mantener forzosamente el texto que se visualizará en pantalla. De los controles de tipo botón se mantiene el nombre del icono que opcionalmente se visualizará.

La activación de un control tiene como consecuencia la ejecución de una única acción (todo control ejecutará una acción al menos). Una acción requiere siempre un control que pueda ejecutarla. De las acciones se mantiene el código y la descripción.

Las acciones pueden ser de dos tipos, de función y de llamada. Las acciones de función ejecutan una función interna del propio entorno (de la que se ha de guardar el nombre). Las acciones de llamada invocan una única ventana.

Discusión del enunciado

PARTE 1

> *"La unidad básica de acceso a los subsistemas es el denominado perfil de acceso. Un usuario tendrá acceso a todos los subsistemas a los que permiten acceder los distintos perfiles de que disfruta (al menos uno). Un perfil permite el acceso de al menos un subsistema y para cualquier subsistema habrá siempre un perfil que permita acceder al mismo."*

En este párrafo pueden detectarse las siguientes entidades candidatas: *SUBSISTEMA*, *PERFIL* y *USUARIO*. Al mismo tiempo, se está introduciendo una interrelación **Accede** entre *PERFIL* y *SUBSISTEMA*, de la cual se están proporcionando las cardinalidades mínimas, pero no las máximas (por lo que se ha de suponer el caso más general). Además, se tiene otra interrelación **Disfruta** entre *USUARIO* y *PERFIL* de la cual se dan las cardinalidades del usuario respecto de los perfiles; en el sentido contrario, habremos de realizar la suposición de que varios usuarios pueden compartir perfiles (por generalidad y conocimiento del universo de discurso) y que puede haber

perfiles que no tengan usuarios (por generalidad). Por tanto, las entidades candidatas *USUARIO*, *PERFIL* y *SUBSISTEMA* son entidades reales.

En una lectura alternativa de este primer párrafo, podría haberse pensado que aparecían las tres entidades anteriores y entre éstas una sola interrelación ternaria **Accede**. Esta alternativa, sin embargo, puede desecharse, ya que la relación entre *USUARIO* y *SUBSISTEMA* no es directa, sino que tiene a *PERFIL* como elemento intermedio, por ejemplo el usuario accede a aquellos subsistemas cuando se lo permite el(los) perfil(es) de que disfruta.

> *"De cada usuario se mantiene el DNI, nombre, teléfono y terminales en que trabaja."*

De este párrafo se obtienen los atributos de la entidad *USUARIO*, se decide emplear el "DNI" como identificador principal, mientras que el atributo "Teléfono" puede considerarse como un identificador alternativo. Se considerará "Terminales" como un atributo multivaluado.

> *"De los perfiles de acceso, lo mismo que de los subsistemas, se mantiene un código y una descripción. De los subsistemas, se mantiene, además, la ventana en la que arranca."*

En este párrafo se están dando los atributos de las entidades *PERFIL* y *SUBSISTEMA*; ambas tienen como atributos un "Código" y una "Descripción" y se supone en ambas que el "Código" es el identificador principal. La entidad *SUBSISTEMA* tendría además un atributo "Ventana de Arranque" que serviría para describir la ventana en que arranca cada subsistema.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.34.

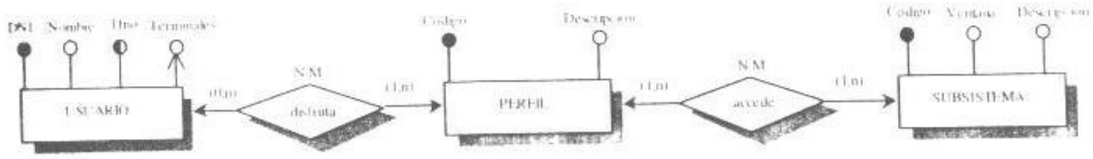


Figura 1.34

PARTE 2

> *"Las ventanas están compuestas por controles; toda ventana tendrá un control que permita cerrarla. Todo control ha de emplearse en alguna ventana y el mismo control puede ser empleado en distintas ventanas. De las ventanas y controles se mantiene también un código y una descripción."*

Se pueden distinguir dos nuevas entidades *VENTANA* y *CONTROL;* las dos entidades tienen los mismos atributos, "Código" y "Descripción" y se entiende que en ambas entidades "Código" es el identificador principal.

Se tiene una interrelación **Compuesta** entre estas entidades cuyas cardinalidades se deducen del enunciado; en particular, el hecho de que toda ventana tiene un control que permite cerrarla nos está indicando que cualquier ventana tiene como mínimo un botón, precisamente el que permite cerrarla. En este caso, el sentido de la frase no pretende que haya diferentes tipos de controles; como se verá más adelante, los controles se podrán distinguir entre sí por la acción que realizan (entre otras cosas cerrar la ventana), pero no procederá introducir un tipo especial de control para cualquier acción que éstos puedan realizar. Si se hubiese introducido el subtipo de los controles que cierran una ventana, al darnos cuenta de lo anterior, debería eliminarse.

Al aparecer *VENTANA* como una entidad, se debería replantear la necesidad del atributo "Ventana de Arranque" que aparece en la entidad *SUBSISTEMA* y preguntarse si no se trata realmente de una interrelación entre estas entidades. Efectivamente, todo subsistema debería arrancar en una ventana, y la misma ventana podría emplearse como punto de arranque de diversos subsistemas (no se afirma nada en contra). Por ello, se introduce la interrelación **Arranca.**

> *"Los controles pueden ser de dos tipos: botones o ítems de menú. Para soportar la estructura jerárquica de menús, de un ítem de menú pueden depender otros ítems, pero no puede darse la situación de que el mismo ítem dependa de varios ítems. En los ítems de menú se ha de mantener forzosamente el texto que se visualizará en pantalla. De los controles de tipo botón se mantiene el nombre del icono que opcionalmente se visualizará."*

Se introducen aquí dos clases de controles, los botones y los ítems de menú, que requieren respectivamente las entidades *BOTÓN* e *ÍTEM*. Los ítems tienen a "Texto" como atributo específico, mientras que los botones tienen el atributo "Icono" que contiene el nombre del fichero con la imagen que opcionalmente se visualizará, hay que destacar que este atributo admite nulos, pues aquellos botones que no requieran que se visualice un icono no tendrán un valor en este atributo.

Estas entidades se relacionan con la entidad *CONTROL* mediante una interrelación **Es-Un,** por ejemplo se trata de una jerarquía que es total (no hay otros tipos de controles) y exclusiva (un ítem de menú no es un botón ni viceversa).

Nuevamente aparece aquí una estructura jerárquica arborescente en el caso de ítems de menú. Para recoger la misma se emplea una interrelación reflexiva **Depende;** el tipo de esta interrelación es 1:N y las cardinalidades son (0,1) y (0,n), es decir, un hijo sólo puede tener un padre, mientras que un padre puede tener varios hijos. Para evitar ciclos, es preciso exigir que un ítem de menú no pueda depender de sí mismo ni de ningún antecesor suyo.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.35.

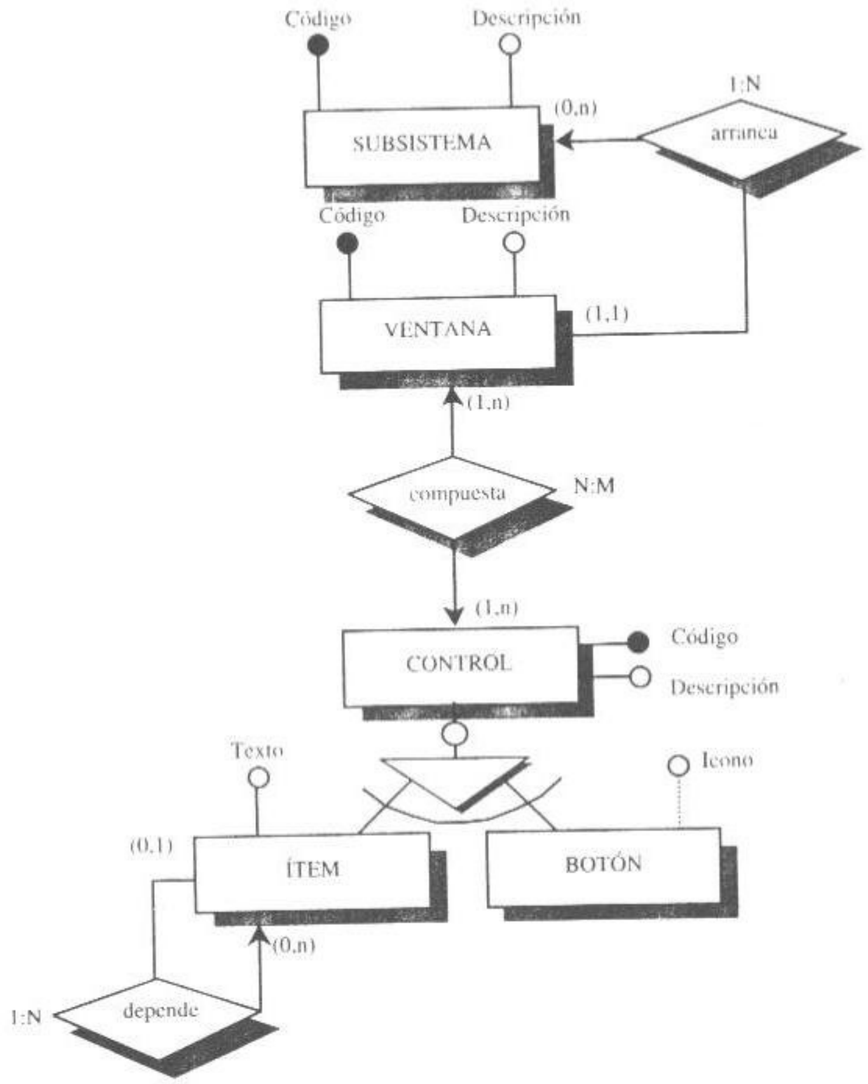


Figura 1.35.

PARTE 3

> *"La activación de un control tiene como consecuencia la ejecución de una única acción (todo control ejecutará una acción al menos). Una acción requiere siempre un control que pueda ejecutarla. De las acciones se mantiene el código y la descripción."*

Se tiene una nueva entidad *ACCIÓN* con atributos "Código" y "Descripción". Nuevamente, el identificador principal será "Código". Se tiene, además, una nueva interrelación 1:N **Ejecuta** que relaciona las acciones con los controles que las ejecutan. Se tiene cardinalidad mínima 1 en ambos extremos de la interrelación.

> *"Las acciones pueden ser de dos tipos, de función y de llamada. Las acciones de función ejecutan una función interna del propio entorno (de la que se ha de guardar el nombre). Las acciones de llamada invocan una única ventana."*

Se introduce aquí, nuevamente, una jerarquía con los subtipos *LLAMADA* y *FUNCIÓN* del supertipo *ACCIÓN;* esta jerarquía es total y exclusiva. Las funciones tendrán un atributo "Nombre" que puede considerarse como identificador alternativo de la entidad *FUNCIÓN.* Por el contrario, las llamadas no tienen atributos propios y, sin embargo, tienen una interrelación 1:N **Invoca** con ventana que permite que en cada acción de llamada se refleje la ventana que ésta invoca.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.36.

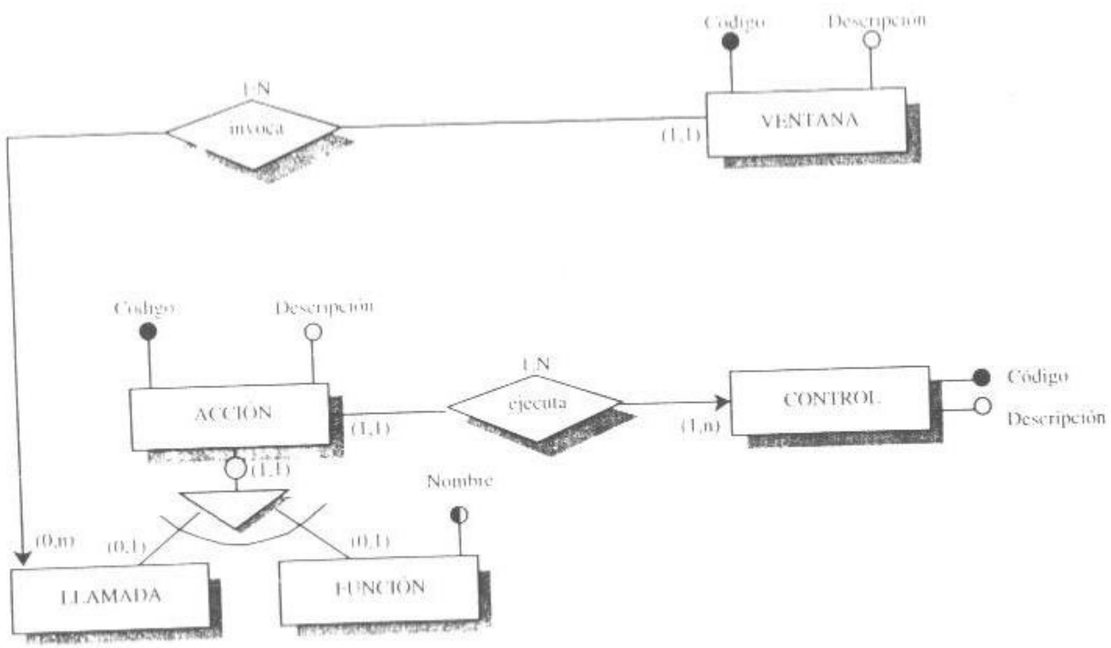


Figura 1.36

Propuesta de solución

Agrupando las distintas soluciones parciales se obtiene el esquema E/R mostrado en la figura 1.37.

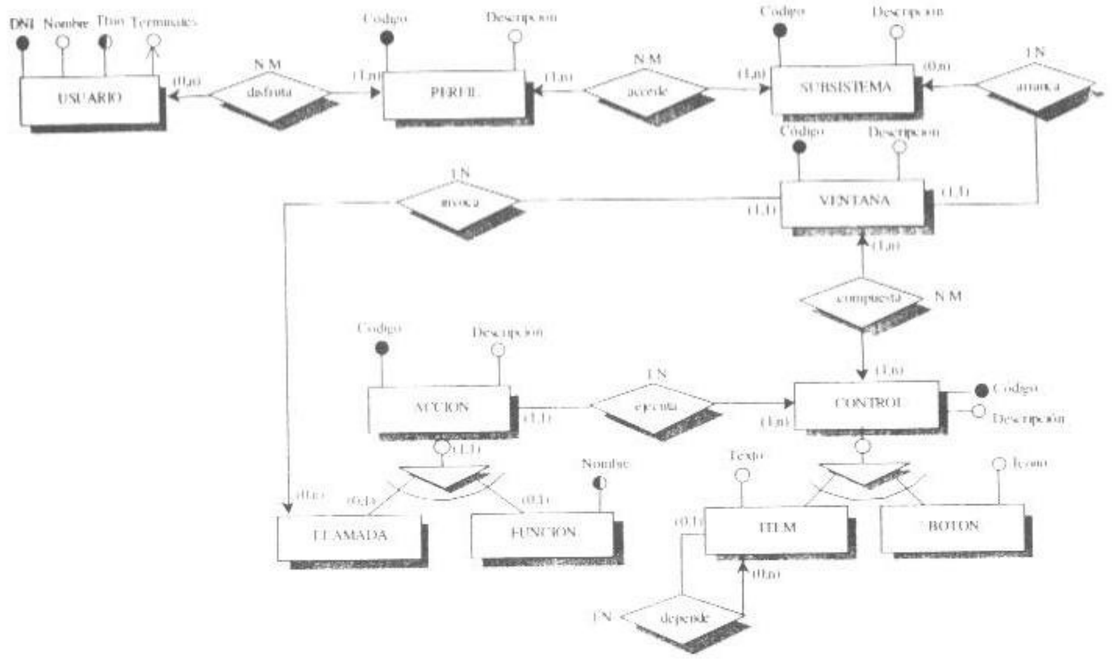


Figura 1.37

Aunque existe un ciclo en el esquema E/R entre **Invoca, Compuesta** y **Ejecuta,** estas interrelaciones no son semánticamente equivalentes , lo que significa que no existe redundancia en el esquema. La semántica representada por la interrelación **Invoca** es diferente de la representada por **Compuesta** y **Ejecuta,** la de **Compuesta** diferente a la de **Invoca** y **Ejecuta,** y la de **Ejecuta** diferente a la de **Invoca** y **Compuesta.**

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se supone que un ítem de menú no puede depender de él mismo ni de ningún antecesor suyo.
- Se ha supuesto que un perfil permita el acceso a varios subsistemas y que a un subsistema puedan acceder varios perfiles.
- Se ha supuesto que varios usuarios puedan tener el mismo perfil de acceso.
- Se ha supuesto que varios subsistemas puedan arrancar en la misma ventana.
- Se ha supuesto que la misma ventana pueda ser invocada por varias acciones de llamada.

No se ha podido recoger en el esquema E/R el siguiente supuesto semántico:

- Cualquier ventana tiene entre sus controles uno cuya acción asociada es la correspondiente acción de cerrarla.

PROBLEMA 8: ORGANIZACIONES NO GUBERNAMENTALES

Enunciado

La coordinadora nacional de Organizaciones No Gubernamentales (ONG's) desea mantener una base de datos de las asociaciones de este tipo que existen en nuestro país. Para ello necesita almacenar información sobre cada asociación, los socios que las componen, los proyectos que realizan y los trabajadores de las mismas.

De las asociaciones se desea almacenar su CIF, denominación, dirección y provincia, su tipo (ecologista, integración, desarrollo...), así como si está declarada de utilidad pública por el Ministerio del Interior.

Cada asociación está formada por socios de los que se precisa conocer su DNI, nombre, dirección, provincia, fecha de alta en la asociación, la cuota mensual con que colaboran y la aportación anual que realizan (que se obtendrá multiplicando la cuota mensual por los meses del año).

Los trabajadores de estas organizaciones pueden ser de dos tipos: asalariados y voluntarios.

Los asalariados son trabajadores que cobran un sueldo y ocupan cierto cargo en la asociación. Se desea almacenar la cantidad que éstos pagan a la seguridad social y el tanto por ciento de IRPF que se les descuenta.

Los voluntarios trabajan en la organización desinteresadamente, siendo preciso conocer su edad, profesión y las horas que dedican a la asociación a efectos de cálculo de estadísticas.

Cada trabajador se identifica por su DNI, tiene un nombre y una fecha de ingreso.

Un socio no puede ser trabajador de la asociación.

Las asociaciones llevan a cabo proyectos a los que están asignados sus trabajadores. Un trabajador puede trabajar en diferentes proyectos de un mismo país. De cada proyecto se desea almacenar su número de identificación dentro de la asociación, en qué país se lleva a cabo y en qué zona de éste, así como el objetivo que persigue y el número de beneficiarios a los que afecta. Un proyecto se compone a su vez de subproyectos (que tienen entidad de proyectos).

Discusión del enunciado

PARTE 1

> *"De las asociaciones se desea almacenar su CIF, denominación, dirección y provincia, su tipo (ecologista, integración, desarrollo...), así como si está declarada de utilidad pública por el Ministerio del Interior."*
>
> *"Cada asociación está formada por socios de los que se precisa conocer su DNI, nombre, dirección, provincia, fecha de alta en la asociación, la cuota mensual con que colaboran y la aportación anual que realizan (que se obtendrá multiplicando la cuota mensual por los meses del año)".*

Primeramente podemos distinguir las entidades *ASOCIACIÓN* y *SOCIO*. De los atributos del primer tipo de entidad, se puede tomar el "CIF" como identificador principal, pues se supone que éste es único para cada asociación, además podemos suponer que no existen dos asociaciones con la misma denominación, por lo que el atributo "Denominación" puede ser identificador alternativo.

De entre todos los atributos que se desea almacenar para el tipo de entidad *SOCIO*, el "DNI" se puede considerar como identificador principal.

Entre las entidades *ASOCIACIÓN* y *SOCIO*, existe una interrelación **Está_Formada**, un socio debe pertenecer por lo menos a una asociación, por lo que la cardinalidad mínima es 1, además, en principio el enunciado no nos dice nada en contra de que una persona pueda ser socio de varias asociaciones, por lo tanto la cardinalidad máxima es *n*. Por otro lado una asociación puede estar formada por uno o varios socios, por lo que la interrelación **Está_Formada** es N:M. Además esta interrelación tiene un atributo propio "F_Alta" que almacena la fecha en la que un socio es dado de alta en una asociación determinada. Supondremos por simplicidad que sólo interesan los datos de los socios actuales, por tanto, un socio sólo tendrá una única fecha de alta, aunque se hubiera dado de baja y vuelto a apuntar en diferentes ocasiones.

En cuanto a los atributos "Cuota Mensual" y "Aportación Anual", los podemos considerar como atributos de la interrelación **Está_Formada** pues, si los consideramos como propios de la entidad *SOCIO*, no se recogerían las cuotas mensuales y aportaciones anuales realizadas a las distintas asociaciones de las que se puede ser socio. Además, el atributo "Aportación Anual" lo podemos considerar como derivado, pues se puede calcular mediante una función de la cuota mensual.

> *"Los trabajadores de estas organizaciones pueden ser de dos tipos: asalariados y voluntarios."*
>
> *"Los asalariados son trabajadores que cobran un sueldo y ocupan cierto cargo en la asociación. Se desea almacenar la cantidad que éstos pagan a la seguridad social y el tanto por ciento de IRPF que se les descuenta."*

"Los voluntarios trabajan en la organización desinteresadamente, siendo preciso conocer su edad, profesión y las horas que dedican a la asociación a efectos de cálculo de estadísticas.

"Cada trabajador se identifica por su DNI, tiene un nombre y una fecha de ingreso.

"Un socio no puede ser trabajador de la asociación".

Los trabajadores de las asociaciones pueden ser de dos tipos, asalariados y voluntarios. Aquí se está estableciendo una clasificación de trabajadores que obliga a introducir una jerarquía total y sin solapamiento entre el tipo de entidad *TRABAJADOR*, que actúa como supertipo, y los tipos de entidad *ASALARIADO* y *VOLUNTARIO*, que actúan como subtipos. De cualquiera de los subtipos definidos se desea almacenar el DNI, el nombre y la fecha de ingreso, por lo que éstos serán atributos comunes propios del supertipo *TRABAJADOR*, siendo el "DNI" el identificador principal. En cuanto a los atributos de cada uno de los subtipos, mención especial merece el atributo "Edad" de la entidad *VOLUNTARIO*: cualquiera que sea el tipo de entidad en el que se incluya, este atributo debe ser siempre derivado de la fecha de nacimiento (pues se puede calcular en función de ésta), lo que nos llevaría en el caso de considerarlo, a introducir la fecha de nacimiento como atributo.

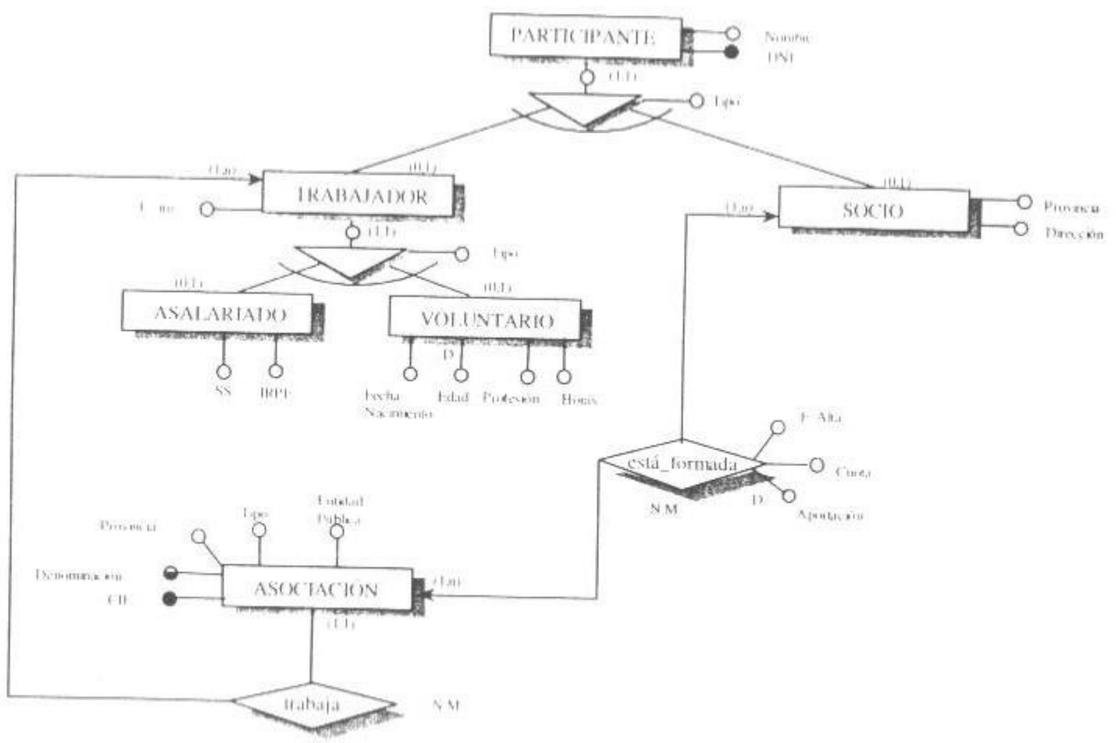


Figura 1.38

En las asociaciones trabajan tanto voluntarios como asalariados y hay que tener en cuenta que un socio no puede ser un trabajador. Esa exclusión nos obliga a

introducir una nueva jerarquía *PARTICIPANTE* donde esta entidad actúa como supertipo y *TRABAJADOR* y *SOCIO* como subtipos. Esta jerarquía será total (pues no existen más tipos de participantes) y sin solapamiento (para exigir la exclusividad); al establecer esta jerarquía, los atributos "DNI" y "Nombre", que son comunes a *SOCIO* y a *TRABAJADOR*, pasan a ser atributos del supertipo *PARTICIPANTE*.

Además definiremos una interrelación **Trabaja** entre *ASOCIACIÓN* y *TRABAJADOR* 1:N, suponiendo que ningún trabajador, aunque sea voluntario, puede trabajar en más de una asociación. Todas las consideraciones relativas a esta primera parte aparecen reflejadas en la figura 1.38 (ver figura en la página anterior).

Para modelar que un trabajador no puede ser un socio, se podría haber pensado en no construir la jerarquía *PARTICIPANTE*, sino en una interrelación exclusiva, como se muestra en la figura 1.39, pero entonces no habría forma de reflejar que un socio no puede trabajar en la asociación, pues con esto lo que se representaría es que por cada asociación o bien trabajan trabajadores o bien está formada por socios, pero no por ambos a la vez.

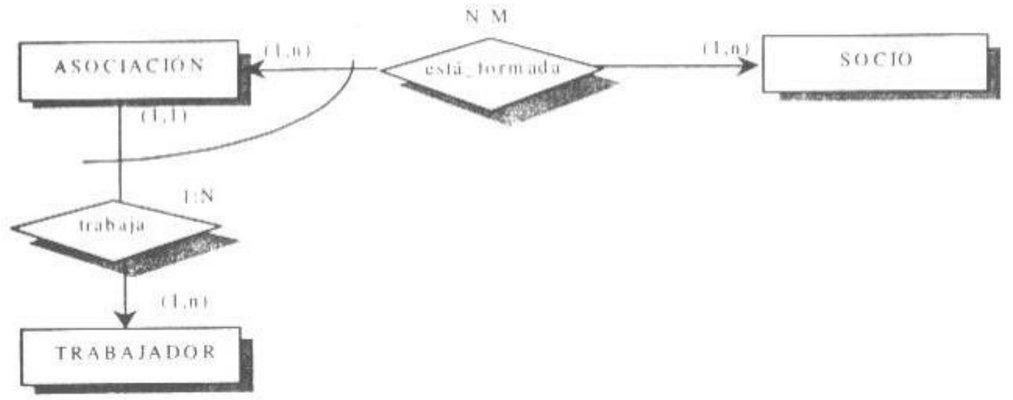


Figura 1.39

Por último, en la solución de la figura 1.38, un trabajador no puede ser socio de ninguna asociación. Si queremos que se permita, la jerarquía de *PARTICIPANTE* no debería ser exclusiva y además las interrelaciones **Trabaja** y **Está_Formada** deberían ser exclusivas para impedir que en una misma asociación una persona trabaje y sea socio.

PARTE 2

"Las asociaciones llevan a cabo proyectos a los que están asignados sus trabajadores. Un trabajador puede trabajar en diferentes proyectos de un mismo país. De cada proyecto se desea almacenar su número de identificación dentro de la asociación, en qué país se lleva a cabo y en qué

zona de éste, así como el objetivo que persigue, y el número de beneficiarios a los que afecta. Un proyecto se compone a su vez de subproyectos (que tienen entidad de proyectos)".

Se incluye en esta parte un nuevo tipo de entidad *PROYECTO*. De cada proyecto se desea almacenar el número de identificación dentro de la asociación que lo lleva a cabo, por tanto *PROYECTO* es una entidad débil con respecto a *ASOCIACIÓN*, que estará relacionada con ella a través de la dependencia en identificación 1:N **Lleva**. Si suponemos que toda asociación debe llevar a cabo al menos un proyecto, la cardinalidad mínima será 1, en otro caso la cardinalidad mínima será 0, nosotros hemos optado por la primera suposición. Por otro lado parece lógico pensar que cada proyecto es asignado a una única asociación (cardinalidad mínima 1).

Un proyecto se puede componer de subproyectos que tienen entidad de proyectos, es decir todo subproyecto es un proyecto, por lo que podemos establecer una interrelación reflexiva **Compuesto**, 1:N con *PROYECTO*, puesto que cada subproyecto (en el caso en que lo sea) debe serlo de un único proyecto.

Todo lo expuesto hasta ahora queda reflejado en la figura 1.40.

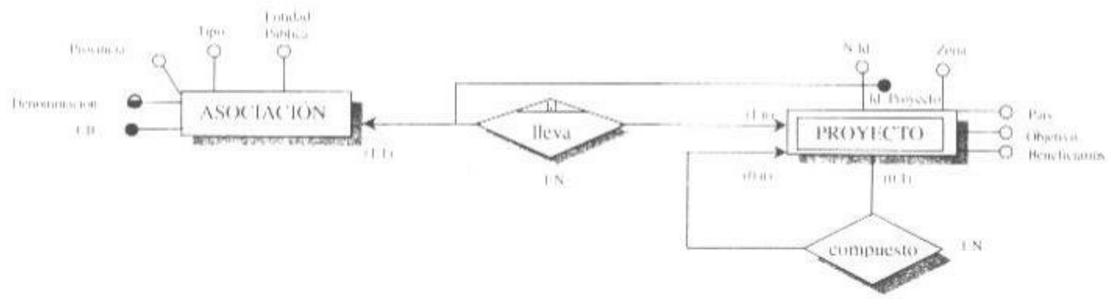


Figura 1.40

Se podría haber pensado inicialmente, sin tener en cuenta la entidad débil, en establecer una interrelación ternaria entre *ASOCIACIÓN*, *TRABAJADOR* y *PROYECTO*, pero al ser cada trabajador y cada proyecto de una única asociación no es necesario. Si además tenemos en cuenta la dependencia en identificación de *PROYECTO* con respecto a *ASOCIACIÓN*, esta solución carece por completo de sentido.

Por otra parte, si ponemos una interrelación ternaria entre *TRABAJADOR PAÍS* y *PROYECTO*, podría pensarse que podemos reflejar que un trabajador sólo trabaja en proyectos de un determinado país, pero las cardinalidades admitidas no lo permiten (lo que refleja esta interrelación es que un proyecto asignado a un trabajador se lleva a cabo en un país, y no que todos los proyectos asignados a un mismo trabajador se lleven a cabo en un mismo país).

Propuesta de solución

La figura 1.41 muestra la solución definitiva.

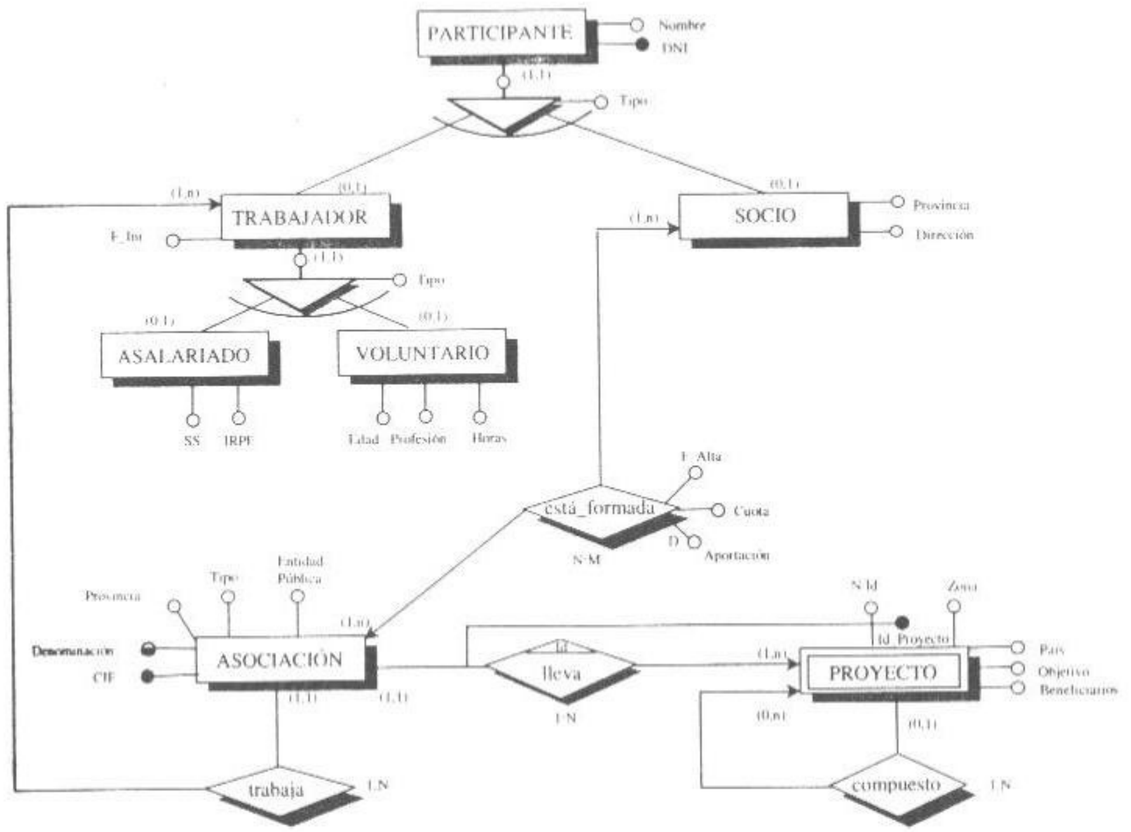


Figura 1.41

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- El atributo "Tipo" se puede describir como un dominio que agrupa a todos los tipos de ONG's (ecologista, integración...). De igual forma se puede considerar la "Entidad Pública" como un dominio que permite los valores SÍ o NO.
- Se ha supuesto que un trabajador asalariado en ningún caso puede ser voluntario y viceversa.
- Habría que comprobar que los subproyectos de un mismo proyecto se realizan en el mismo país en el que se realiza el proyecto.
- Tampoco se puede reflejar el hecho de que todos los proyectos asignados a un trabajador sean del mismo país.

- Con nuestro diseño no podemos obligar a que un socio de una asociación no pueda trabajar en otra.

PROBLEMA 9: ADMINISTRACIÓN DE FINCAS

Enunciado

Una firma de abogados dedicada a la administración de fincas desea tener una base de datos para facilitar la gestión de la información de sus clientes, es decir, de las distintas comunidades de vecinos que administra. La información que debe contener la BD concierne a los aspectos que se describen a continuación.

La firma tiene varios abogados y cada uno de ellos ejerce de administrador de una o más comunidades de vecinos, por lo que cobra a cada una de ellas unos honorarios anuales. Una comunidad de vecinos es gestionada por un único administrador (Nombre, DNI y Nº de Colegiado). Las funciones de un administrador, sobre las que en este caso interesa guardar información, consisten en llevar la contabilidad de la comunidad, gestionando los recibos que pagan los vecinos mensualmente, así como los pagos a las distintas compañías que proporcionan algún servicio a la comunidad (limpieza, ascensores, seguridad, luz, etc.).

De las empresas que tienen contratadas las distintas comunidades de vecinos (por ejemplo, Iberdrola, Unión Fenosa, OTIS, etc.) se guarda su nombre, CIF, dirección, teléfono y una persona de contacto. Además, interesa tener estas compañías agrupadas en distintos sectores (luz, seguridad, ascensores, etc.).

De cada comunidad de vecinos gestionada por la firma de abogados interesa almacenar un código identificador, su nombre, calle, código postal y población. Cada comunidad consta de una serie de propiedades que pueden ser de tres tipos (vivienda particular, local comercial y oficina). Cada propiedad se caracteriza por un número de portal, planta y letra, un nombre y apellidos del propietario con su dirección completa (que puede ser ésta u otra) y un teléfono de contacto, un porcentaje de participación en los gastos de la comunidad así como los datos de la cuenta bancaria en la que el propietario desea se le domicilie el pago de los recibos.

Si el propietario no habita en su propiedad entonces se necesitan sus datos (nombre, apellidos, dirección y teléfono de contacto) así como los del inquilino que la habita (nombre, apellidos y teléfono de contacto), en caso de que esté habitada la propiedad. Si el propietario habita en la propiedad sólo son necesarios sus datos (nombre, apellidos, teléfono de contacto).

Si la vivienda es particular se guardará el número de habitaciones de que dispone; si es un local comercial se almacenará el tipo de comercio que se desarrolla en él y el

horario (en caso de que esté en uso); si es una oficina se guardará la actividad a la que se destina.

Cada comunidad de vecinos tiene además un presidente y varios vocales (nombre, apellidos y propiedad de la que son dueños) elegidos entre todos los propietarios, que se encargan de tratar directamente con el administrador los distintos problemas que pudieran surgir.

En cuanto a la contabilidad, cada comunidad de vecinos tiene una cuenta en un banco. De los distintos bancos se almacena el código de banco, el nombre y una persona de contacto, mientras que para una cuenta bancaria se guarda un código de cuenta (que consta de un código de sucursal, dos dígitos de control y un número de cuenta) y un saldo. Para identificar una cuenta es necesario añadir al código de cuenta el código del banco en el que se encuentra.

Es necesario almacenar dos tipos de apuntes (ingresos y gastos) para la contabilidad[9] de cada comunidad de vecinos.

- Por un lado, aunque es el banco el que emite los recibos de las cuotas de comunidad a los distintos propietarios, el administrador guarda información sobre dichos recibos que se ingresan en las cuentas bancarias de las comunidades, es decir, el número de recibo, fecha, importe y si se ha podido cobrar o no. Esta última información es importante para realizar a final de cada trimestre una relación de impagados.

- En cuanto a los apuntes relativos a los gastos se tienen los importes que cobran las empresas contratadas por cada comunidad de vecinos. Las compañías cobran sus recibos (Número de recibo, fecha e importe) cargándolos en la cuenta de cada comunidad.

Discusión del enunciado

PARTE 1

> *"La firma tiene varios abogados y cada uno de ellos ejerce de administrador de una o más comunidades de vecinos por lo que cobra a cada una de ellas unos honorarios anuales. Una comunidad de vecinos es gestionada por un único administrador (Nombre, DNI y Nº de Colegiado)."*

Se detectan como posibles entidades *ADMINISTRADOR* y *COMUNIDAD DE VECINOS*. No es necesaria una entidad *FIRMA* puesto que se trata de gestionar sólo la

[9] Aunque en la realidad existen otros gastos sobre los que habría que guardar información (derramas, nóminas, gastos extras, etc.), en este caso sólo se tendrán en cuenta los aquí expuestos.

firma en cuestión dedicada a la administración de fincas, es decir, sólo habría una ocurrencia de esta entidad. Con el fin de reflejar la semántica de que un abogado ejerce de administrador de una o varias comunidades de vecinos se requiere la interrelación 1:N binaria **Gestiona** cuyas cardinalidades indican que una comunidad de vecinos es gestionada por un único administrador y un administrador puede gestionar o bien ninguna o bien varias comunidades de vecinos.

También se distingue un atributo en la interrelación **Gestiona**, los "Honorarios" que cobra cada administrador a cada comunidad de vecinos que gestiona. Por último, el identificador principal de la entidad *COMUNIDAD DE VECINOS* puede ser el "Núm_Colegiado"; el resto de atributos de esta entidad serían el "DNI" (que se definirá como identificador alternativo) y el "Nombre".

> *"Las funciones de un administrador, sobre las que en este caso interesa guardar información, consisten en llevar la contabilidad de la comunidad, gestionando los recibos que pagan los vecinos mensualmente, así como los pagos a las distintas compañías que proporcionan algún servicio a la comunidad (limpieza, ascensores, seguridad, luz, etc.)."*

En estos supuestos semánticos se mencionan los recibos de la comunidad que pagan los vecinos así como los pagos que se realizan a las compañías que ofrecen distintos servicios a las comunidades. Estos conceptos se detallarán con supuestos adicionales en la Parte 3.

> *"De las empresas que tienen contratadas las distintas comunidades de vecinos (por ejemplo, Iberdrola, Unión Fenosa, OTIS, etc.) se guarda su nombre, CIF, dirección, teléfono y una persona de contacto. Además, interesa tener estas compañías agrupadas en distintos sectores (luz, seguridad, ascensores, etc.)."*

Este supuesto semántico concierne a las empresas que prestan sus servicios a las distintas comunidades de vecinos. Así, se distingue la entidad *COMPAÑÍA* cuyos atributos son "CIF" (que es el identificador principal de la entidad), el "Nombre", "Dirección", "Persona de Contacto", "Teléfono" y "Sector" al que pertenece. "Sector" es un atributo univaluado de la entidad *COMPAÑÍA* puesto que suponemos que una compañía pertenece a un único sector.

También aparece la interrelación **Contrata** entre las entidades *COMUNIDAD DE VECINOS* y *COMPAÑÍA* con cardinalidades (1,N) a cada lado, de tal manera que una comunidad de vecinos puede tener contratadas varias compañías (puede que no tenga ninguna al principio) y una compañía puede ser contratada por una o varias comunidades de vecinos

El esquema E/R correspondiente a esta primera parte se muestra en la figura 1.42.

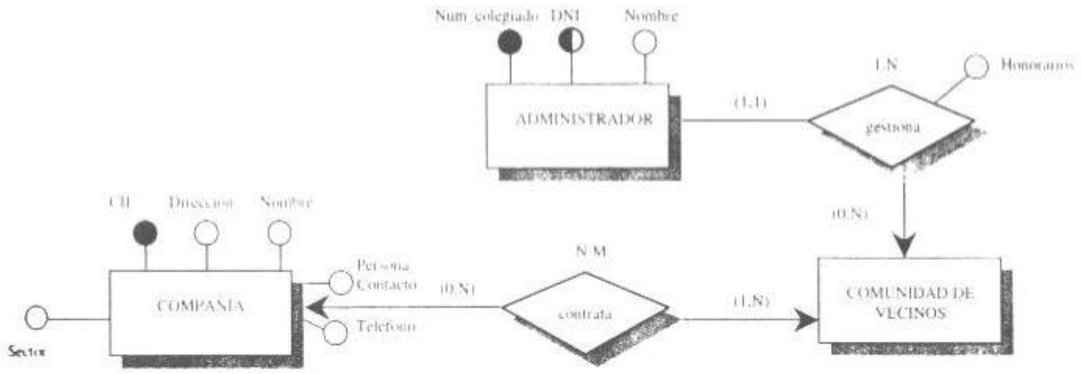


Figura 1.42

PARTE 2

> *"De cada comunidad de vecinos gestionada por la firma de abogados interesa almacenar un código identificador, su nombre, calle, código postal y población."*

En este párrafo se indica que los atributos de la entidad *COMUNIDAD DE VECINOS* son "Código de Comunidad" (como identificador principal de la entidad), "Nombre", "Calle", "Código Postal" y "Población".

> *"Cada comunidad consta de una serie de propiedades que pueden ser de tres tipos (vivienda particular, local comercial y oficina). Cada propiedad se caracteriza por un número de portal, planta y letra, un nombre y apellidos del propietario con su dirección completa (que puede ser ésta u otra) y un teléfono de contacto, un porcentaje de participación en los gastos de la comunidad así como los datos de la cuenta bancaria en la que el propietario desea se le domicilie el pago de los recibos."*

En estos supuestos semánticos se indica que existe la entidad denominada *PROPIEDAD* que engloba las viviendas particulares, los locales comerciales y las oficinas que forman cada comunidad de vecinos. Identificamos, además, la interrelación **Consta** entre *COMUNIDAD DE VECINOS* y *PROPIEDAD* de tal forma que una propiedad pertenece a una única comunidad y una comunidad consta de una propiedad como mínimo.

Los atributos de la entidad *PROPIEDAD* son "Portal", "Planta", "Letra", "Porcentaje" (que representa la participación en los gastos de la comunidad que no es igual para todas las propiedades, pues depende de su tamaño) y "Nº Cuenta". La entidad *PROPIEDAD* es una entidad débil pues cada propiedad requiere para su identificación el "Código de Comunidad" a la que pertenece, es decir, el portal, planta y letra de una

©RA-MA

determinada propiedad pueden repetirse en distintas comunidades de vecinos, por lo que es necesario añadir el "Código de Comunidad" a estos atributos para identificar una propiedad.

Para reflejar los datos del propietario se necesita estudiar el supuesto semántico mostrado a continuación.

> *"Si el propietario no habita en su propiedad entonces se necesitan sus datos (nombre, apellidos, dirección y teléfono de contacto) así como los del inquilino que la habita (nombre, apellidos y teléfono de contacto), en caso de que esté habitada la propiedad. Si el propietario habita en la propiedad sólo son necesarios sus datos (nombre, apellidos, teléfono de contacto)."*

Modelaremos los datos de propietario e inquilino de una propiedad como atributos compuestos. Como una propiedad siempre tiene un propietario, se necesita un atributo obligatorio formado por "Nombre Propietario" (que contiene el nombre y apellidos del propietario), "Teléfono de Contacto" y "Dirección". El atributo "Dirección" es a su vez un atributo opcional que no tendrá valor si el propietario habita en su propiedad; tendrá un valor si el propietario habita en otra vivienda.

En cuanto a los datos del inquilino, utilizaremos un atributo compuesto opcional (pues puede que la propiedad no tenga inquilino) formado por los atributos "Nombre Inquilino" (que contiene el nombre y apellidos del inquilino) y "Teléfono de Contacto".

> *"Si la vivienda es particular se guardará el número de habitaciones de que dispone; si es un local comercial se almacenará el tipo de comercio que se desarrolla en él y el horario (en caso de que esté en uso); si es una oficina se guardará la actividad a la que se destina."*

Este supuesto concierne a los tipos de propiedades sobre los que se desea almacenar información. Se distinguen tres tipos de propiedades (vivienda particular, local comercial y oficina) cada una con atributos específicos. Por ello, se requieren tres tipos de entidades para cada tipo de propiedad que formarán una jerarquía con el supertipo *PROPIEDAD*. La entidad *VIVIENDA PARTICULAR* tiene el atributo "Número de Habitaciones", la entidad *OFICINA* tiene el atributo "Actividad" y la entidad *LOCAL COMERCIAL* tiene los atributos opcionales "Tipo Comercio" y "Horario".

> *"Cada comunidad de vecinos tiene además un presidente y varios vocales (nombre, apellidos y propiedad de la que son dueños) elegidos entre todos los propietarios, que se encargan de tratar directamente con el administrador los distintos problemas que pudieran surgir."*

©RA-MA

Se modelarán los presidentes y vocales de las comunidades de vecinos como interrelaciones. Así, existe la interrelación **Presidente** entre *COMUNIDAD DE VECINOS* y *PROPIEDAD* de tal forma que una comunidad de vecinos tiene una única propiedad cuyo propietario es presidente y para una determinada propiedad puede que su propietario sea o no presidente de una comunidad de vecinos. También existe la interrelación **Vocal** cuyas cardinalidades indican que una comunidad de vecinos tiene varios vocales (como mínimo uno) y que una determinada propiedad puede que sea no vocal de una comunidad de vecinos. Además, las interrelaciones **Presidente** y **Vocal** son exclusivas pues si una propiedad figura como presidente no puede figurar como vocal y viceversa.

El esquema E/R correspondiente a esta segunda parte se muestra en la figura 1.43.

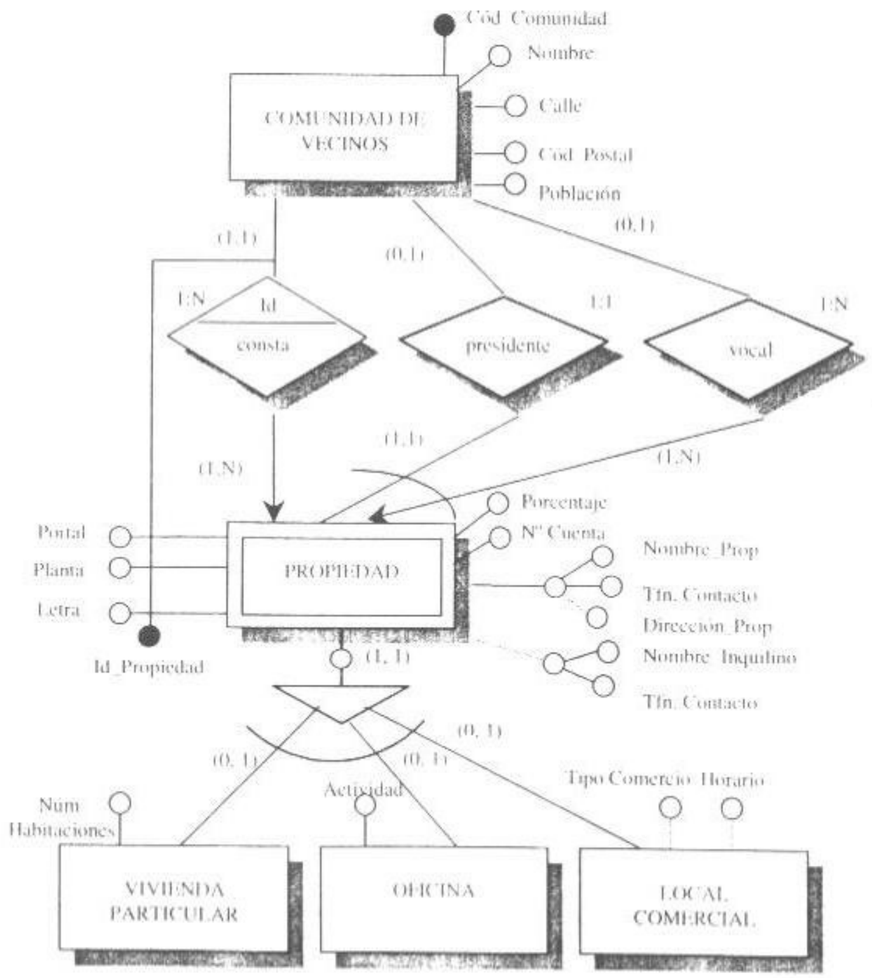


Figura 1.43

PARTE 3

> *"En cuanto a la contabilidad, cada comunidad de vecinos tiene una cuenta en un banco. De los distintos bancos se almacena el código de banco, el nombre y una persona de contacto, mientras que para una cuenta bancaria se guarda un código de cuenta (que consta de un código de sucursal, dos dígitos de control y un número de cuenta) y un saldo. Para identificar una cuenta es necesario añadir al código de cuenta el código del banco en el que se encuentra".*

En este supuesto semántico se indica que se necesitan las entidades *BANCO* y *CUENTA*. La entidad *BANCO* tiene los atributos "Código Banco", "Nombre" y "Persona de Contacto". La entidad *CUENTA* es una entidad débil pues una cuenta necesita el código de banco en que se encuentra para su identificación; por ello, el identificador principal de *CUENTA* es la concatenación de los atributos "Sucursal", "DC" (dígito de control), "Número" y "Código Banco". Además, la entidad *CUENTA* también contiene el atributo "Saldo".

Para conocer qué cuenta tiene cada comunidad de vecinos se necesita la interrelación **Controlada** entre *COMUNIDAD DE VECINOS* y *CUENTA* cuyas cardinalidades denotan que una comunidad tiene una única cuenta asociada y que una determinada cuenta corresponde a una única comunidad de vecinos.

> *"Es necesario almacenar dos tipos de apuntes (ingresos y gastos) para la contabilidad de cada comunidad de vecinos. Por un lado, aunque es el banco el que emite los recibos de las cuotas de comunidad a los distintos propietarios, el administrador guarda información sobre dichos recibos que se ingresan en las cuentas bancarias de las comunidades, es decir, el número de recibo, fecha, importe y si se ha podido cobrar o no. Esta última información es importante para realizar a final de cada trimestre una relación de impagados".*

En cuanto a la gestión de los recibos, en este párrafo se distingue la entidad *RECIBO CUOTA COMUNIDAD* correspondiente a las cuantías que pagan los propietarios de las comunidades de vecinos; esta entidad tiene los atributos "Número Recibo" (como identificador principal), "Fecha", "Importe" y "Estado" (para saber si se ha cobrado o no). Para saber a qué propiedad pertenece cada recibo se necesita la interrelación **Corresponde** entre *PROPIEDAD* y *RECIBO CUOTA COMUNIDAD* cuyas cardinalidades indican que un determinado recibo corresponde a una única propiedad y que a una propiedad le corresponden varios recibos.

Podría pensarse que es necesaria una interrelación **Se Ingresa** entre las entidades *CUENTA* y *RECIBO CUOTA COMUNIDAD*, pero esta interrelación es redundante puesto que esta información puede obtenerse a través de las interrelaciones **Corresponde** y **Controlada**.

> *"En cuanto a los apuntes relativos a los gastos se tienen los importes que cobran las empresas contratadas por cada comunidad de vecinos. Las compañías cobran sus recibos (Número de recibo, fecha e importe) cargándolos en la cuenta de cada comunidad".*

En relación con los gastos de las comunidades, se necesita la entidad *RECIBO COMPAÑÍA* cuyos atributos son "Número Recibo" (identificador principal), "Fecha" e "Importe". Para saber a qué compañía pertenece cada recibo se requiere la interrelación **Emite** entre *COMPAÑÍA* y *RECIBO COMPAÑÍA* de tal manera que un recibo corresponde a una única compañía y que una compañía puede emitir varios recibos. Por último, la interrelación **Se Carga** entre *RECIBO COMPAÑÍA* y *CUENTA* denota en qué cuentas se cargan los distintos recibos que emiten las empresas (un recibo se carga en una única cuenta y en una cuenta se pueden cargar varios recibos).

El esquema E/R correspondiente a esta parte se muestra en la figura 1.44.

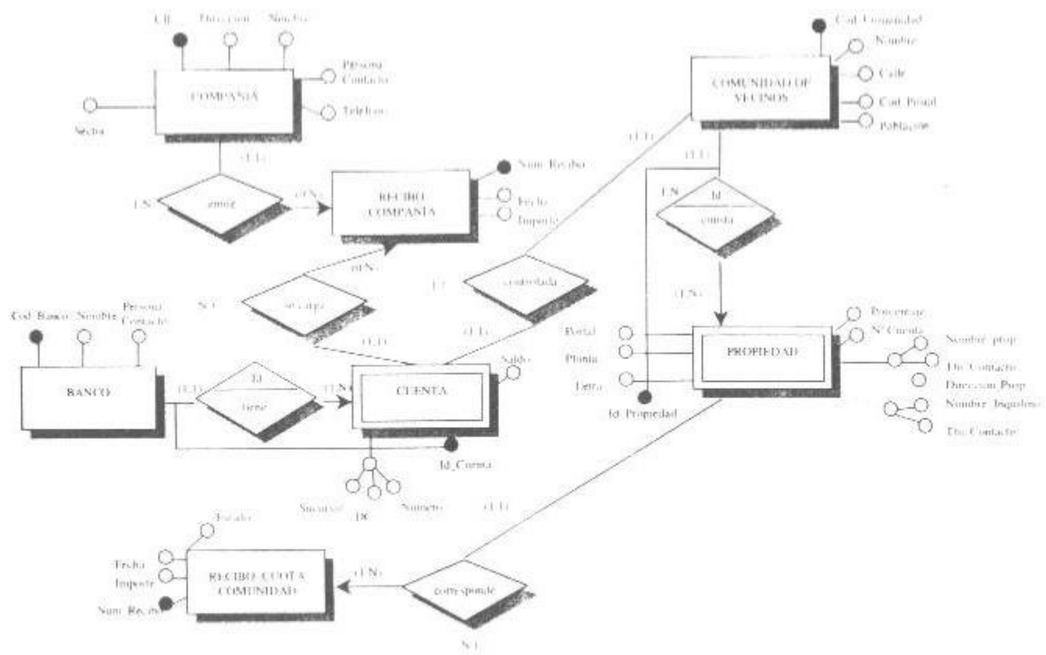


Figura 1.44

Propuesta de solución

El esquema E/R final se muestra en la figura 1.45.

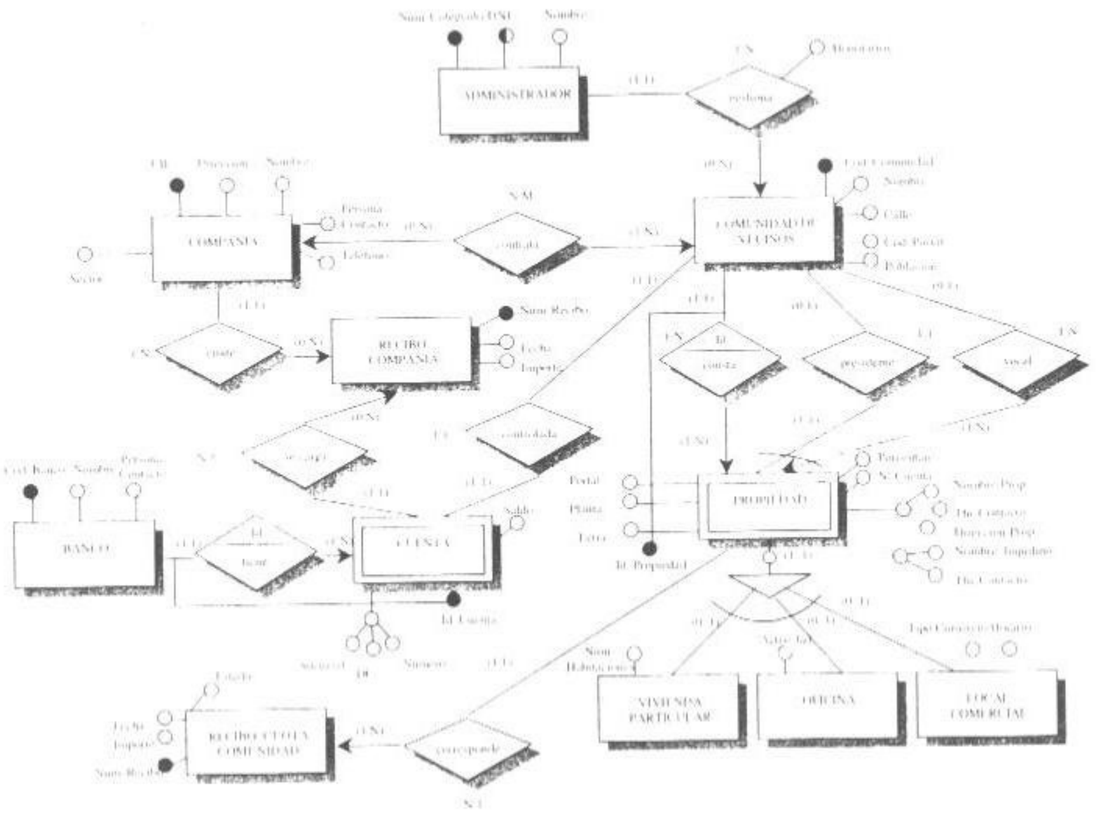


Figura 1.45

Existen otras posibilidades de modelado de algunos supuestos semánticos de este caso. Por ejemplo, en las entidades *RECIBO COMPAÑÍA* y *RECIBO CUOTA COMUNIDAD* se observa que existen tres atributos comunes ("Número Recibo", "Fecha" e "Importe"); esto nos indica que sería posible definir una jerarquía total y exclusiva cuyo supertipo fuese la entidad *RECIBO* y cuyos subtipos fuesen *RECIBO COMPAÑÍA* y *RECIBO CUOTA COMUNIDAD*, como muestra la figura 1.46. La entidad *RECIBO COMPAÑÍA* no tendría atributos propios y la entidad *RECIBO CUOTA COMUNIDAD* contendría el atributo "Estado". Estas entidades subtipo mantendrían las mismas interrelaciones que las definidas en el esquema E/R propuesto como solución, como se pueden observar en la figura 1.46.

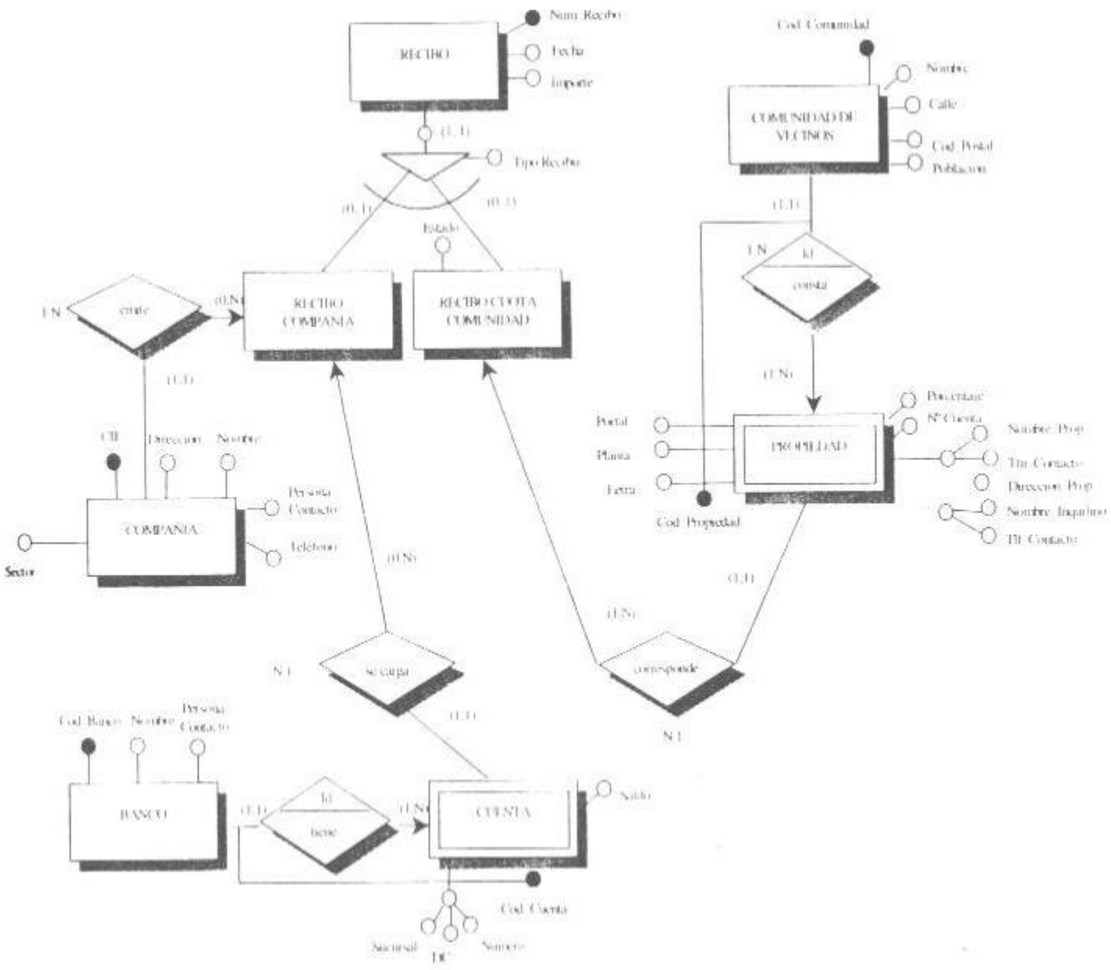


Figura 1.46

Otra posibilidad es mantener también una jerarquía de cuentas entre las cuentas bancarias de las comunidades y las cuentas de los propietarios, como se muestra en la figura 1.47. Así, la entidad supertipo sería *CUENTA* con "Cód_Cuenta" como identificador principal (formado por "Código Banco", "Sucursal", "DC" y "Número"). Los subtipos serían las entidades *CUENTA COMUNIDAD* que contiene el atributo "Saldo" y "Cuenta Propietario" que no tiene atributos propios. En este caso, se eliminaría el atributo "Nº Cuenta" de la entidad *PROPIEDAD* y se definiría una nueva interrelación denominada **Es De** entre las entidades *PROPIEDAD* y *CUENTA PROPIETARIO*, de tal forma que una cuenta de propietario pertenece a una única propiedad y una propiedad tiene una única cuenta de propietario.

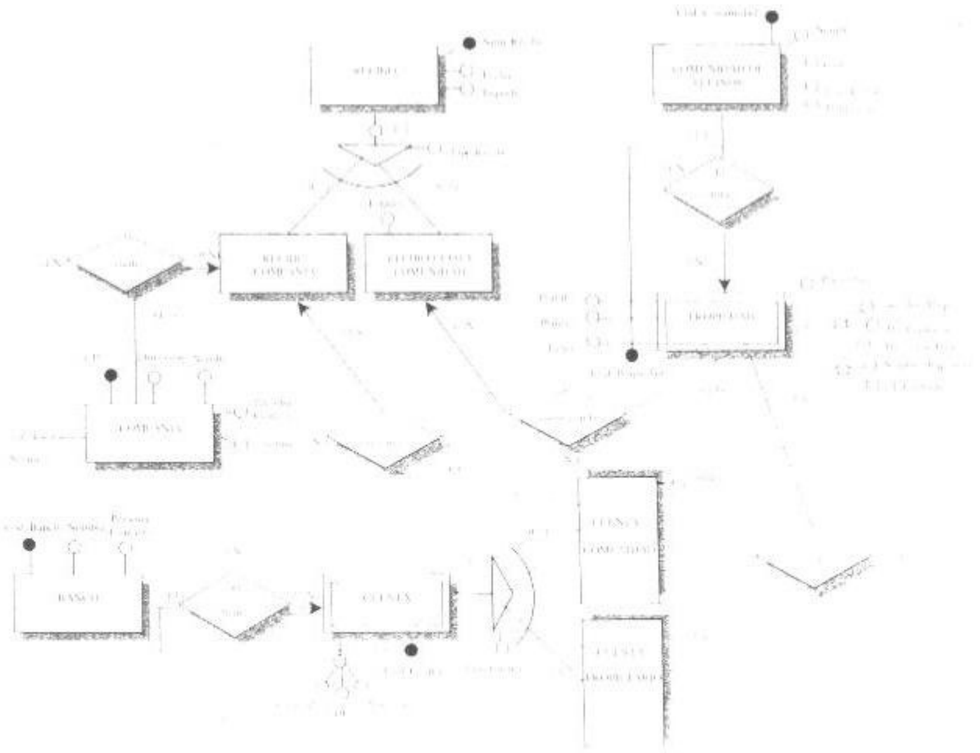


Figura 1.47

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

No ha sido necesario realizar supuestos adicionales.

PROBLEMA 10: VENTA DE COCINAS

Enunciado

Una empresa dedicada a comercializar cocinas desea aumentar su control sobre aquellos elementos que le afectan. Del resultado del análisis que realiza obtiene las siguientes informaciones:

Hay una serie de fabricantes de muebles de cocina. De cada fabricante se dispone de un nombre, una dirección y una relación de números de teléfono. Cada uno de ellos fabrica varios muebles de cocina. Un mueble de cocina tiene una determinada línea, un determinado color, unas dimensiones dadas (ancho * alto * largo), y puede tener una de las siguientes categorías excluyentes: mueble alto, mueble bajo, panel y encimera. De los muebles bajos interesa saber la altura sobre el suelo y de las encimeras interesa saber su tipo (mármol o aglomerado).

Cada fabricante puede trabajar con varios distribuidores y cada distribuidor trabaja al menos con un fabricante. De un distribuidor se dispone del nombre, dirección y una relación de números de teléfono.

Una cocina la componen una serie de muebles de cocina de distinto tipo, cada mueble de cocina sólo podrá formar parte de una única cocina. De una cocina nos interesa saber el número de muebles que la componen, así como cuántos de ellos hay de cada tipo.

Cada cocina la puede vender un único distribuidor en una determinada fecha de venta, aunque cada distribuidor puede vender varias cocinas. Un distribuidor puede ceder una cocina a otro, para que éste pueda venderla.

Cada cocina la debe montar al menos un montador, y el mismo montador puede montar varias cocinas. De un montador nos interesa su NIF, nombre, dirección, único número de teléfono y el número de cocinas que ha montado.

Cada cocina pueden comprarla uno o varios clientes, y el mismo cliente puede comprar varias cocinas. De un cliente nos interesa su NIF, su nombre, dirección y único número de teléfono.

Discusión del enunciado

PARTE 1

> *"Hay una serie de fabricantes de muebles de cocina. De cada fabricante se dispone de un nombre, una dirección y una relación de números de teléfono. Cada uno de ellos fabrica varios muebles de cocina. Un mueble de cocina tiene una determinada línea, un determinado color, unas dimensiones dadas (ancho * alto * largo), y puede tener una de las siguientes categorías excluyentes: mueble alto, mueble bajo, panel y encimera. De los muebles bajos interesa saber la altura sobre el suelo y de las encimeras interesa saber su tipo (mármol o aglomerado)."*

En este primer párrafo se reconocen inicialmente las entidades, *FABRICANTE* y *MUEBLE*, los fabricantes tienen una serie de atributos que son "Nombre", "Dirección" y "Teléfono", este último se trata de un atributo multivaluado, pues un fabricante puede tener varios números de teléfonos, es de suponer que dos fabricantes no pueden tener el mismo nombre, por lo que se empleará el atributo "Nombre" como identificador principal de los distintos ejemplares de esta entidad, además puede también suponerse que los fabricantes no comparten dirección, por lo que el atributo "Dirección" puede verse como un identificador alternativo.

Se tiene además una interrelación 1:N **Fabrica** entre estas dos entidades, si suponemos que todo fabricante de muebles de cocina lo es porque fabrica muebles de

ocina, pondremos cardinalidad mínima 1, ahora bien, si suponemos que podemos ner fabricantes de muebles de cocina registrados que no fabriquen muebles, lo que odría suceder por ejemplo si se registraran los fabricantes antes de que éstos mpezaran a producir, se podría suponer cardinalidad mínima 0, optamos en nuestro aso por la primera de las opciones. Por contra, todo mueble lo debe fabricar un eterminado fabricante y no se concibe que haya muebles no fabricados por nadie, por que la cardinalidad mínima será 1.

En cuanto a la entidad *MUEBLE*, nótese la distinción en varios tipos que de ésta e hace, consideraremos que esta clasificación da lugar a una jerarquía que será total y xclusiva en la que el supertipo será la entidad *MUEBLE*, mientras que los subtipos erán las entidades *ENCIMERA*, *MUEBLE ALTO*, *MUEBLE BAJO* y *PANEL*. Cada no de los ejemplares de *MUEBLE* tiene unas características que serán comunes a ualquiera de los subtipos y que constituirán por tanto los atributos del supertipo, se endrá así los atributos "Línea", "Color", "Ancho", "Alto" y "Largo". Dado que se requiere un identificador en esta entidad y ninguno de los atributos, ni ninguna combinación de éstos identifica unívocamente a los diversos ejemplares de *MUEBLE*, se introduce un atributo "Código" que será el identificador principal de esta entidad. En cuanto a los subtipos, tendremos que la entidad *ENCIMERA* tiene un atributo "Tipo" cuyo dominio contendrá tan sólo dos valores y que la entidad *MUEBLE BAJO* tiene un atributo que será la "Altura".

> *"Cada fabricante puede trabajar con varios distribuidores y cada distribuidor trabaja al menos con un fabricante. De un distribuidor se dispone del nombre, dirección y una relación de números de teléfono."*

Se tiene una nueva entidad *DISTRIBUIDOR* y una interrelación **Trabaja** con la ntidad *FABRICANTE*. Los atributos de *DISTRIBUIDOR*, de manera similar a lo que asaba con la entidad *FABRICANTE* serán "Nombre", "Dirección" y "Teléfono" con as mismas características que tenían en *FABRICANTE*.

Podría pensarse, dado que las entidades *FABRICANTE* y *DISTRIBUIDOR* comparten los mismos atributos, con las mismas características, que existe una generalización de ambas entidades y que éstas dos entidades no son sino subtipos de otra entidad que podría llamarse *EMPRESA*; esta generalización parece carecer de nterés en este caso, ya que esa entidad no sería más que un concepto artificialmente establecido que carecería de una de las propiedades esenciales de las entidades, la de ratarse de objetos de interés para los propósitos del sistema. No interesan las empresas en sí mismas, sino tan sólo aquellas que puedan ser fabricantes o distribuidoras. Por este motivo, se rechaza esta generalización.

Entre los fabricantes y distribuidores, existe una interrelación **Trabaja**. Esta interrelación es de tipo N:M; además como todo distribuidor debe trabajar al menos con un fabricante se tiene una cardinalidad mínima 1.

El resultado de todas las consideraciones realizadas en esta parte, se presenta en la figura 1.48.

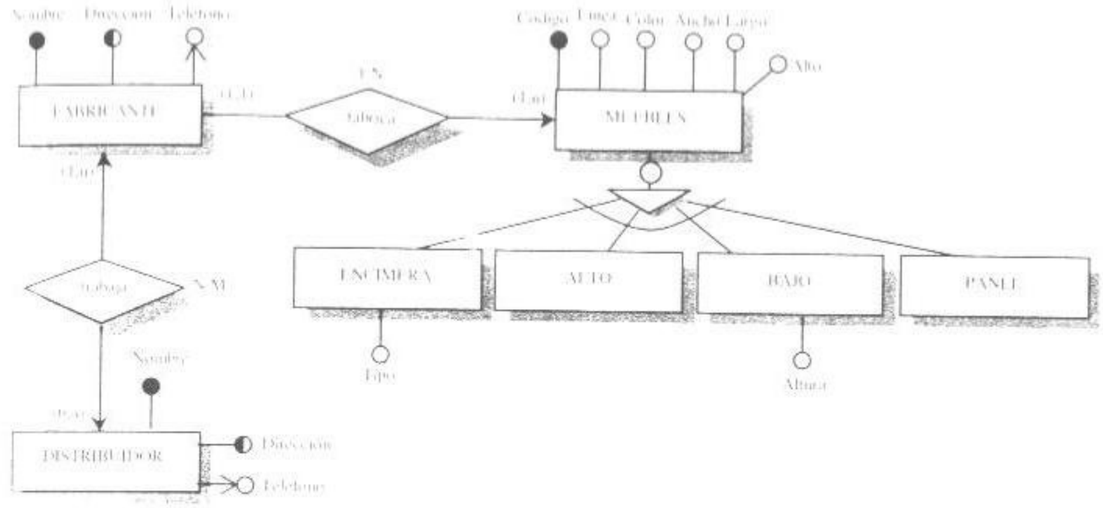


Figura 1.48

PARTE 2

> *"Una cocina la componen una serie de muebles de cocina de distinto tipo, cada mueble de cocina sólo podrá formar parte de una única cocina. De una cocina nos interesa saber el número de muebles que la componen, así como cuántos de ellos hay de cada tipo."*

Se introduce una nueva entidad *COCINA* que mantiene una interrelación 1:N **Compone** con la entidad *MUEBLE*, no todo mueble formará parte de una cocina, cardinalidad mínima 0, pero sí toda cocina la compondrá al menos un mueble, cardinalidad mínima 1. Para disponer de un identificador principal en la entidad *COCINA*, precisamos introducir el atributo "Código", además de este atributo, esta entidad tiene los siguientes atributos, "N_Muebles", "N_Altos", "N_Bajos", "N_Encimeras" y "N_Paneles", todos ellos son calculables a partir de los muebles que componen cada cocina, por ello se tratan como atributos derivados.

> *"Cada cocina la puede vender un único distribuidor en una determinada fecha de venta, aunque cada distribuidor puede vender varias cocinas. Un distribuidor puede ceder una cocina a otro, para que éste pueda venderla."*

Se tiene una interrelación 1:N **Vende** entre las entidades *COCINA* y *DISTRIBUIDOR*, esta interrelación tiene un atributo propio que es la "Fecha de Venta". Se tiene además otra interrelación reflexiva N:M **Cede** en la entidad *DISTRIBUIDOR*, aunque no se dice en el enunciado puede suponerse que un distribuidor puede hacer cesiones a varios distribuidores y que el mismo distribuidor puede recibir cesiones de varios distribuidores.

El resultado de todas las consideraciones realizadas en esta parte se presenta en la figura 1.49.

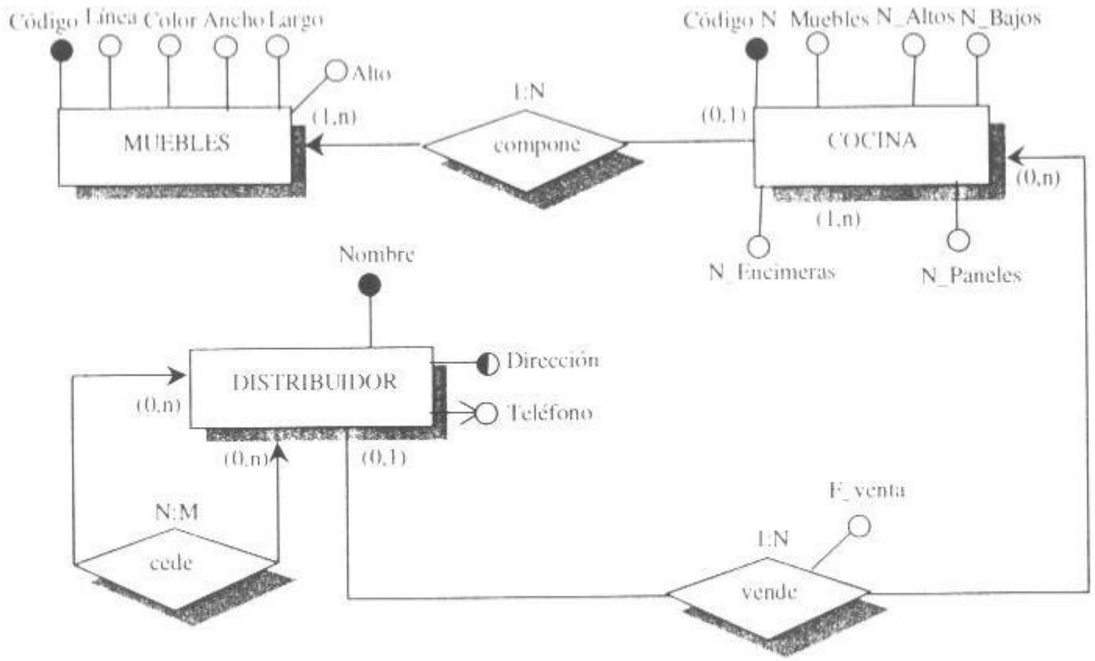


Figura 1.49

Con la anterior interrelación **Cede** se registrarían tan sólo las cesiones entre distribuidores, sin tener en cuenta qué cocina ha sido cedida en cada cesión, dado que el enunciado no dice nada al respecto, se ha optado por esta solución a fin de simplificar; sin embargo, si se quisiera captar mayor semántica y reflejar en cada cesión la cocina cedida, se tendría que establecer la interrelación ternaria presentada en la figura 1.50.

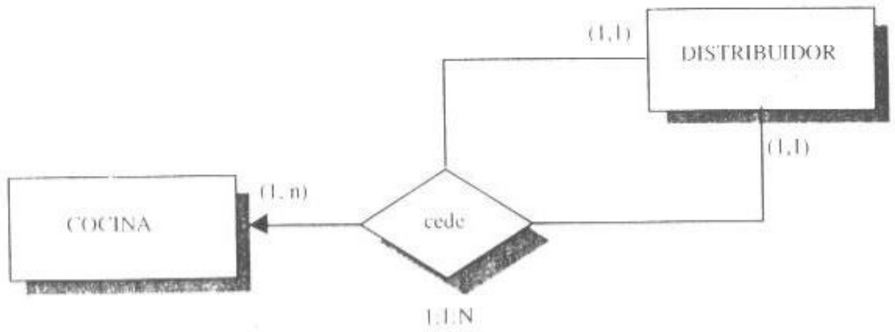


Figura 1.50

La lectura que de esta interrelación ternaria se realiza es que por cada distribuidor que cede y cada distribuidor que recibe se cede una cocina como mínimo, pero se

pueden ceder varias. Además, cada distribuidor que cede y cada cocina cedida sólo puede cederse exactamente a otro distribuidor (uno y sólo uno). Por último, por cada cocina y cada distribuidor que la recibe sólo se la puede haber cedido exactamente otro distribuidor.

Esta interrelación ternaria puede simplificarse en dos interrelaciones binarias dando lugar a las interrelaciones presentes en la figura 1.51, el hecho de que la misma entidad *DISTRIBUIDOR* aparezca dos veces en la interrelación ternaria no influye en el proceso de su simplificación. Debería asegurarse además mediante una restricción que no se puede reflejar en el esquema E/R que siempre que haya ejemplares en la interrelación **Cede** debería haber ejemplares en la interrelación **Recibe**, dado que en otro caso podría haber cocinas cedidas por un distribuidor pero no recibidas por ningún otro

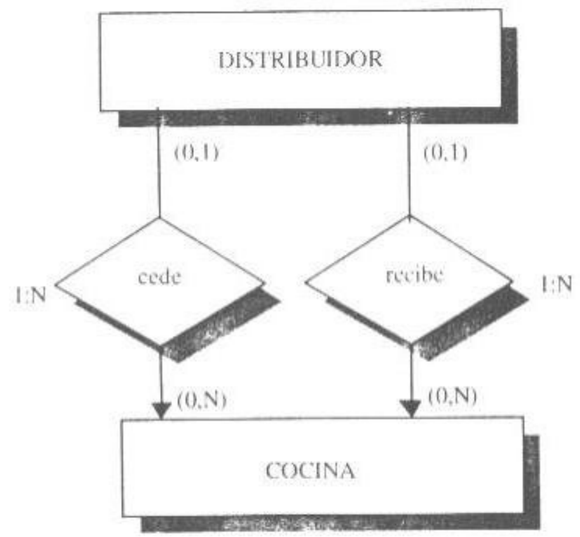


Figura 1.51

PARTE 3

> *"Cada cocina la debe montar al menos un montador, y el mismo montador puede montar varias cocinas. De un montador nos interesa su NIF, nombre, dirección, único número de teléfono y el número de cocinas que ha montado."*

Surge una nueva entidad *MONTADOR* con una interrelación N:M **Monta** con la entidad *COCINA*, esta interrelación tiene cardinalidad mínima 1 en ambos extremos. Son atributos de un montador el "Nombre", el "NIF", la "Dirección", el "Teléfono" y el "Número de Cocinas", éste último sería un atributo derivado, pues este dato está implícitamente presente en la interrelación **Monta**, además consideraremos como identificador principal de esta entidad el "NIF" y como identificadores

alternativos la "Dirección" y el "Teléfono". En este caso, "Teléfono" no es un atributo multivaluado, pues por cada montador sólo se tiene un único número de teléfono.

> *"Cada cocina puede ser comprada por uno o varios clientes, y el mismo cliente puede comprar varias cocinas. De un cliente nos interesa su NIF, nombre, dirección y único número de teléfono."*

Aparece una nueva entidad *CLIENTE* con atributos similares la entidad *MONTADOR*, puede pensarse aquí nuevamente en una generalización similar a la que planteábamos con *FABRICANTE* y *DISTRIBUIDOR*, pero se rechaza por motivos similares.

Del enunciado se desprende que hay una interrelación N:M **Compra** con la entidad *COCINA*, la cual tiene cardinalidad mínima 1 en ambos extremos.

El resultado de todas las consideraciones realizadas en esta parte, se presenta en la figura 1.52.

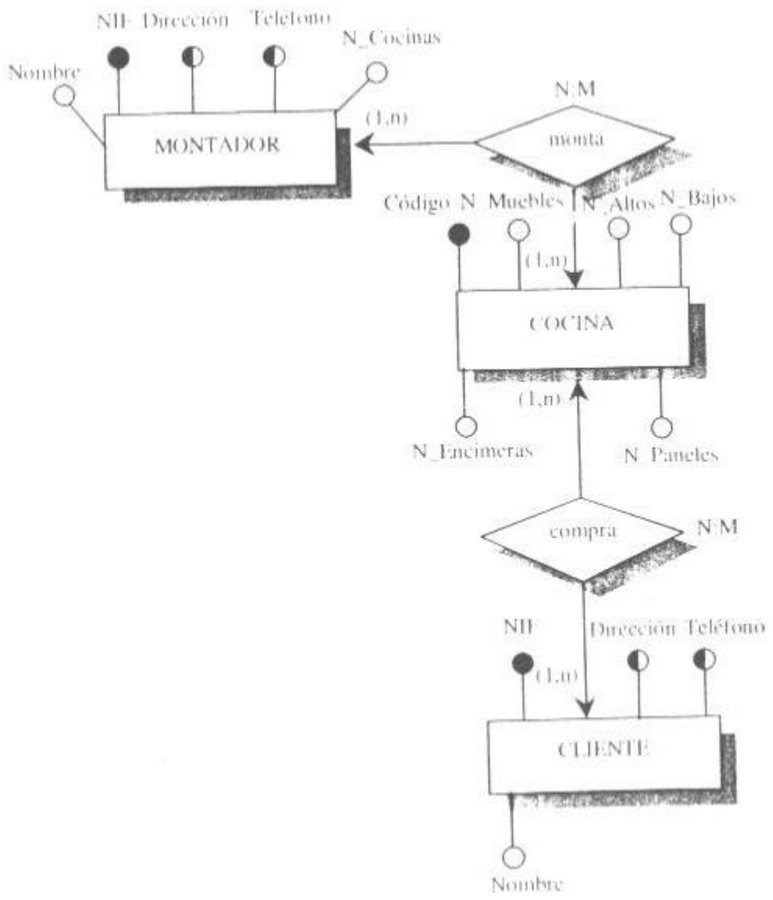


Figura 1.52

Puede observarse, sin embargo, que la interrelación **Compra** entre *COCINA* y *CLIENTE* es semánticamente complementaria de la interrelación **Vende** entre *COCINA* y *DISTRIBUIDOR*, y que aunque por el enunciado no sería estrictamente preciso, podría recogerse mayor semántica tratando de fundir estas dos interrelaciones en una única interrelación ternaria (figura 1.53).

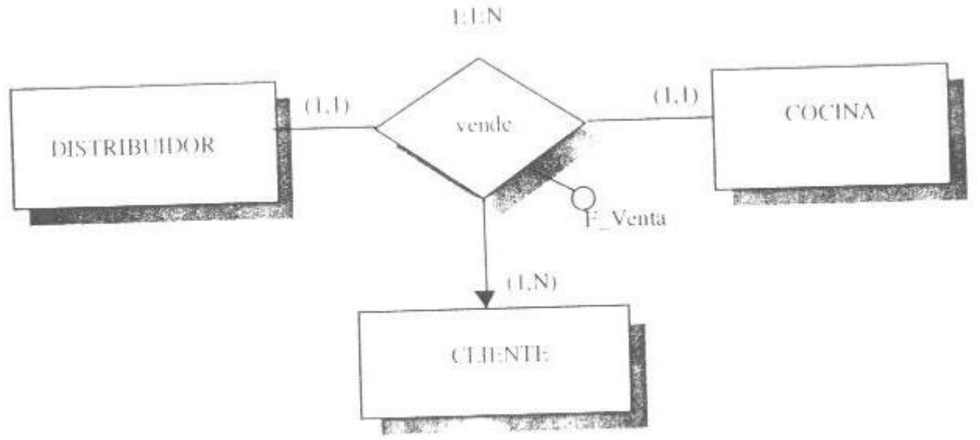


Figura 1.53

Se registraría así para cada venta, o compra, el distribuidor, la cocina y clientes que intervienen como participantes, además dado que las fechas de venta y de compra coincidirían se tendría para las compras este dato como información adicional. Si así lo hiciéramos, resultaría la interrelación ternaria presente en la figura 1.53. Aunque en esta solución no se recogería, al menos explícitamente en el esquema E/R, el hecho de que un cliente puede comprar varias cocinas (al mismo o a diferentes distribuidores) que sí se refleja en el enunciado. Por este motivo, optamos por la primera solución y no fundir esas interrelaciones.

Propuesta de solución

Agrupamos las distintas soluciones parciales obtenidas en la figura 1.54.

Vemos que en la solución obtenida aparece un ciclo, realizamos un análisis de redundancia para comprobar que no hay ninguna interrelación redundante, lo cual resulta claro, pues la semántica recogida en cada una de las interrelaciones es diferente de la obtenida a lo largo del ciclo, el único caso dudoso se presenta con la interrelación **Trabaja**, de la cual podría pensarse que es redundante, sin embargo, esto no es así, ya que el distribuidor que vende una cocina puede no trabajar con el fabricante (porque le haya sido cedida por otro que sí trabaja con el fabricante). Por tanto, no hay ninguna interrelación redundante.

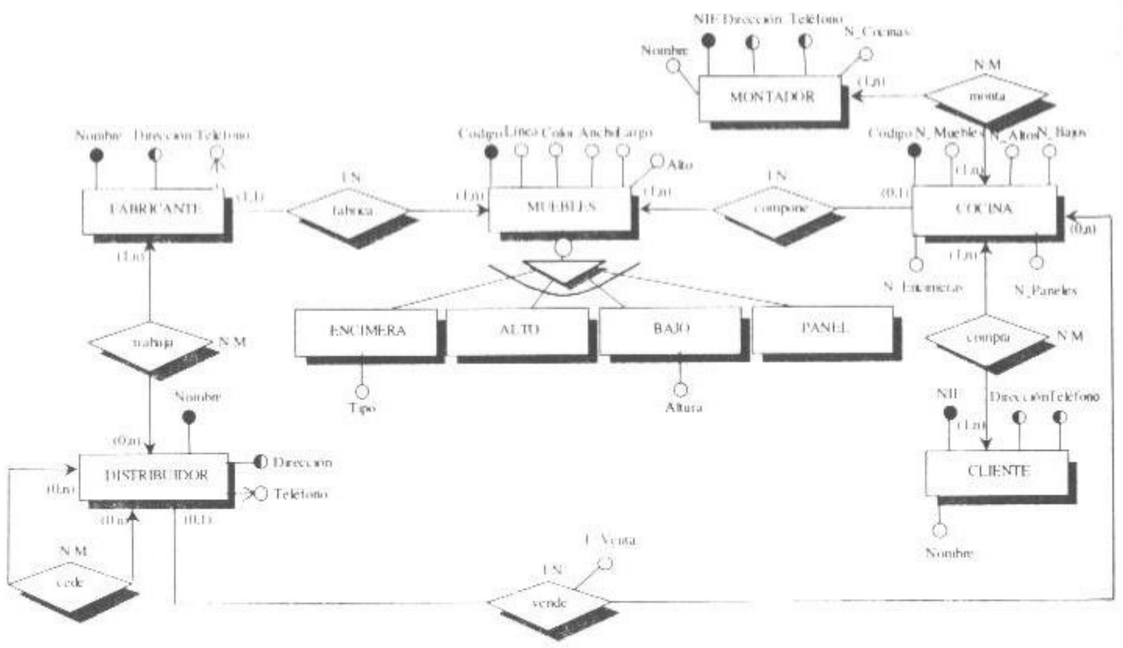


Figura 1.54

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha supuesto que todo fabricante produce al menos un mueble.
- Se ha supuesto que no hay muebles cuyo fabricante se desconozca.
- Se ha supuesto que en las cesiones entre distribuidores no interesa recoger la cocina cedida.
- Se ha supuesto que todo montador ha montado al menos una cocina.
- Se ha supuesto que todo cliente ha comprado al menos una cocina.

Por último, no figuran supuestos semánticos en el enunciado que no se hayan podido recoger.

PROBLEMA 11: ALQUILER DE VEHÍCULOS

Enunciado

Se desea diseñar una base de datos sobre la información de las reservas de una empresa dedicada al alquiler de automóviles teniendo en cuenta que:

Un determinado cliente puede tener en un momento dado hechas varias reservas. De cada cliente se desea almacenar su DNI, nombre, dirección y teléfono. Además dos clientes se diferencian por un código único.

Cada cliente puede ser avalado por otro cliente de la empresa.

Una reserva la realiza un único cliente pero puede involucrar a varios coches.

Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los coches, los litros de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el coche o los coches han sido entregados.

No se mantienen los datos de reservas anteriores.

Todo coche tiene siempre asignado un determinado garaje que no puede cambiar. De cada coche se requiere la matrícula, el modelo, el color y la marca.

Cada reserva se realiza en una determinada agencia.

Discusión del enunciado

PARTE 1

> *"Un determinado cliente puede tener en un momento dado hechas varias reservas. De cada cliente se desea almacenar su DNI, nombre, dirección y teléfono. Además dos clientes se diferencian por un código único."*
>
> *"Cada cliente puede ser avalado por otro cliente de la empresa."*

Del análisis del primer párrafo se pueden distinguir dos tipos de entidad, *CLIENTE* y *RESERVA*. De los atributos de la entidad *CLIENTE* se puede tomar el código ("Cód_Cli") como identificador principal y el "DNI" como identificador alternativo.

Entre las entidades *CLIENTE* y *RESERVA* existe una interrelación **Realiza**. Como los clientes pueden realizar varias reservas, y no se nos dice nada en contra de que los clientes puedan no realizar ninguna, las cardinalidades mínima y máxima de la interrelación en el extremo de la entidad reserva serán, respectivamente, 0 y *n*. En cuanto a las cardinalidades de la entidad cliente y, por tanto del tipo de correspondencia de la interrrelación, todavía no tenemos suficiente información para dilucidarlas.

Cada cliente puede ser avalado por otro cliente, estamos ante una interrelación reflexiva, cuyas cardinalidades mínima y máxima serán 0 y 1 en este sentido. Es más,

un cliente puede avalar como mínimo a 0 clientes y como máximo a *n*, pues el enunciado no impone ninguna restricción,. Por todo ello el tipo de correspondencia será 1:N.

Todas las consideraciones hechas hasta el momento quedan reflejadas en la figura 1.55.

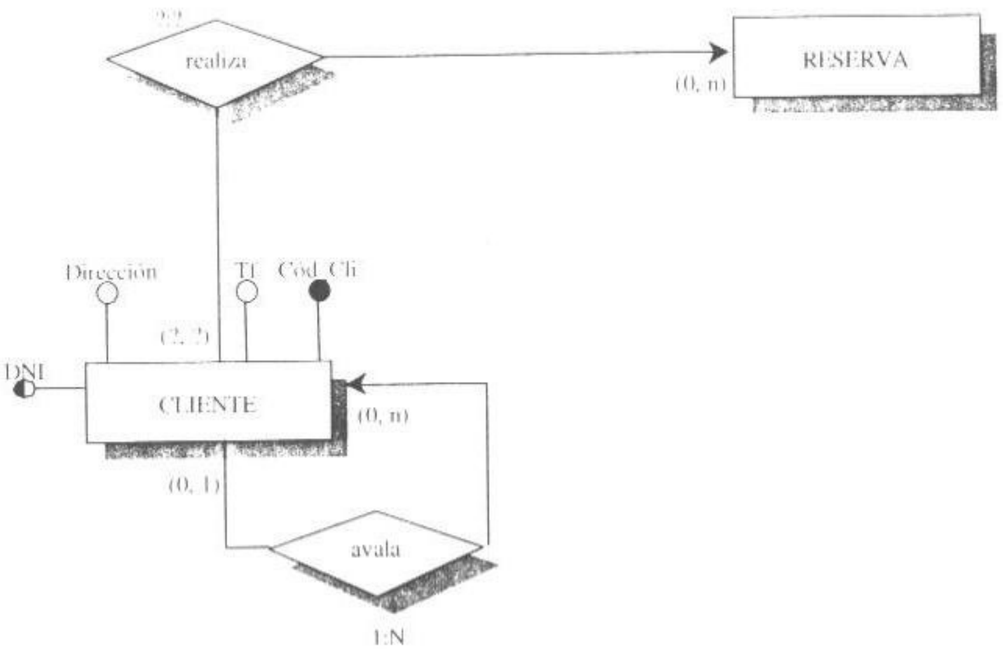


Figura 1.55

PARTE 2

"Una reserva la realiza un único cliente pero puede involucrar a varios coches."

"Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los coches, los litros de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el coche o los coches han sido entregados."

"No se mantienen los datos de reservas anteriores."

Una reserva es realizada por un único cliente, esto significa que las cardinalidades mínima y máxima de la interrelación **Realiza** en el extremo de la entidad *CLIENTE* (que antes no se conocían) son 1 y 1, respectivamente, y, por lo tanto el tipo de correspondencia de la interrelación **Realiza** es 1:N.

Por otra parte, como una reserva puede involucrar a varios coches, tenemos un nuevo tipo de entidad *COCHE*, que se relaciona con *RESERVA* mediante interrelación **Involucra**, con cardinalidad máxima *n*. Además, parece lógico pensar que cada reserva involucra como mínimo a un coche (cardinalidad mínima 1). Por otro lado, un coche está involucrado en una única reserva (recordemos que no se almacena la historia de reservas), con lo que las cardinalidades mínima y máxima de la entidad *RESERVA* con respecto a *COCHE* son 1 y 1, respectivamente. Por todo ello, el tipo de correspondencia de la interrelación **Involucra** es 1:N. En el caso en que se precisara almacenar la historia de las reservas, esta interrelación sería N:M e incluiría los atributos "F_I" y "F_F" como atributos multivaluados (el primero sería obligatorio).

En cuanto a los atributos de la entidad *RESERVA*, como en el enunciado no se nos dice nada acerca de como se identifican las distintas reservas, podemos considerar un atributo código de reserva "Cód_R" como identificador principal. El precio de alquiler de cada coche "P_Alquiler" debe ser un atributo propio de la interrelación **Involucra**. Por último, el precio total de la reserva "Precio_T" será un atributo derivado calculable a partir del precio de alquiler de cada uno de los coches involucrados en la reserva.

Todo lo descrito en esta segunda parte aparece reflejado en la figura 1.56.

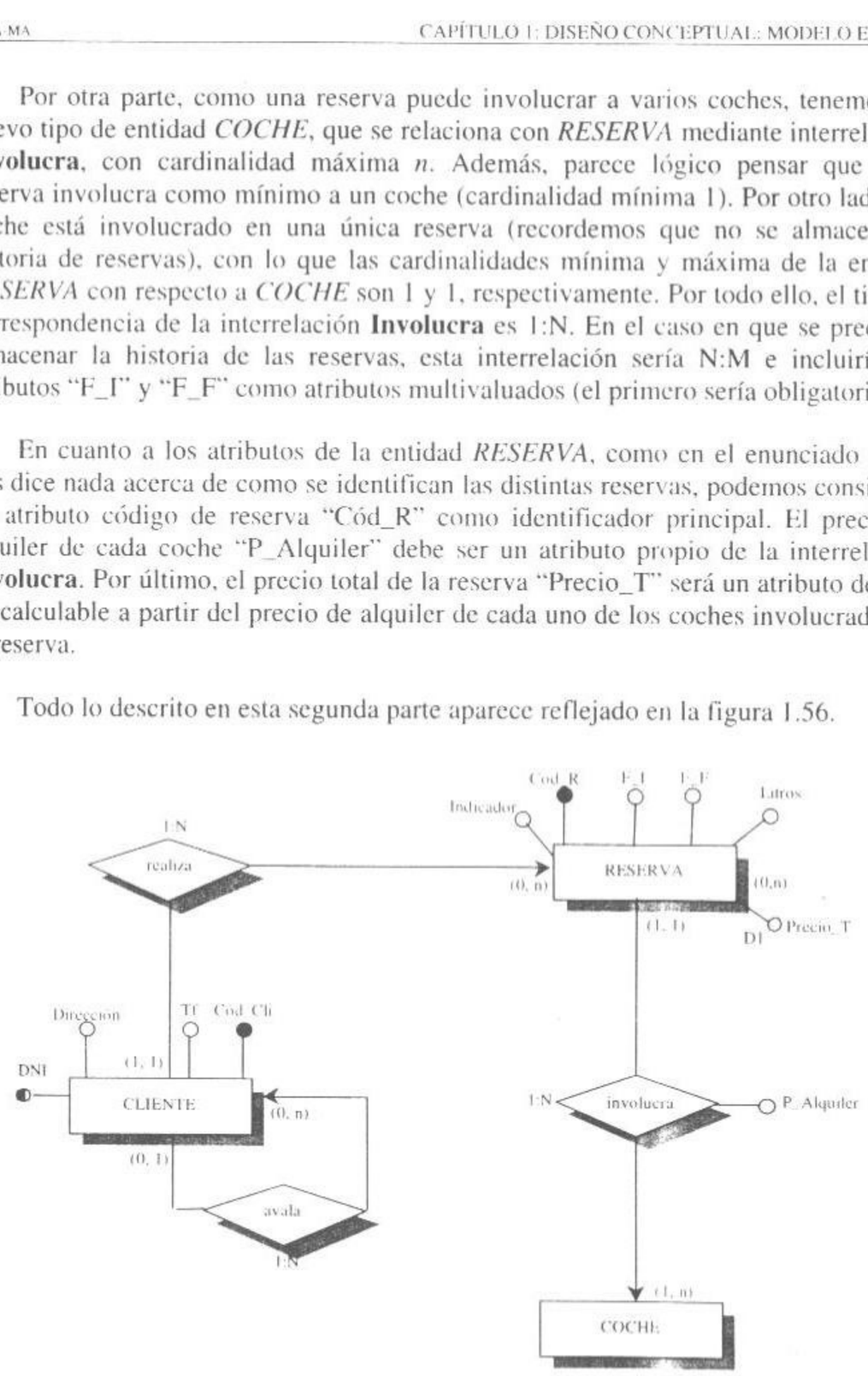


Figura 1.56

PARTE 3

> *"Todo coche tiene siempre asignado un determinado garaje que no puede cambiar. De cada coche se requiere la matrícula, el modelo, el color y la marca."*
>
> *"Cada reserva se realiza en una determinada agencia."*

Del análisis de esta última parte se deduce la existencia de dos nuevas entidades, *GARAJE* y *AGENCIA*. De cualquiera de estas dos entidades, como no se nos dice nada al respecto, supondremos un código como identificador principal, "Cód_G" y "Cód_A", para *GARAJE* y *AGENCIA*, respectivamente.

En cuanto a los atributos de la entidad *COCHE*, supondremos "Matrícula" como identificador principal.

Entre las entidades *COCHE* y *GARAJE* existe una interrelación que denominaremos **Asigna**. Un coche siempre está asignado a un único garaje, esto significa que las cardinalidades mínima y máxima de la entidad *GARAJE* serán 1. Sin embargo, un garaje puede albergar como mínimo a un coche (cardinalidad mínima 1) y como máximo a *n* (cardinalidad máxima *n*), por lo tanto el tipo de correspondencia para la interrelación **Asigna** será 1:N.

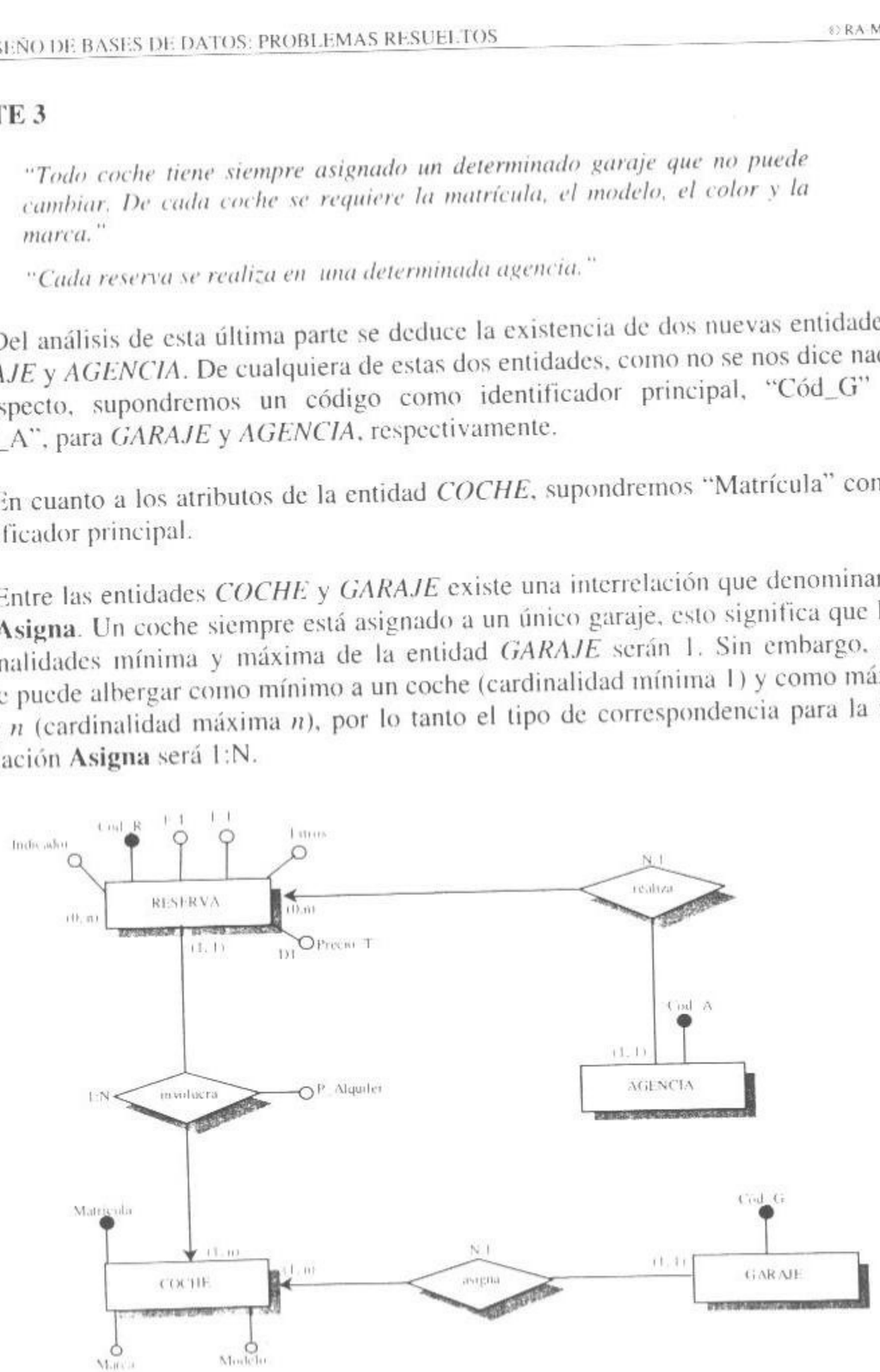


Figura 1.57

Como cada reserva se realiza en una agencia, entre las entidades *RESERVA* y *AGENCIA* podemos crear una interrelación **Realiza**, cuyo tipo de correspondencia es 1:N, pues cada reserva se realiza en una única agencia (cardinalidades mínima y máxima 1 y 1 para la entidad *RESERVA*), pero en una misma agencia se pueden realizar varias reservas y, en particular ninguna (cardinalidades mínima y máxima 0 y *n* para la entidad *AGENCIA*).

Todas estas consideraciones aparecen reflejadas en la figura 1.57.

Propuesta de solución

La figura 1.58 muestra el esquema definitivo.

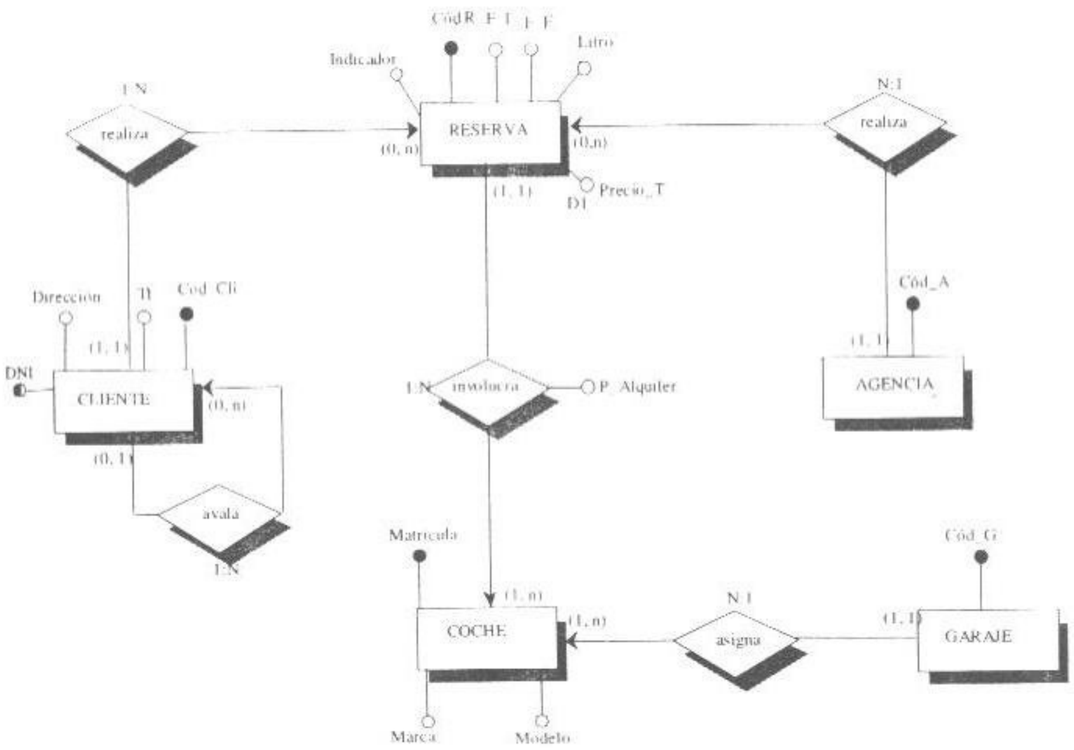


Figura 1.58

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

Toda la semántica del problema ha sido reflejada, y no ha sido necesario realizar supuestos adicionales.

PROBLEMA 12: EMPRESA NUMISMÁTICA

Enunciado

DURHOS, S. A., empresa dedicada a la compra-venta de material filatélico y numismático, ha decidido desarrollar una base de datos para llevar un mejor control de su negocio. En una primera fase se contemplarán únicamente las monedas españolas, dejando para más adelante la parte filatélica y la ampliación a material de otros países. Tras una conversación con el dueño de la empresa y la lectura de diversos libros y publicaciones especializados, hemos obtenido la siguiente información:

Existen modelos de monedas, identificados por un código, y caracterizados por una serie de propiedades, que harán que un modelo sea identificable externamente por los ciudadanos[10], tales como el valor facial, la unidad monetaria (pesetas, euros, etc.), diámetro, peso en gramos y una descripción. Habrá de especificarse, además, el metal o metales que se emplearán para cada modelo, así como la proporción de éstos y, en su caso, la ley del mismo (por ejemplo, plata de 925).

Ajustándose a los anteriores modelos, se realizarán moldes o troqueles que servirán para la acuñación de las monedas[11]. Un molde estará identificado por un código, vendrá caracterizado por un año de acuñación, y estamos interesados en almacenar determinadas características que pueden aparecer (o no) grabadas en el mismo:

- El año, grabado en caracteres visibles, y que no tiene por qué coincidir con el de acuñación[12].
- Determinadas monedas (o, mejor dicho, los troqueles con que se acuñan) tienen, normalmente a ambos lados de la fecha anterior, un par de estrellas en cuyo interior se graba en diminutos caracteres la fecha (habitualmente, la de acuñación). No estamos tan interesados en conocer el número de estrellas que contiene como la cifra que aparece grabada en ellas[13] (caso de existir).

[10] Por ejemplo, un modelo sería el correspondiente a las pequeñas monedas de 5 pesetas que comenzaron a acuñarse a partir de 1989. Aunque pueden tener distintos motivos grabados en su interior, son claramente identificables del resto de modelos, y a su vez distintas de las correspondientes al anterior modelo de monedas del mismo valor de níquel que se acuñaron hasta ese mismo año.

[11] Aunque el caso más habitual es que cada año se realice un molde nuevo para un modelo dado (hasta que éste deje de utilizarse), puede ocurrir que en un mismo año se realice más de un molde, o que no se realice ninguno. Por ejemplo, en el año 1991 no se realizó ningún molde del modelo correspondiente a la moneda de 50 pts., mientras que al año siguiente se acuñó con dos moldes distintos.

[12] De hecho, suele coincidir con el de promulgación de la ley que obliga a la acuñación de la moneda con determinadas características.

[13] Esta cifra no tiene por qué ser un año completo, sino que puede ser la última o últimas cifras del año.

- Una descripción, que podremos utilizar para especificar los distintos motivos o leyendas que aparecerán en la moneda.

Una moneda se acuñará de acuerdo a un único molde o troquel. Sin embargo, para un mismo molde podrían obtenerse distintos tipos de monedas. Esto ocurrirá en el caso de que por causa de roturas en el troquel, defectos de acuñación, o cualquier otra circunstancia, un conjunto de monedas presenten algún tipo de variante respecto al troquel original. Solamente estamos interesados en aquellas variantes que aparecen recogidas en los catálogos utilizados por los coleccionistas. Toda moneda (incluidas las variantes catalogadas) se identificará por un código único, y tendrá como características propias si es o no variante y, en caso de serlo, una descripción del error que presenta la moneda[14].

Las monedas, en función de su estado de conservación, tendrán un precio aproximado. Un estado de conservación se identifica por sus iniciales (BC, MBC, EBC, SC, etc..)[15], y tiene un nombre y una descripción. Interesa almacenar el precio estimado de una moneda según su estado de conservación.

La empresa tendrá ejemplares concretos de monedas, que se identificarán por el código de la moneda más un número correlativo. Los ejemplares podrán haber sido adquiridos a proveedores o conseguidas por cualquier otro medio (adquiridas en mercadillos, regaladas, llegadas a nuestro poder por el uso cotidiano, etc.). Aunque estamos interesados en almacenar (siempre que lo conozcamos) el precio y la fecha de adquisición del ejemplar, sólo estamos interesados en la procedencia de aquellos que fueron adquiridos a proveedores, de los que nos interesa conocer su nombre, dirección y teléfono de contacto.

Asimismo, los ejemplares podrán ser vendidos a clientes. Cada ejemplar podrá ser vendido como máximo a un cliente, y no se admiten devoluciones. Interesa conocer la fecha y el precio de venta, y del cliente estamos interesados en su nombre, dirección, teléfono y el número de compras que nos ha realizado.

Todo ejemplar se encontrará en un estado de conservación determinado. Éste, sin embargo, no tiene por qué coincidir exactamente con los especificados anteriormente, sino que puede necesitar un pequeño ajuste (indicado por los signos '+' y '-') y un comentario (por ejemplo, BC --, "Pequeño golpe en el borde").

[14] Debido a que las variantes pueden deberse a causas difícilmente clasificables, únicamente almacenaremos una descripción de la misma, incluso aunque este error se deba a una grabación incorrecta de la fecha, que es uno de los errores más comunes.

[15] Buena Conservación, Muy Buena Conservación, Extraordinariamente Buena Conservación, Sin Circular.

Discusión del enunciado

PARTE 1

> *"Existen modelos de monedas, identificados por un código, y caracterizados por una serie de propiedades, que harán que un modelo sea identificable externamente por los ciudadanos, tales como el valor facial, la unidad monetaria (pesetas, euros, etc.), diámetro, peso en gramos y una descripción. Habrá de especificarse, además, el metal o metales que se emplearán para cada modelo, así como la proporción de éstos y, en su caso, la ley del mismo (por ejemplo, plata de 925)."*

En el primer párrafo podemos detectar claramente la existencia de una entidad *MODELO* de moneda, que tendrá como identificador principal "Código", y como atributos "Valor Facial", "Unidad Monetaria", "Diámetro", "Peso" y "Descripción". Para poder almacenar el metal o metales con que se fabricarán las monedas que sigan ese modelo, podríamos pensar en un atributo multivaluado o bien en crear una nueva entidad *METAL* relacionada con *MODELO*. Hemos optado por esta segunda opción porque supondremos que en un futuro se añadirán nuevos atributos a la entidad *METAL*. De modo que crearemos una entidad *METAL*, y tanto la "Proporción" como la "Ley" de un metal en un molde los modelaremos como atributos de la interrelación entre ambas entidades. Las cardinalidades máximas son bastante evidentes. En cuanto a las mínimas, parece claro que un modelo necesita al menos un metal (cardinalidad mínima 1), mientras que hemos considerado que podemos tener almacenados metales que no formen parte de ningún modelo (cardinalidad mínima 0). Las consideraciones anteriores se reflejan en el esquema de la figura 1.59.

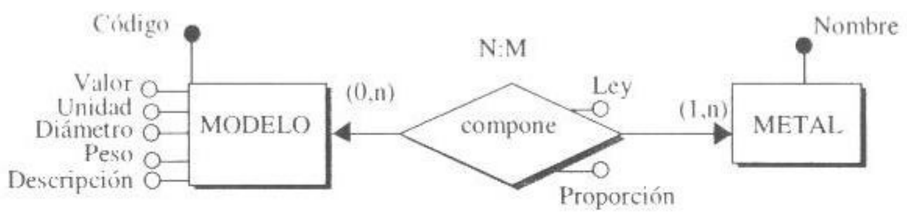


Figura 1.59

PARTE 2

> *"Ajustándose a los anteriores modelos, se realizarán moldes o troqueles que servirán para la acuñación de las monedas. Un molde estará identificado por un código, vendrá caracterizado por un año de acuñación, y estamos interesados en almacenar determinadas características que pueden aparecer (o no) grabadas en el mismo:*
>
> *El año, grabado en caracteres visibles, y que no tiene por qué coincidir con el de acuñación.*
>
> *Determinadas monedas (o, mejor dicho, los troqueles con que se acuñan) tienen, normalmente a ambos lados de la fecha anterior, un par de estrellas en cuyo interior se graba en diminutos caracteres la fecha (habitualmente, la de acuñación). No estamos tan interesados en conocer el número de estrellas que contiene como la cifra que aparece grabada en ellas (caso de existir).*
>
> *Una descripción, que podremos utilizar para especificar los distintos motivos o leyendas que aparecerán en la moneda.*
>
> *Una moneda se acuñará de acuerdo a un único molde o troquel. Sin embargo, para un mismo molde podrían obtenerse distintos tipos de monedas. Esto ocurrirá en el caso de que por causa de roturas en el troquel, defectos de acuñación, o cualquier otra circunstancia, un conjunto de monedas presenten algún tipo de variante respecto al troquel original. Solamente estamos interesados en aquellas variantes que aparecen recogidas en los catálogos utilizados por los coleccionistas. Toda moneda (incluidas las variantes catalogadas) se identificará por un código único, y tendrá como características propias si es o no variante y, en caso de serlo, una descripción del error que presenta la moneda."*

A continuación pasaremos a modelar las monedas en sí. Tal como se nos especifica en el enunciado, toda moneda se acuña utilizando un determinado molde o troquel, el cual se basa, a su vez, en un modelo. Tendremos por tanto una interrelación **Sigue** entre *MOLDE* y *MODELO*, y una interrelación **Acuña** entre *MONEDA* y *MOLDE*. De cada moneda catalogada (que puede o no ser variante del molde original) tendremos distintos ejemplares concretos, los cuales, según nos dice el enunciado, necesitan para su identificación del código de la moneda. Esto nos especifica claramente una interrelación con Dependencia en Identificación entre *MONEDA* y *EJEMPLAR*.

> *"Las monedas, en función de su estado de conservación, tendrán un precio aproximado. Un estado de conservación se identifica por sus iniciales (BC, MBC, EBC, SC, etc..), y tiene un nombre y una descripción. Interesa almacenar el precio estimado de una moneda según su estado de conservación.*
>
> *La empresa tendrá ejemplares concretos de monedas, que se identificarán por el código de la moneda más un número correlativo. [...]*
>
> *Todo ejemplar se encontrará en un estado de conservación determinado. Éste, sin embargo, no tiene por qué coincidir exactamente con los especificados anteriormente, sino que puede necesitar un pequeño ajuste (indicado por los signos '+' y '-') y un comentario (por ejemplo, BC --, "Pequeño golpe en el borde")."*

Con respecto al *ESTADO* en el que se pueden encontrar las monedas, decidimos modelarlo como una entidad por dos motivos: en primer lugar, una moneda catalogada en determinado estado tiene un precio aproximado, con lo que dicho precio sería un atributo que depende no sólo de una moneda, sino de una moneda en un estado determinado. En segundo lugar, los propios ejemplares concretos de monedas (las monedas "reales" que compramos y vendemos) se encuentran en un estado

determinado, el cual quizá necesite ser ajustado hacia arriba o hacia abajo. Es decir, *ESTADO* participa en dos interrelaciones, una con las monedas concretas (*EJEMPLAR*), y otra con las Monedas catalogadas (*MONEDA*). Además, como la entidad *ESTADO* contendrá ocurrencias para los estados *habituales* o catalogados de las monedas, necesitaremos almacenar el "Ajuste" que modificará el estado de un ejemplar. Consideramos que tanto este atributo como el que contiene una descripción aclaratoria del estado de la moneda ("Comentario") deben ser atributos de la interrelación **Está En**, que relaciona *EJEMPLAR* con *ESTADO*.

En cuanto a las cardinalidades, podemos comentar que se ha considerado que las mínimas de las interrelaciones que enlazan *ESTADO* con *EJEMPLAR* y *MONEDA* son 0 porque no tenemos por qué tener almacenada información acerca del valor de determinadas monedas (ni, por supuesto, ejemplares) en algunos estados (de hecho, es habitual que para monedas de reciente acuñación sólo se especifiquen precios aproximados para monedas en estado SC). Con respecto a las cardinalidades de un *EJEMPLAR* con respecto a *ESTADO*, es evidente que un ejemplar se encontrará en un único estado. Hemos considerado, además, que conocemos el estado de todos nuestros ejemplares (cardinalidad mínima 1), aunque también sería perfectamente admisible haberlo modelado con una cardinalidad mínima 0, es decir, dando a entender que de determinados ejemplares no conocemos su estado (por ejemplo, porque nuestra empresa no disponga de un tasador). En este último caso, sería más apropiado que el "Comentario" acerca del estado fuera una propiedad del ejemplar, y no de la interrelación. La figura 1.60 muestra el esquema que refleja la semántica anterior.

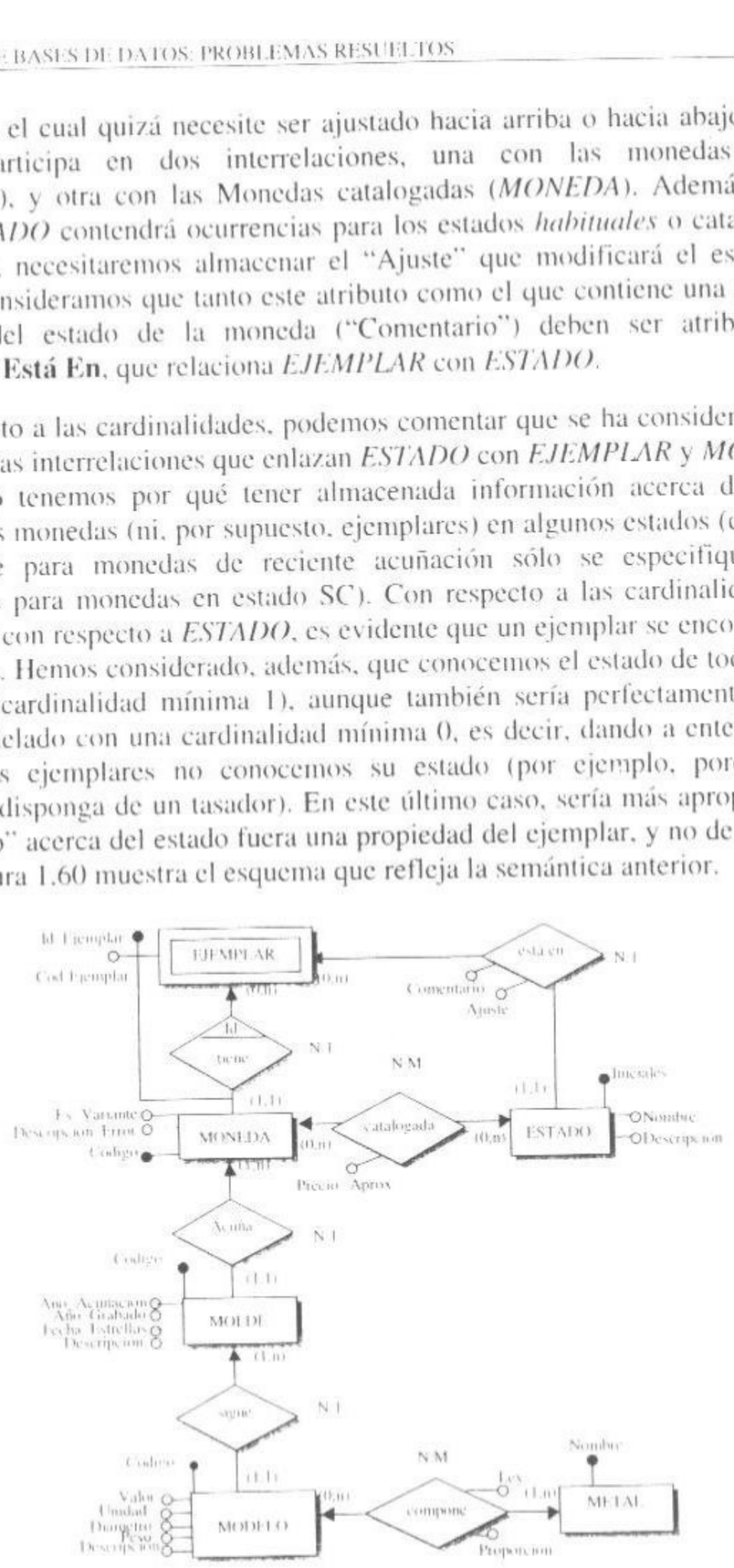


Figura 1.60

PARTE 3

> *"Los ejemplares podrán haber sido adquiridos a proveedores o conseguidas por cualquier otro medio (adquiridas en mercadillos, regaladas, llegadas a nuestro poder por el uso cotidiano, etc.). Aunque estamos interesados en almacenar (siempre que lo conozcamos) el precio y la fecha de adquisición del ejemplar, sólo estamos interesados en la procedencia de aquellos que fueron adquiridos a proveedores, de los que nos interesa conocer su nombre, dirección y teléfono de contacto.*
>
> *Asimismo, los ejemplares podrán ser vendidos a clientes. Cada ejemplar podrá ser vendido como máximo a un cliente, y no se admiten devoluciones. Interesa conocer la fecha y el precio de venta, y del cliente estamos interesados en su nombre, dirección, teléfono y el número de compras que nos ha realizado."*

Por último, nos falta por representar las ventas y compras de ejemplares de monedas por parte de los proveedores y clientes, respectivamente. En principio, podríamos representarlo como se muestra en la figura 1.61 (no se han incluido los atributos), es decir, limitarnos literalmente a las especificaciones del enunciado.

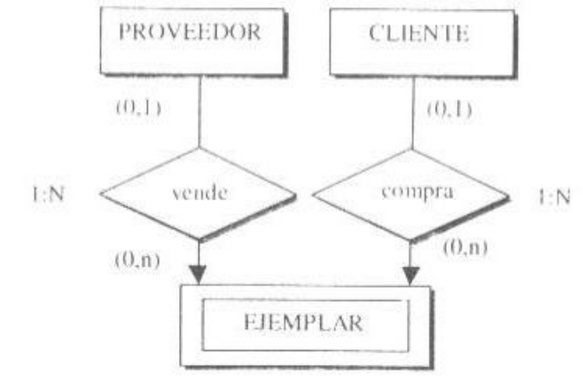


Figura 1.61

Sin embargo, podría pensarse en agrupar a las entidades *PROVEEDOR* y *CLIENTE* mediante una generalización. De hecho, en el enunciado podemos ver que ambos comparten los mismos atributos, excepto el atributo "Número de Compras", que es propio de los clientes.

Propuesta de solución

Por lo tanto, el esquema definitivo quedaría como se muestra en la figura 1.62. En el caso de los clientes y proveedores se ha añadido un "Código" como identificador principal.

El atributo de *CLIENTE* "Núm_Compras" es un atributo derivado.

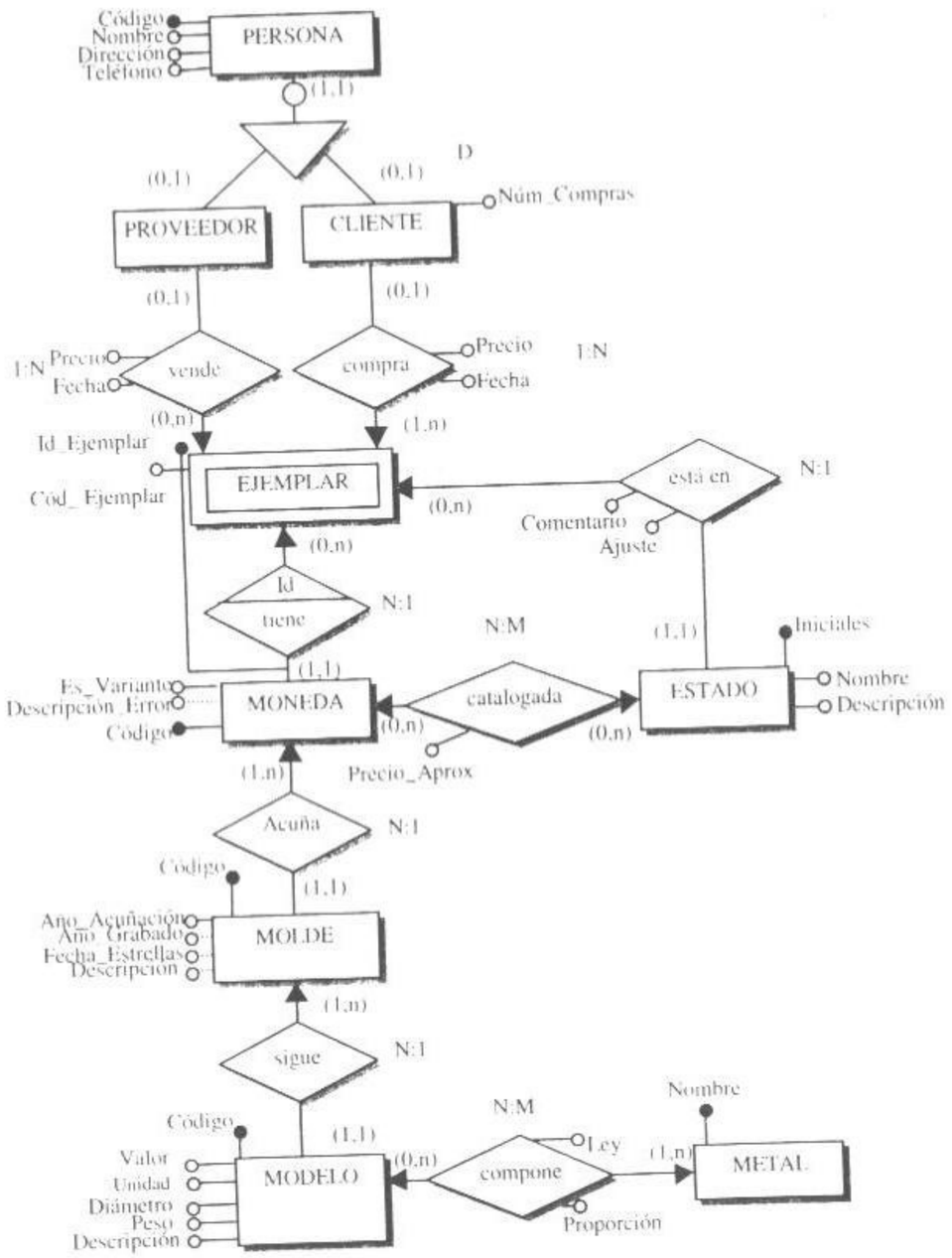


Figura 1.62

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- En el esquema no podemos representar que un ejemplar adquirido por un cliente no puede ser devuelto.
- Se supone que todo molde acuña al menos un tipo de moneda.
- [illegible] de clientes y proveedores.

- Deberíamos exigir que el atributo opcional que recoge la "Descripción del Error" sea obligatorio en el caso de que la moneda sea variante.

PROBLEMA 13: PARQUES NATURALES

Enunciado

La ministra de Medio Ambiente ha decidido crear un sistema de información sobre los parques naturales gestionados por cada comunidad autónoma. Después de realizar un detallado análisis, se ha llegado a las siguientes conclusiones:

Una comunidad autónoma (CA) puede tener varios parques naturales. En toda comunidad autónoma existe uno y sólo un organismo responsable de los parques. Un parque puede estar compartido por más de una comunidad.

Un parque natural se identifica por un nombre, fue declarado en una fecha, se compone de varias áreas identificadas por un nombre y caracterizadas por una determinada extensión. Por motivos de eficiencia se desea favorecer las consultas referentes al número de parques existentes en cada comunidad y la superficie total declarada parque natural en cada CA.

En cada área forzosamente residen especies que pueden ser de tres tipos: vegetales, animales y minerales. Cada especie tiene una denominación científica, una denominación vulgar y un número inventariado de individuos por área. De las especies vegetales se desea saber si tienen floración y en qué periodo se produce ésta; de las animales se desea saber su tipo de alimentación (herbívora, carnívora u omnívora) y sus periodos de celo; de las minerales se desea saber si se trata de cristales o de rocas. Además, interesa registrar qué especies sirven de alimento a otras especies, teniendo en cuenta que ninguna especie mineral se considera alimento de cualquier otra especie y que una especie vegetal no se alimenta de ninguna otra especie.

Del personal del parque se guarda el DNI, número de seguridad social, nombre, dirección, teléfonos (domicilio, móvil) y sueldo. Se distinguen los siguientes tipos de personal:

- Personal de gestión: registra los datos de los visitantes del parque y están destinados en una entrada del parque (las entradas se identifican por un número).
- Personal de vigilancia: vigila un área determinada del parque que recorre en un vehículo (tipo y matrícula).
- Personal de conservación: mantiene y conserva un área determinada del parque. Cada uno lo realiza en una especialidad determinada (limpieza, caminos...)

interrelación contiene el atributo "Individuos" que representa el número inventariado de cada especie en cada área del parque natural.

Las especies se clasifican en vegetales, animales y minerales; así, se tienen las entidades *ANIMAL*, *VEGETAL* y *MINERAL* que son subtipos exclusivos de la entidad *ESPECIE*. La entidad *ANIMAL* posee como atributos propios "Alimentación" (representa el tipo de alimentación de la especie), "Fecha Inicio" y "Fecha Fin" (fechas que representan el comienzo y el final del periodo de celo de la especie); la entidad *VEGETAL* se caracteriza por los atributos "Floración" (indica si la especie vegetal tiene floración o no) y "Fecha Inicio" y "Fecha Fin" (atributos opcionales que representan el periodo de floración en caso de que lo haya); por último, la entidad *MINERAL* contiene el atributo "Tipo" que puede tomar los valores cristal o roca.

Para reflejar qué especies sirven de alimento a otras especies se requieren dos interrelaciones. La interrelación reflexiva N:M **Come** relaciona la entidad *ANIMAL* consigo misma, de tal forma que una determinada especie animal puede servir de alimento a varias y una especie animal puede alimentarse, a su vez, de varias especies animales. Por otro lado, también existe una relación de alimento entre las especies animales y vegetales representada por la interrelación **Se Alimenta**; una especie animal puede alimentarse de varias especies vegetales y una especie vegetal puede servir de alimento a varias especies animales.

El esquema E/R correspondiente a los supuestos semánticos anteriores se muestra en la figura 1.64.

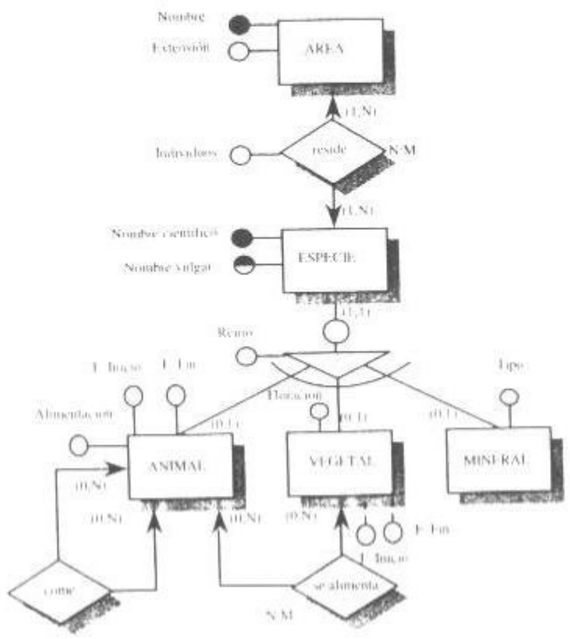


Figura 1.64

PARTE 3

"Del personal del parque se guarda el DNI, número de seguridad social, nombre, dirección, teléfonos (domicilio, móvil) y sueldo. Se distinguen los siguientes tipos de personal:

Personal de gestión: registra los datos de los visitantes del parque y están destinados en una entrada del parque (las entradas se identifican por un número).

Personal de vigilancia: vigila un área determinada del parque que recorre en un vehículo (tipo y matrícula).

Personal de conservación: mantiene y conserva un área determinada del parque. Cada uno lo realiza en una especialidad determinada (limpieza, caminos...)

Personal investigador: Tiene una titulación que ha de recogerse y pueden realizar (incluso conjuntamente) proyectos de investigación sobre una determinada especie."

En relación con los empleados que trabajan en los parques naturales se necesita la entidad *PERSONAL* caracterizada por los atributos "DNI" (identificador principal), "NSS" (número de la seguridad social que por ser único para cada empleado se considera identificador alternativo), "Nombre", "Dirección", "Sueldo" y el atributo multivaluado y opcional "Teléfonos" (pues un empleado puede tener más de un teléfono).

También existe una generalización total y exclusiva entre los empleados. Así, se distinguen las entidades *CONSERVACIÓN*, *VIGILANCIA*, *INVESTIGADOR* y *GESTIÓN* para los distintos tipos de personal. El subtipo *CONSERVACIÓN* se caracteriza por el atributo "Especialidad" (representa la función que hace el empleado); el subtipo *VIGILANCIA* posee los atributos "Tipo" y "Matrícula" (datos del vehículo que conduce); el subtipo *INVESTIGADOR* posee el atributo "Titulación" y, por último, el subtipo *GESTIÓN* no posee atributos propios pues la entrada del parque a la que están asignados será una nueva entidad, como se estudiará a continuación.

En cuanto a las interrelaciones, entre las entidades *PERSONAL* y *PARQUE NATURAL* existe la interrelación N:M **Trabaja** cuyas cardinalidades indican que una determinado empleado trabaja en un único parque natural y en un parque natural trabajan varios empleados (como mínimo uno). La interrelación 1:N **Conserva** entre *ÁREA* y *CONSERVACIÓN* refleja que un empleado de conservación está asignado a un único área y que un área está conservada por uno o varios empleados de conservación. La interrelación 1:N **Vigila** entre *ÁREA* y *VIGILANCIA* denota que un empleado de vigilancia controla una única área y que un área es controlada por uno o varios empleados de vigilancia.

Aunque definiremos después la entidad *PROYECTO*, estudiaremos ahora la interrelación ternaria **Realiza** entre *ESPECIE*, *PROYECTO* e *INVESTIGADOR* que refleja la participación de los empleados de investigación en diversos proyectos estudiando distintas especies. Las cardinalidades indican que un determinado investigador en un proyecto puede estudiar varias especies, una determinada especie en un proyecto puede ser estudiada por varios investigadores y, por último, una especie puede ser estudiada por un investigador en diversos proyectos.

Por último, se necesita la entidad *ENTRADA* para poder recoger las entradas que posee un determinado parque natural, así como las entradas que tienen asignadas los empleados de gestión. La entidad *ENTRADA* tiene como único atributo "Número Entrada" (identificador principal) y está relacionada con *PARQUE NATURAL* mediante la interrelación 1:N **Hay** (una entrada pertenece a un único parque y un parque tiene varias entradas). Además, existe la interrelación 1:1 **Controla** entre *ENTRADA* y *GESTIÓN* para reflejar que un empleado de gestión controla una única entrada del parque y que una determinada entrada tiene asignado un único empleado de gestión (suponemos que hay un único empleado de gestión en cada entrada).

La figura 1.65 muestra el esquema E/R correspondiente a los supuestos anteriormente descritos.

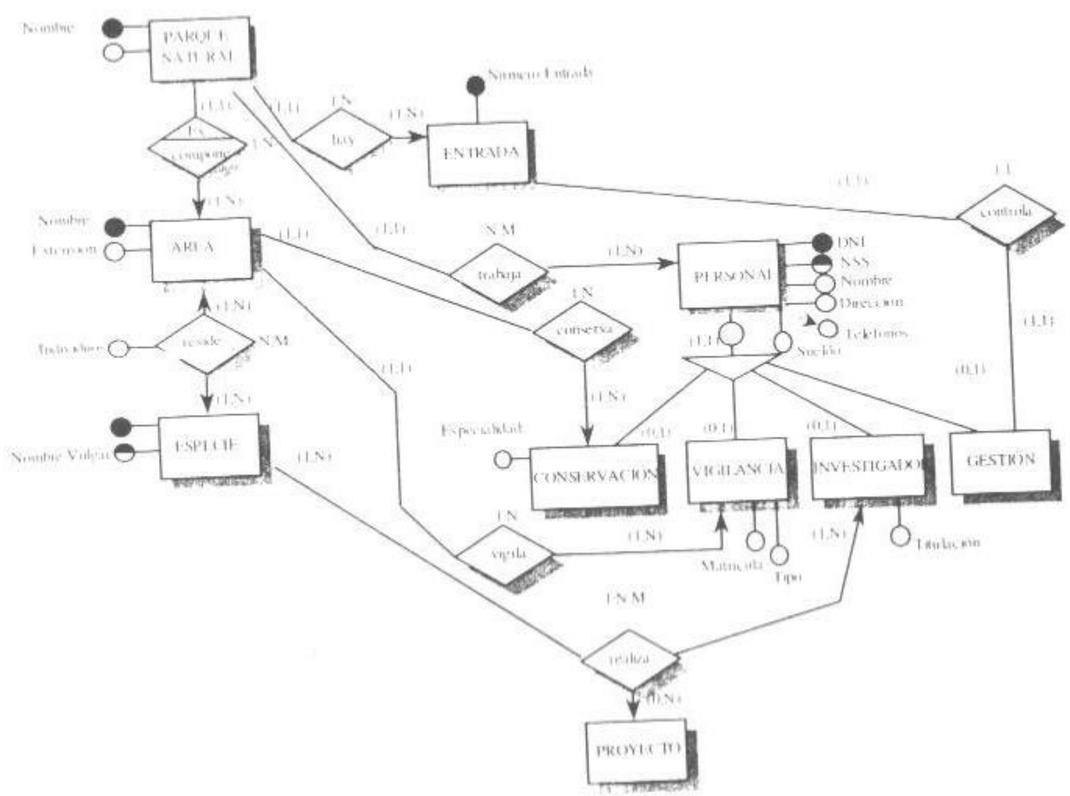


Figura 1.65

PARTE 4

> *"Un proyecto de investigación tiene un presupuesto y un periodo de realización".*

Este supuesto nos indica los atributos de la entidad *PROYECTO*; el atributo "Nombre" es el identificador principal y los atributos descriptivos son "Presupuesto" y "Fecha Inicio" y "Fecha Fin" para reflejar el comienzo y el final del periodo de realización.

> *"Un visitante (DNI, nombre, domicilio y profesión) debe alojarse dentro de los alojamientos de que dispone el parque; éstos tienen una capacidad limitada y tienen una determinada categoría".*

Se necesitan ahora la entidad *VISITANTE* caracterizada por los atributos "DNI" (identificador principal), *NOMBRE*, *DIRECCIÓN* y *PROFESIÓN* y la entidad *ALOJAMIENTO* con los atributos "Nombre" (identificador principal), "Capacidad" y "Categoría". La interrelación 1:N **Dispone** entre *PARQUE NATURAL* y *ALOJAMIENTO* recoge los alojamientos de los que consta un parque natural. Además, es una dependencia en existencia.

Aunque descrita de forma implícita en el enunciado, se requiere la interrelación N:M **Visita** entre *VISITANTE* y *PARQUE NATURAL* para recoger las visitas al parque. Puesto que un visitante puede visitar varias veces el mismo parque natural se ha añadido en la interrelación el atributo multivaluado "Fecha Visita" para poder distinguir las ocurrencias de la interrelación.

Por otro lado, también se tiene la interrelación N:M **Aloja** entre *VISITANTE* y *ALOJAMIENTO* para almacenar las estancias de determinados visitantes. Al igual que ocurría en la interrelación **Visita**, también se requieren los atributos multivaluados "Fecha Llegada", "Fecha Salida" y "Habitación" (un visitante puede alojarse en el mismo alojamiento en distintas fechas y en varias habitaciones).

Como se mencionó en la parte 3 también se almacenarán los registros de los visitantes en las entradas de los parques. La interrelación N:M **Registra** entre *ENTRADA* y *VISITANTE* recoge esta información (la interrelación contiene el atributo multivaluado "Fecha Registro" para indicar que un mismo visitante puede registrarse en la misma entrada en distintas ocasiones).

> *"Los alojamientos organizan excursiones al parque, en un vehículo o a pie, en determinados días de la semana y a una hora determinada. A estas excursiones puede acudir cualquier visitante del parque".*

Para almacenar la información de las excursiones se ha creado la entidad *EXCURSIÓN* con los atributos "Código Excursión", "Vehículo" (para indicar si es a pie o en vehículo), "Día" y "Hora". La interrelación **Organiza** entre *ALOJAMIENTO* y *EXCURSIÓN* recoge que un alojamiento puede organizar o no excursiones y que una excursión puede ser organizada por varios alojamientos.

Por último, la interrelación **Disfruta** refleja las excursiones que realizan los visitantes de los parques. Un determinado visitante puede que no realice ninguna excursión y una excursión puede ser realizada por varios visitantes.

El esquema E/R correspondiente a esta parte se muestra en la figura 1.66.

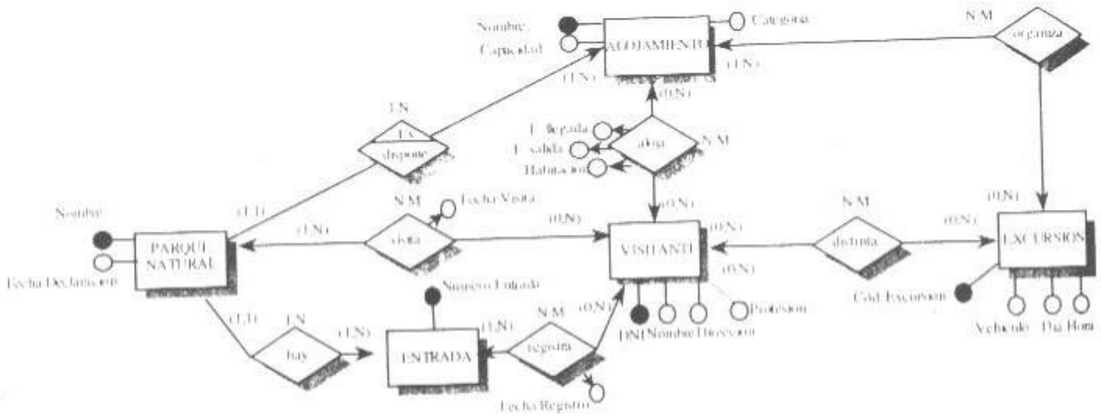


Figura 1.66

Observando la figura 1.66, existen tres ciclos que hay que analizar con el fin de detectar posibles redundancias. En el ciclo compuesto por las interrelaciones **Disfruta**, **Aloja** y **Organiza** no existen redundancias, pues las interrelaciones no están semánticamente relacionadas (el que un visitante se aloje en el parque no implica que vaya a realizar excursiones y viceversa).

En cuanto a las interrelaciones **Dispone**, **Aloja** y **Visita** ocurre algo similar (el que un visitante acuda a un parque no implica que vaya a alojarse en algún establecimiento).

En el ciclo entre las interrelaciones **Visita**, **Hay** y **Registra** sí existe redundancia, pues la semántica recogida en la interrelación **Visita** puede obtenerse a través de las interrelaciones **Hay** y **Registra**. Se deja al lector como ejercicio su comprobación.

Propuesta de solución

La figura 1.67 muestra el esquema E/R completo.

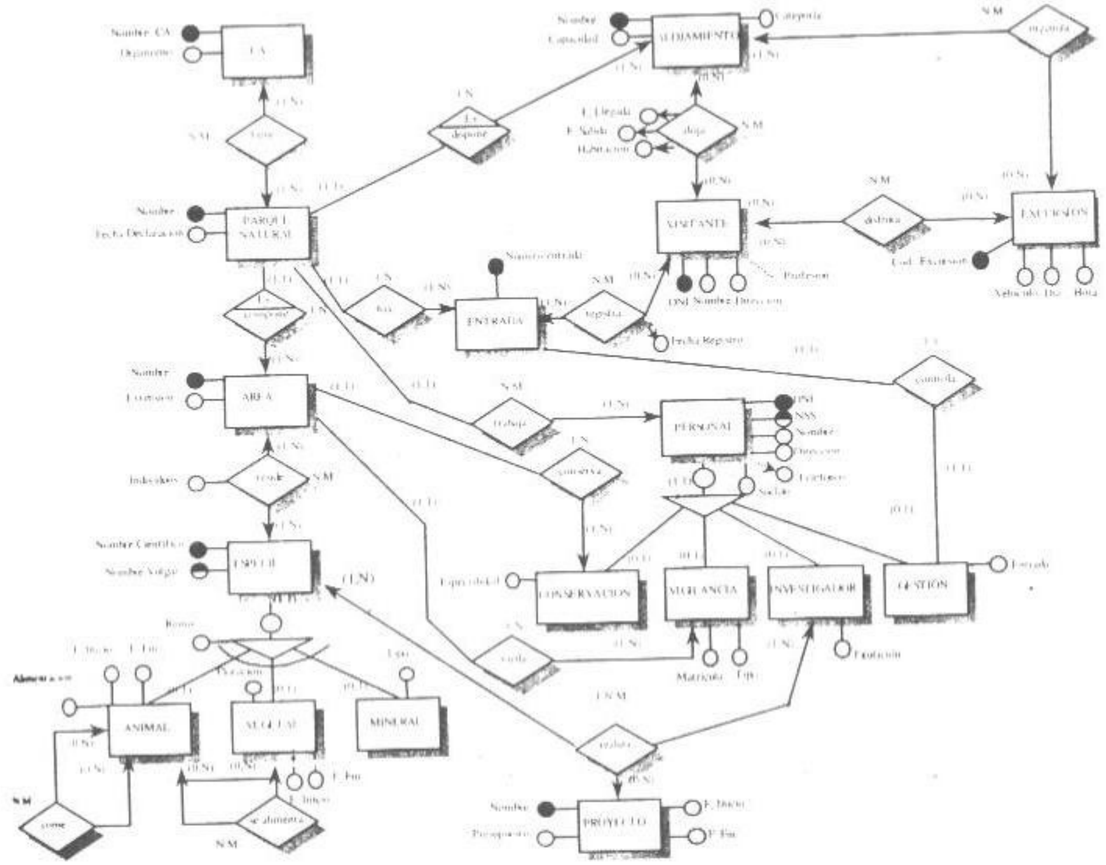


Figura 1.67

SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

No han sido necesario supuestos adicionales.

PROBLEMA 14: GESTIÓN DE HOSPITALES

Enunciado

Una compañía aseguradora de tipo sanitario desea diseñar una BD para informatizar parte de su gestión hospitalaria. En una primera fase sólo quiere contemplar los siguientes supuestos semánticos:

Los hospitales de sú red pueden ser propios o concertados; además de unos datos comunes a todos ellos como son el código de hospital (Cod_H), su nombre (N_H), número de camas (Num_C), etc., cuando el hospital es propio se tienen otros específicos como el presupuesto (P), tipo de servicio (TS), etc.

Una póliza, que se identifica por un número de póliza (Cod_P), tiene varios atributos que, en principio, no interesa especificar y que se agrupan bajo el nombre de datos de póliza (Datos_P). Una póliza cubre a varios asegurados, los cuales se identifican por un número correlativo (Num), añadido al código de la póliza, y tienen un nombre (NA), fecha de nacimiento (FN), etc.

Los asegurados cubiertos por una misma póliza pueden ser de distintas categorías. Mientras los asegurados de primera categoría (A1C) pueden ser hospitalizados e cualquier hospital, los de segunda categoría (A2C) sólo pueden ser hospitalizados en hospitales propios. Aunque las otras categorías no tienen derecho a hospitalización, en la BD se guardan todos los asegurados sea cual sea su categoría.

Interesa saber en qué hospitales han estado (o están) hospitalizados los asegurados, el médico que prescribió la hospitalización, así como las fechas de inicio (FI) y de fin (FF) de la misma.

Existen áreas, identificadas por un código (Cod_A) y con datos sobre su superficie (S), número de habitantes (NUM_H), etc. Los hospitales concertados tienen que estar asignados a una única área, que no puede cambiar, mientras que los propios no están asignados a áreas.

Los médicos, que se identifican por un código (Cod_M), tienen un nombre (N_M), teléfonos de contacto, etc. Interesa conocer las áreas a las que está adscrito un médico. Existe una dependencia jerárquica entre médicos de forma que un médico tiene un único jefe.

Discusión del enunciado

PARTE 1

> *"Los hospitales de su red pueden ser propios o concertados; además de unos datos comunes a todos ellos como son el código de hospital (Cod_H), su nombre (N_H), número de camas (Num_C), etc., cuando el hospital es propio se tienen otros específicos como el presupuesto (P), tipo de servicio (TS), etc."*

El enunciado nos dice que los hospitales pueden ser de dos tipos, y que existen algunos atributos que son comunes a todos los hospitales y otros que son propios de determinado tipo de hospital. Éste es el típico ejemplo que podemos modelar mediante una jerarquía. Parece deducirse del enunciado, y así lo supondremos, que un hospital

propio no puede ser concertado y viceversa, es decir, que ambos tipos son exclusivos, y que además no hay ningún otro tipo de hospital que nos interesa almacenar. Lo modelaremos por tanto como una jerarquía exclusiva y total, tal y como se muestra en la figura 1.68. Se ha añadido el atributo Tipo de Hospital ("Tipo_H"), que no aparece en el enunciado, y que nos permitirá identificar el tipo de hospital de que se trate.

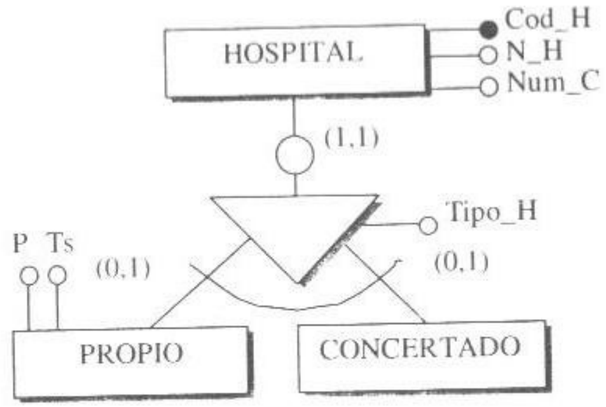


Figura 1.68

PARTE 2

> *"Una póliza, que se identifica por un número de póliza (Cod_P), tiene varios atributos que, en principio, no interesa especificar y que se agrupan bajo el nombre de datos de póliza (Datos_P). Una póliza cubre a varios asegurados, los cuales se identifican por un número correlativo (Num), añadido al código de la póliza, y tienen un nombre (NA), fecha de nacimiento (FN), etc."*

Una *PÓLIZA* la modelaremos mediante una entidad. Además, puesto que cada asegurado que cubre una póliza se identifica por un número correlativo incorporado al número de póliza, tenemos un claro ejemplo de dependencia en identificación entre *PÓLIZA* y *ASEGURADO*. Por ser una dependencia en identificación, las cardinalidades de la interrelación entre *ASEGURADO* y *PÓLIZA* serán (1,1). Esto nos implica que cada asegurado sólo puede estar en una póliza o, para ser más exactos, significa que cada ocurrencia de *ASEGURADO* está relacionada con una ocurrencia de *PÓLIZA*. Puesto que cada asegurado se identifica con un número correlativo añadido al de la póliza, podría ocurrir que una misma persona apareciera varias veces en distintas pólizas, aunque como ocurrencias de *ASEGURADO* distintas. Para poder recoger adecuadamente esta circunstancia deberíamos añadir supuestos nuevos, o bien modificar los existentes. Por ejemplo, se podría plantear a la compañía de seguros que identificara a cada asegurado por su DNI, con lo cual desaparecería la dependencia anterior, y no duplicaríamos la información de un asegurado que estuviera cubierto por varias pólizas.

PARTE 3

> *"Los asegurados cubiertos por una misma póliza pueden ser de distintas categorías. Mientras los asegurados de primera categoría (A1C) pueden ser hospitalizados en cualquier hospital, los de segunda categoría (A2C) sólo pueden ser hospitalizados en hospitales propios. Aunque las otras categorías no tienen derecho a hospitalización, en la BD se guardan todos los asegurados sea cual sea su categoría."*

Dejaremos de lado de momento las hospitalizaciones de los asegurados, ya que se nos aclarará en el párrafo siguiente, y nos centraremos ahora en la distinción entre tipos de asegurado. Como vemos en el enunciado, los asegurados pueden ser de distintas categorías, hecho que modelaremos mediante otra jerarquía. Parece deducirse de los requisitos, al igual que en el caso de los hospitales, que un asegurado no puede pertenecer a más de una categoría (no existe solapamiento). Sin embargo, el enunciado también nos indica explícitamente que existen otras categorías, por lo tanto la jerarquía no será total. La figura 1.69 recoge la semántica anterior.

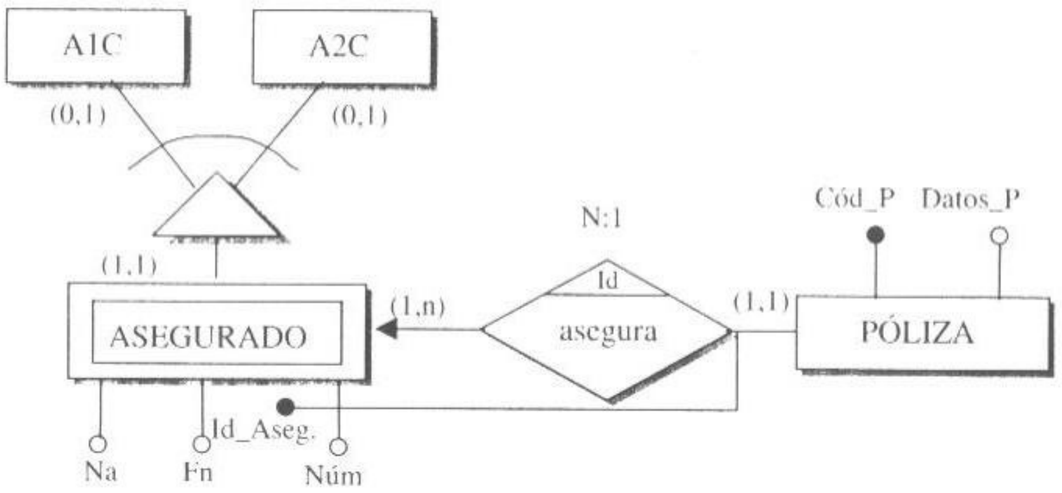


Figura 1.69

PARTE 4

> *"Interesa saber en qué hospitales han estado (o están) hospitalizados los asegurado', el médico que prescribió la hospitalización, así como las fechas de inicio (FI) y de fin (FF) de la misma."*

A continuación modelaremos los ingresos de los asegurados en los hospitales. Los requisitos especifican que los ingresos de los asegurados en los hospitales son prescritos por médicos. Esto podemos modelarlo mediante una interrelación ternaria (en realidad son dos interrelaciones ternarias, en una participan los hospitales propios y los asegurados de segunda categoría y en la otra todos los hospitales y los

asegurados de primera categoría, pero el razonamiento es equivalente en ambos casos). Cada interrelación ternaria no puede sustituirse por tres binarias, ya que no recogeríamos la semántica completa que nos especifica el enunciado: por ejemplo, podríamos saber en qué hospitales ha estado ingresado cada asegurado y qué médicos le han prescrito ingresos (cada uno de estos datos con una de las interrelaciones binarias), sin embargo no podríamos saber qué médico les prescribió cada ingreso en cada hospital. Las cardinalidades en todos los casos serán (1,n), ya que no existe en el enunciado ninguna restricción que nos imponga participaciones máximas de ninguna de las entidades. Por ejemplo, dado un asegurado ingresado por prescripción de un médico, puede haberlo sido en varios hospitales.

Además, como en los requisitos se nos dice que estamos interesados en conocer los hospitales en que han estado ingresados los asegurados (es decir, queremos almacenar un histórico), será imprescindible que las fechas de ingreso y de alta del hospital, modeladas como atributos de la interrelación, sean atributos multivaluados, ya que un asegurado puede ingresar en un mismo hospital prescrito por el mismo médico en distintas fechas. Supondremos, sin embargo, que este caso no puede darse en un mismo día, es decir, supondremos que un asegurado no puede ingresar en un mismo hospital prescrito por el mismo médico más de una vez el mismo día. Además, la fecha de alta será un atributo opcional, ya que no tomará ningún valor hasta que el asegurado no sea dado de alta en el hospital. Sería necesario controlar que la fecha de alta de un asegurado sea posterior a la de ingreso. Esto, sin embargo, no podemos reflejarlo en el esquema E/R.

PARTE 5

> *"Existen áreas, identificadas por un código (Cod_A) y con datos sobre su superficie (S), número de habitantes (NUM_H), etc. Los hospitales concertados tienen que estar asignados a una única área, que no puede cambiar, mientras que los propios no están asignados a áreas."*
>
> *"Los médicos, que se identifican por un código (Cod_M), tienen un nombre (N_M), teléfonos de contacto, etc. Interesa conocer las áreas a las que está adscrito un médico. Existe una dependencia jerárquica entre médicos de forma que un médico tiene un único jefe."*

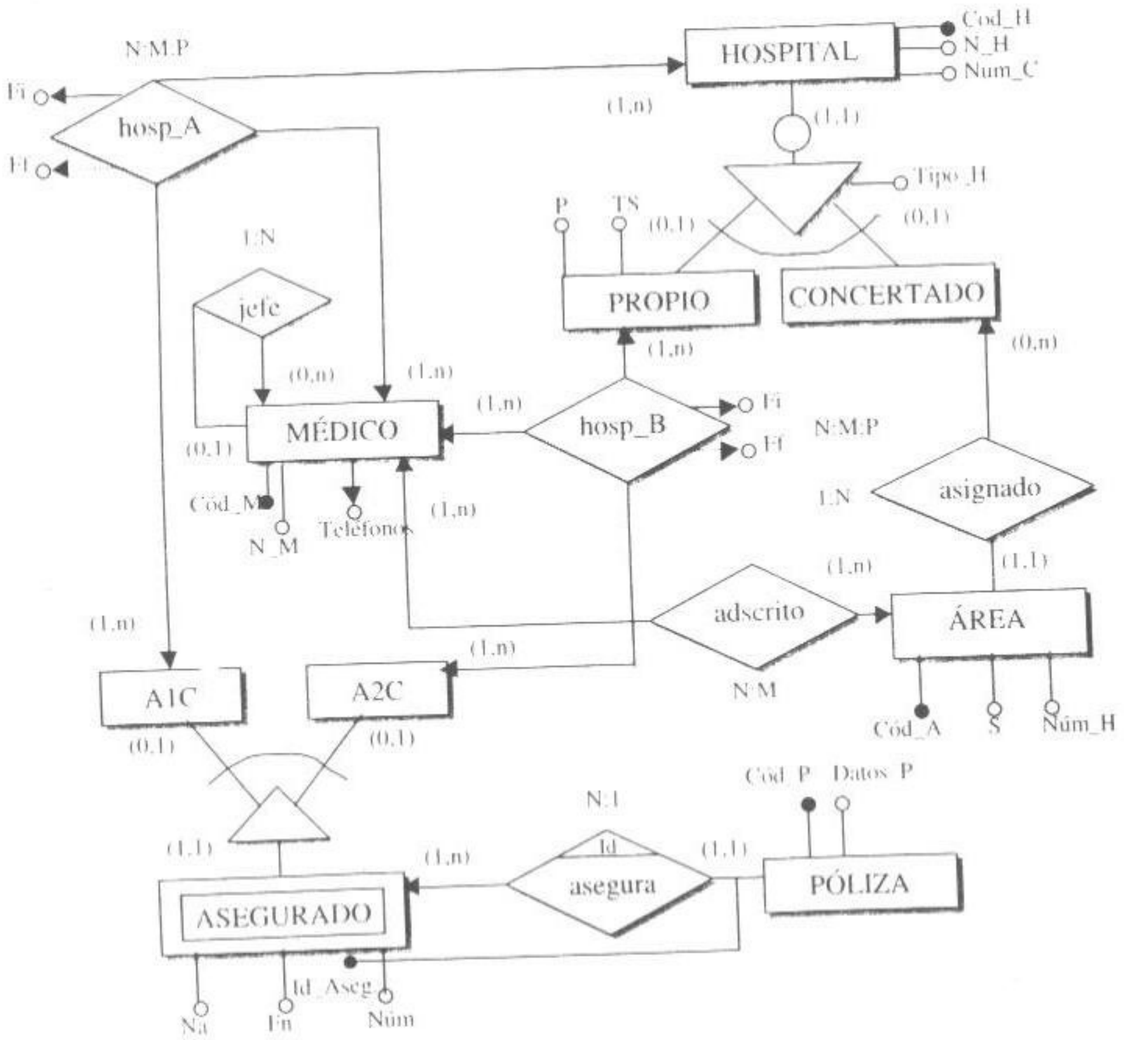
El *ÁREA* la modelaremos como una entidad, que estará relacionada, tal como nos indica el enunciado, con hospitales concertados. Supondremos que puede haber áreas sin hospitales (cardinalidad mínima 0), pero tal como nos exigen los requisitos, todo hospital concertado ha de estar asignado a una única área. En el esquema conceptual no podemos recoger que esta asignación no pueda cambiar, ya que para ello *CONCERTADO* debería ser una entidad débil, cosa que no tiene demasiado sentido.

Las áreas también están relacionadas con los médicos. Un *MÉDICO* será una entidad que tendrá una interrelación reflexiva **Jefe** que nos permitirá recoger que un médico tenga un único jefe (cardinalidades 0,1), y que un médico pueda ser jefe de

varios (o ninguno, mediante cardinalidades 0,n). La entidad *MÉDICO* tiene un atributo multivaluado "Teléfono", que podríamos haber modelado de otra manera, por ejemplo mediante una entidad relacionada con la entidad *MÉDICO* con tipo de correspondencia N:M (lo cual significaría que un mismo teléfono podrían compartirlo varios médicos). La adscripción de los médicos a las áreas la recogemos mediante la interrelación **Adscrito**. Las cardinalidades elegidas obligan a que todo médico esté adscrito al menos a un área, y que toda área esté adscrita al menos a un médico. De igual forma podríamos haber decidido que no fuera obligatorio que toda área tuviera al menos un médico y todo médico estuviera asignado al menos a un área, para ello las cardinalidades mínimas de la interrelación **Adscrito** serían 0.

Propuesta de solución

La figura 1.70 muestra la solución completa del ejercicio.



SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- No se puede recoger que un hospital concertado no puede cambiar el área a la que está asignado.
- No se puede recoger que la fecha de alta de un asegurado sea posterior a la de ingreso.
- Tampoco se puede recoger que el atributo "Núm" de la entidad *ASEGURADO* sea un número correlativo.

PROBLEMA 15: OLIMPIADAS DE INVIERNO

Enunciado

Como parte de la organización de las próximas olimpiadas de invierno, se decide la creación de un sistema de información para realizar la gestión de las pruebas de esquí. Del análisis realizado se obtiene la siguiente información:

Los juegos se componen de una serie de pruebas, en cada una de las cuales intervienen una serie de participantes. Cada participante en una prueba puede intervenir a título individual (esquiador individual) o bien formando parte de un equipo, en cuyo caso el participante será el equipo (no el esquiador). De cada esquiador (individual o de equipo) se desea tener el DNI, el nombre y la edad. A cada participante en una prueba (esquiador individual o equipo participante) se le asigna un código de participación dentro de la prueba (nombre de la prueba y un número secuencial). De cada equipo se mantiene un nombre, un entrenador, los esquiadores que lo componen y el número de éstos. El que un equipo participe en una prueba no significa que todos los esquiadores que lo componen intervengan en la misma. Un esquiador que forma parte de un equipo, no podrá cambiarse a otro ni actuar a título individual mientras duran los Juegos. Tampoco un esquiador individual podrá pasar a formar parte de un equipo.

Existen una serie de federaciones de esquí, cada una de las cuales tiene un nombre y un número de federados (en las federaciones se federan los esquiadores a título individual). Por un acuerdo existente entre las distintas federaciones, no se permite que ningún esquiador se federe en dos federaciones distintas. Tampoco se admite que participen esquiadores (ni a título individual ni formando parte de un equipo) que no estén federados.

Cada federación puede administrar una serie de estaciones de esquí, y toda estación se administrará al menos por una federación, aun cuando puede haber estaciones de esquí administradas conjuntamente por varias federaciones. Una estación de

esquí se identifica por un código, tiene un nombre, unas personas de contacto, una dirección, un teléfono y un número total de kilómetros esquiables, así como las pistas de las que dispone.

Dentro del sistema, cada pista se identifica a partir del código de la estación de esquí y un número secuencial. Se consideran también como pistas (para la realización de pruebas de largo recorrido) a varias de estas pistas (siempre de la misma estación) que por sus características físicas pudieran enlazarse. Así, por ejemplo, la pista diez estaría compuesta por las pistas dos y cuatro. Se requiere, para poder planificar las pruebas, mantener esta utilización combinada de las pistas. Para cada pista se mantiene también su longitud en kilómetros y su grado de dificultad (en la escala azul, verde, roja y negra).

La realización de cada prueba se desarrollará a lo largo de varias jornadas en una serie de pistas de una única estación. Los equipos o esquiadores individuales podrán competir en diferentes pruebas y en distintas pistas. Para cada participante en una prueba (equipo o esquiador individual) se registrará la fecha o fechas en que participa, el tiempo empleado y la posición obtenida; en el caso de equipos, estos datos se obtienen de los correspondientes a cada uno de los esquiadores del equipo que han intervenido en la prueba.

Cada prueba se identifica por un nombre, será de un tipo (fondo, slalom, salto...), tendrá unas fechas previstas de realización y se registrará el participante vencedor y el tiempo empleado por éste.

Discusión del enunciado

PARTE 1

"Los juegos se componen de una serie de pruebas, en cada una de las cuales intervienen una serie de participantes. Cada participante en una prueba puede intervenir a título individual (esquiador individual) o bien formando parte de un equipo, en cuyo caso el participante será el equipo (no el esquiador). De cada esquiador (individual o de equipo) se desea tener el DNI, el nombre y la edad. A cada participante en una prueba (esquiador individual o equipo participante) se le asigna un código de participación dentro de la prueba (nombre de la prueba y un número secuencial). De cada equipo se mantiene un nombre, un entrenador, los esquiadores que lo componen y el número de éstos. El que un equipo participe en una prueba no significa que todos los esquiadores que lo componen intervengan en la misma. Un esquiador que forma parte de un equipo, no podrá cambiarse a otro ni actuar a título individual mientras duran los Juegos. Tampoco un esquiador individual podrá pasar a formar parte de un equipo."

A partir del primer supuesto podemos detectar la existencia de dos conceptos distintos, pero muy relacionados: por una parte, los esquiadores y, por otra, los participantes en las pruebas. Los esquiadores serán deportistas que toman parte en los juegos, bien individualmente, o bien formando equipo con otros. Los participantes serán equipos o esquiadores individuales que participan en cada una de las pruebas. Es necesario distinguirlos, ya que, por ejemplo, un esquiador puede actuar como participante en varias pruebas. De igual forma, habrá equipos que podrán actuar como participantes en distintas pruebas y cuya composición será diferente en cada una de ellas.

Sin duda, habría distintas posibilidades de recoger esta semántica, pero una forma elegante y sencilla sería representarlos por separado, construyendo dos jerarquías. Una de ellas recogería los participantes, divididos entre aquellos que son participantes individuales (*P_INDIVIDUAL*) y los que son equipos participantes (*P_EQUIPO*). La otra recogería a los esquiadores, divididos entre los que únicamente pueden participar como individuos (*E_INDIVIDUAL*) y los que sólo pueden actuar en equipo (*E_EQUIPO*). También tendremos una entidad *EQUIPO*, que almacena los equipos, identificados por un "Nombre". Entre las entidades se establecen distintas interrelaciones que reflejan la semántica del enunciado. Así, tendremos una interrelación **Tiene** con una dependencia en existencia entre los equipos y los equipos participantes, y otra **Es** entre los esquiadores individuales y aquellos que son participantes. También existirán interrelaciones entre los esquiadores que forman parte de equipos y éstos (tanto con los equipos en sí como con los que son participantes). De esta forma podemos recoger tanto los esquiadores que componen los equipos (mediante la interrelación **Compuesto**) como la alineación del equipo en cada prueba, es decir, los esquiadores que efectivamente toman parte en esa prueba (mediante la interrelación **Forma_Parte**). Podría plantearse que la interrelación entre los equipos y los esquiadores de equipo también presentara una dependencia en existencia. La figura 1.71 muestra el esquema realizado hasta ahora.

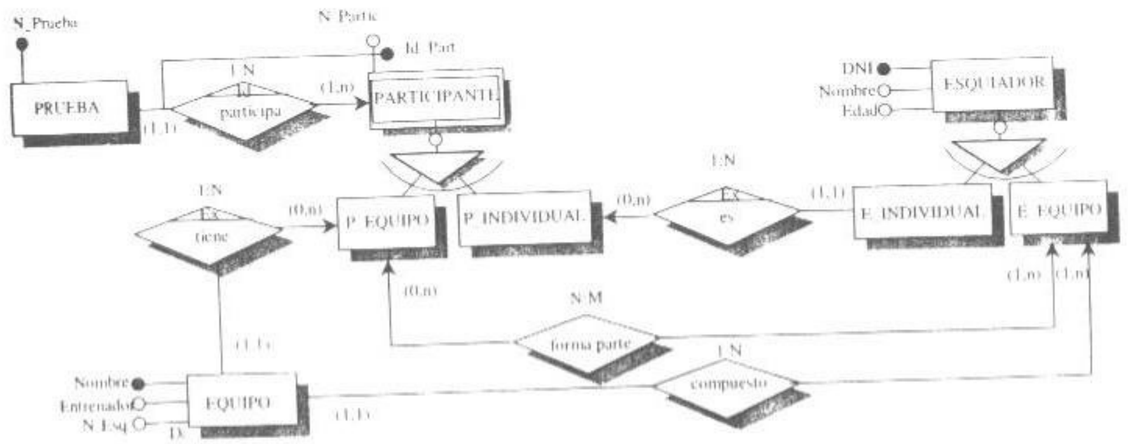


Figura.1.71

PARTE 2

"Existen una serie de federaciones de esquí, cada una de las cuales tiene un nombre y un número de federados (en las federaciones se federan los esquiadores a título individual). Por un acuerdo existente entre las distintas federaciones, no se permite que ningún esquiador se federe en dos federaciones distintas. Tampoco se admite que participen esquiadores (ni a título individual ni formando parte de un equipo) que no estén federados."

Este párrafo únicamente denota la existencia de una interrelación **Federa** 1:N entre los esquiadores y las federaciones a las que pertenecen. Mediante las cardinalidades (1,1) recogemos que todo esquiador está federado en una única federación y mediante las cardinalidades (0,n) permitimos tener almacenadas federaciones de las que no participa ningún esquiador en las olimpiadas. La figura 1.72 muestra la interrelación **Federa** entre *FEDERACIÓN* y *ESQUIADOR*.

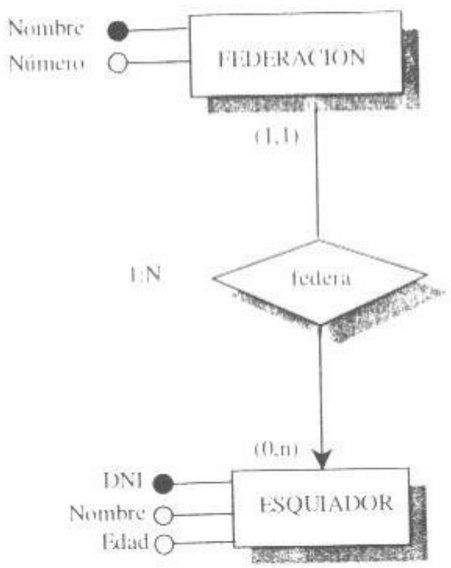


Figura 1.72

PARTE 3

"Cada federación puede administrar una serie de estaciones de esquí, y toda estación se administrará al menos por una federación, aun cuando puede haber estaciones de esquí administradas conjuntamente por varias federaciones. Una estación de esquí se identifica por un código, tiene un nombre, unas personas de contacto, una dirección, un teléfono y un número total de kilómetros esquiables, así como las pistas de las que dispone.

Dentro del sistema, cada pista se identifica a partir del código de la estación de esquí y un número secuencial. Se consideran también como pistas (para la

de la misma estación) que por sus características físicas pudieran enlazarse. Así, por ejemplo, la pista diez estaría compuesta por las pistas dos y cuatro. Se requiere, para poder planificar las pruebas, mantener esta utilización combinada de las pistas. Para cada pista se mantiene también su longitud en kilómetros y su grado de dificultad (en la escala azul, verde, roja y negra)."

Entre las federaciones y las estaciones habrá una interrelación N:M **Admin**, que recoja qué federación o federaciones administran cada estación. Por otra parte, entre la estación y las pistas que tiene, existe una dependencia en identificación, ya que para la identificación de una pista se necesita el código de la estación. El atributo que recoge los kilómetros esquiables de una estación será calculado a partir de las longitudes de las pistas que la componen, y el atributo que almacena las personas de contacto será multivaluado, para permitir que pueda existir más de una persona de contacto en cada estación.

Mediante la interrelación reflexiva **Contiene** recogeremos que una pista puede estar compuesta por otras. Sin embargo, no es posible recoger en el esquema que todas ellas pertenezcan a la misma estación. El atributo que almacena si una pista es o no compuesta será derivado, ya que es calculable. En cuanto al atributo "Longitud", que contiene los kilómetros de la pista, también será calculado, aunque sólo en el caso de las pistas compuestas de otras, lo cual tampoco podemos recogerlo en el esquema. El dominio del atributo "Grado_Dif" estará formado sólo por los cuatro valores permitidos (verde, azul, rojo y negro). La figura 1.73 muestra el esquema correspondiente a esta parte.

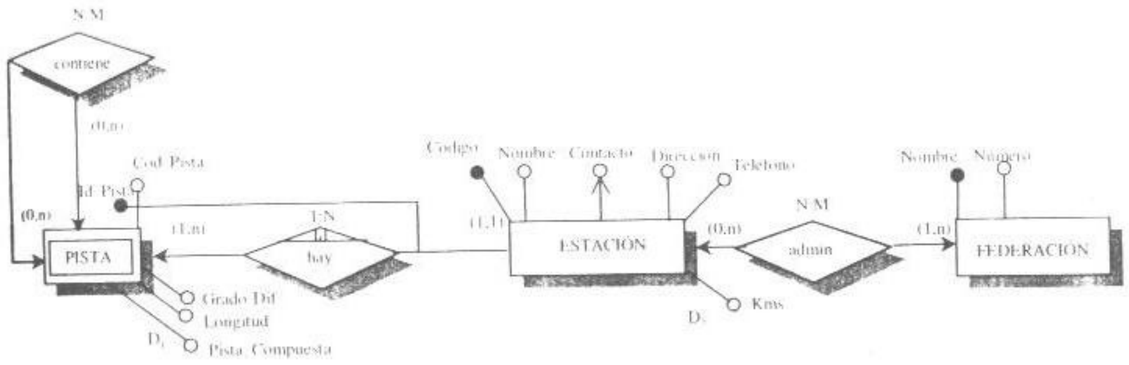


Figura 1.73

PARTE 4

"La realización de cada prueba se desarrollará a lo largo de varias jornadas en una serie de pistas de una única estación. Los equipos o esquiadores individuales podrán competir en diferentes pruebas y en distintas pistas. Para cada participante en una prueba (equipo o esquiador individual) se registrará la fecha o fechas en que participa, el tiempo empleado y la posición obtenida; en el caso de equipos, estos datos se obtienen de los

correspondientes a cada uno de los esquiadores del equipo que han intervenido en la prueba.

Cada prueba se identifica por un nombre, será de un tipo (fondo, slalom, salto...), tendrá unas fechas previstas de realización y se registrará el participante vencedor y el tiempo empleado por éste."

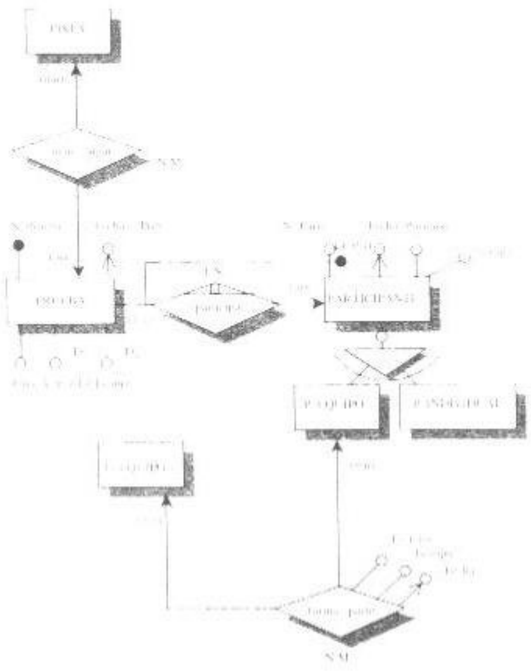


Figura 1.74

A partir de los párrafos anteriores, aparte de algunos atributos nuevos de los que hablaremos a continuación, únicamente añadiremos la interrelación **Tiene_Lugar**, que contendrá las pistas en las que se celebran las pruebas. No podemos recoger la restricción exigida en el enunciado de que una prueba se celebre en pistas de la misma estación. Podría pensarse en una interrelación ternaria entre las entidades *ESTACIÓN*, *PISTA* y *PRUEBA*. Sin embargo, puede verse fácilmente que con la semántica vista para las interrelaciones ternarias esto no es posible.

Como acabamos de decir, en los párrafos anteriores aparecen algunos atributos nuevos que deberemos recoger. En concreto, la fecha o fechas en las que participa un participante en una prueba, el tiempo empleado y la posición obtenida. En principio, estos atributos los ponemos en la entidad *PARTICIPANTE* (que, como sabemos, engloba tanto a participantes individuales como a los que son equipos). Sin embargo, de esta manera no recogemos estos datos para los esquiadores componentes de un equipo (en el enunciado se nos dice que para el caso de los equipos los datos se calculan a partir de los de los esquiadores que lo forman). Una posible solución, que es la que aparece en la figura 1.74, es incorporar estos atributos también en la

interrelación **Forma_Parte**. De esta manera, los atributos de **Forma_Parte** contendrán los valores para el caso de los esquiadores que forman parte de un equipo participante, y los atributos de *PRUEBA* los valores para el caso de los participantes individuales o de equipo. En el caso de estos últimos serán calculados a partir de los de **Forma_Parte**. Sin embargo, al igual que ocurría en el caso de la longitud de una pista, no podemos representar en el esquema que un atributo sea derivado sólo en determinados casos.

Por otra parte, también añadiremos cuatro atributos nuevos a la entidad prueba: "Tipo de Prueba" (fondo, slalom, salto, etc.), "Fechas Previstas de Realización" (que será multivaluado), "Participante Vencedor" y "Tiempo Empleado" por éste. Estos dos últimos atributos serán derivados, y podrán tomar valor nulo, ya que antes de celebrarse la prueba no tendrán ningún valor.

Propuesta de solución

La figura 1.75 muestra el esquema definitivo.

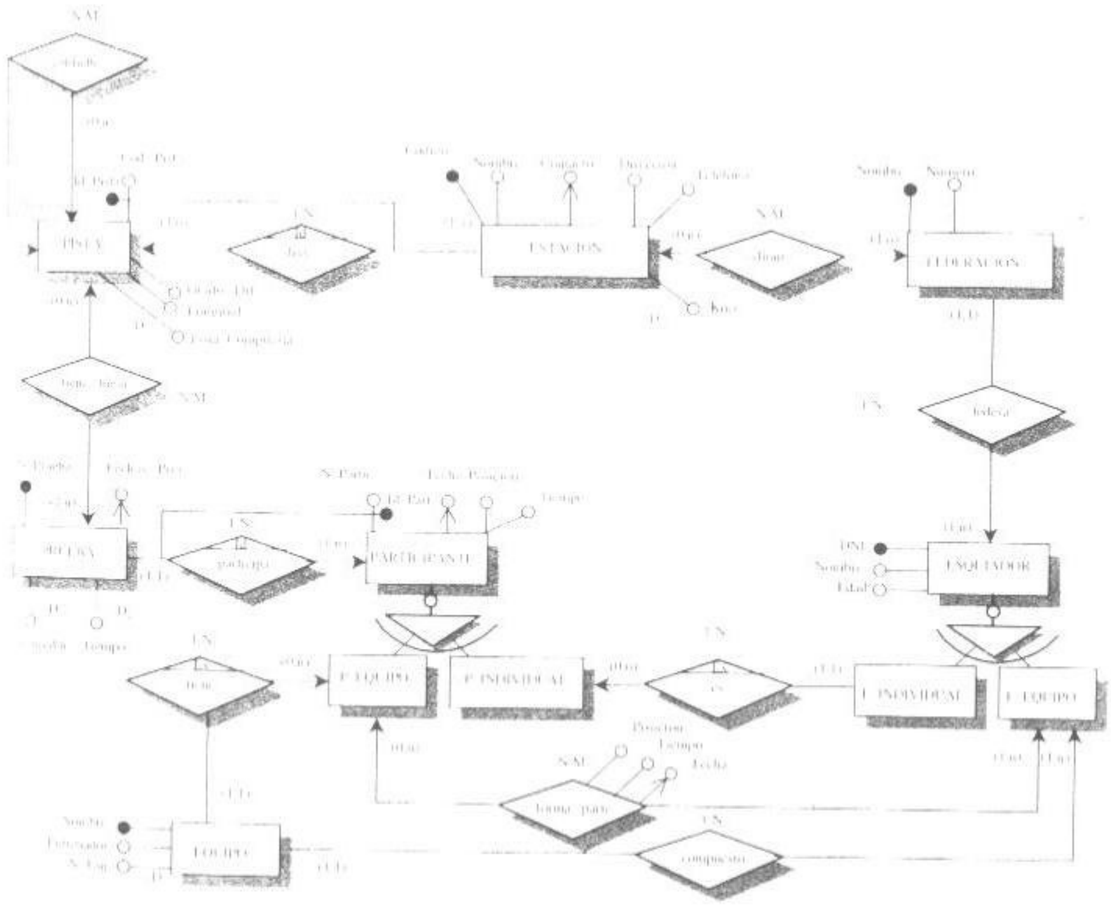


Figura 1.75

UPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO EFLEJADA

Aunque el esquema recoge que en un momento dado un esquiador únicamente es individual o de equipo, no recoge que no pueda cambiar de equipo, o que pase de esquiador de equipo a esquiador individual o al revés.

Tampoco se recoge que una pista compuesta esté formada por pistas de la misma estación, ni que una prueba se celebre en pistas de la misma estación.

El esquema no recoge la secuencialidad del "Número de Participante".

Como ya hemos dicho, en el esquema no se puede recoger que un atributo pueda ser derivado sólo en determinadas ocasiones.

CAPÍTULO 2

DISEÑO LÓGICO: MODELO RELACIONAL

INTRODUCCIÓN

Recordatorio de la estática del Modelo Relacional

El elemento central del modelo relacional es la Relación. Una relación tiene un nombre, un conjunto de atributos que representan sus propiedades y un conjunto de tuplas que incluyen los valores que cada uno de los atributos toma para cada elemento de la relación. Una relación se representa como una tabla de dos dimensiones (las columnas son los atributos de la relación y las filas son las tuplas) con un único valor en cada celda intersección. A continuación se muestra un ejemplo de una relación denominada ALUMNO que consta de los atributos "Cod_matricula", "Nombre", "Ciudad" y "Cod_grupo" (nombres de las columnas de la tabla). Cada fila de la tabla muestra una tupla correspondiente a los datos de un alumno:

ALUMNO

COD_MATRICULA	NOMBRE	CIUDAD	COD_GRUPO
101	Juan Montero	Alcorcón	11
102	Alicia Cristóbal	Leganés	11
202	Ana Vallejo	Leganés	21
300	Ignacio López	Móstoles	31
103	Leticia Martínez	Alcorcón	--

Los atributos de la relación se definen sobre dominios formados por un conjunto de valores homogéneos y atómicos. Por ejemplo, el dominio del "Cod_matrícula" es códigos, formado por el conjunto de cadenas de longitud 3 caracteres. También es posible definir dominios por extensión (enumerando los posibles valores de los que consta el dominio) o bien indicando un rango de valores.

El modelo relacional impone una serie de *restricciones inherentes*:

a) En una relación no puede haber dos tuplas iguales (obligatoriedad de clave primaria).

b) El orden de las tuplas y el de los atributos no es relevante.

c) Cada atributo sólo puede tomar un único valor del dominio sobre el cual está definido (no hay grupos repetitivos).

d) Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo (regla de integridad de entidad).

Por otra parte, los mecanismos que proporciona el modelo relacional para recoger *restricciones semánticas* o de *usuario* son:

a) La restricción de *clave primaria* (PRIMARY KEY) permite declarar un atributo o conjunto de atributos como la clave primaria de una relación (identifica unívocamente cada tupla de la relación). En el ejemplo mostrado anteriormente de la relación ALUMNO la clave primaria es el atributo "Cod_matrícula".

b) La restricción de *unicidad* (UNIQUE) nos permite definir claves alternativas (los valores de uno o varios atributos no pueden repetirse en diferentes tuplas de una relación).

c) La restricción de *obligatoriedad* (NOT NULL) permite declarar si uno o varios atributos de una relación deben tomar siempre un valor, es decir, no pueden tomar valores nulos.

d) La restricción de *clave ajena* (FOREIGN KEY), también denominada *integridad referencial*, se utiliza para, mediante claves ajenas (conjunto de atributos en una relación que es una clave primaria en otra o la misma relación) enlazar relaciones de una base de datos. La integridad referencial nos indica que los valores de la clave ajena en la relación hijo deben corresponderse con los valores de la clave primaria en la relación padre o bien ser nulos si se admiten nulos. Los atributos que son clave ajena en una relación no necesitan tener los mismos nombres que los atributos de la clave primaria con la cual ellos se corresponden. A continuación se muestra un ejemplo de integridad referencial con las relaciones EMPLEADO y DEPARTAMENTO:

EMPLEADO (NOMBRE, DEPARTAMENTO, SALARIO, FECHA_NAC, EXT_TELEFÓNICA)

DEPARTAMENTO (NUMERO_DEPT, NOMBRE)

Las claves primarias de la relación hija EMPLEADO y de la relación padre DEPARTAMENTO son "Nombre" y "Número_dept", respectivamente. El atributo "Departamento" de la relación EMPLEADO es una clave ajena que referencia la relación DEPARTAMENTO, es decir, los valores del atributo "Departamento" deben corresponderse con los valores del atributo clave primaria "Número_dept" de la relación DEPARTAMENTO. Algunas tuplas de estas relaciones se muestran a continuación:

EMPLEADO

NOMBRE	DEPARTAMENTO	SALARIO	FECHA_NAC	EXT_TLF
Pablo Montero	14	220.000	10-11-67	6543
Beatriz Cristóbal	13	300.000	20-9-68	6577
J. Luis Martín	11	150.000	25-6-77	6433
Almudena López	13	350.000	4-5-60	6422
Ángel Vallejo	14	400.000	15-4-72	6321
Pedro García	11	200.000	12-3-70	6323

DEPARTAMENTO

NUMERO_DEPT	NOMBRE
11	Contabilidad
13	Marketing
14	Informática

La integridad referencial asegura que los empleados de la relación EMPLEADO sólo pueden trabajar en departamentos que hayan sido dados de alta en la relación DEPARTAMENTO.

Además de la integridad referencial que nos permite enlazar relaciones entre sí dando lugar a la estructura de la BD, el modelo relacional permite también definir las opciones de borrado y modificación en las claves ajenas. Estas opciones indican las acciones que hay que llevar a cabo cuando se produce un borrado o modificación de una tupla en la relación padre referenciada por una relación hija. Las posibilidades para una operación de actualización (borrado o modificación) son:

- Borrado/modificación en cascada (CASCADE): El borrado (o modificación) de una tupla en la relación padre ocasiona un borrado (o modificación) de todas las tuplas relacionadas en la relación hija (tuplas cuya clave ajena coincida con el valor de la clave primaria de la tupla eliminada o modificada en la relación padre). Así, en el ejemplo anterior de

Empleados y Departamentos, el borrado de un departamento supone un borrado de todos los empleados que trabajan en él.

- Borrado/modificación restringido (RESTRICT): En este caso si existen tuplas en la relación hija relacionadas con la tupla de la relación padre sobre la que se realiza la operación, entonces no se permitirá llevar a cabo dicha operación. En el ejemplo anterior, no se podría borrar un departamento que tenga empleados que trabajen en él.

- Borrado/modificación con puesta a nulos (SET NULL): Esta posibilidad nos permite poner el valor de la clave ajena referenciada a NULL cuando se produce el borrado o modificación de una tupla en la relación padre. Continuando con el ejemplo anterior, el borrado de un departamento implicaría que todos los empleados que trabajan en él pasarán a contener NULL en el atributo DEPARTAMENTO. Evidentemente, esta opción está permitida siempre y cuando la clave ajena admita nulos.

- Borrado/modificación con puesta a un valor por defecto (SET DEFAULT): Su funcionamiento es similar al caso anterior, con la excepción de que el valor al que se ponen las claves ajenas referenciadas es un valor por defecto que se habrá especificado en la definición de la tabla correspondiente. En este caso, en el ejemplo anterior, cuando se borra un determinado departamento es posible asignar sus empleados a un departamento ficticio (que debe encontrarse evidentemente presente en la relación DEPARTAMENTO).

Las opciones de borrado y modificación pueden ser distintas para una determinada clave ajena de una relación; por ejemplo, es posible definir el borrado en cascada y la modificación restringida.

e) Restricciones de *Verificación* (CHECK): En algunos casos puede ocurrir que sea necesario especificar una condición que deben cumplir los valores de determinados atributos de una relación de la BD aparte de las restricciones ya vistas de clave primaria, unicidad, obligatoriedad y clave ajena. Para ello se definen las restricciones de verificación que, al igual que las ya estudiadas, siempre llevan implícitas un rechazo en caso de que no se cumpla la condición especificada y que también se comprueban ante una inserción, borrado o modificación. Como las restricciones de verificación siempre van ligadas a un único elemento de la BD (generalmente una relación) no es necesario que tengan un nombre. Por ejemplo, para la relación EMPLEADO podría definirse una restricción sobre el atributo "Salario" que estableciera que "el rango del salario de un empleado puede oscilar entre las 100.000 y las 500.000 ptas. Así, si se va a insertar un empleado con un sueldo superior a las 500.000 ptas. la operación se rechazaría.

f) Aserciones (ASSERTION): Otro tipo de restricción, que generaliza al anterior, lo forman las aserciones en las que la condición se establece sobre elementos de distintas relaciones (por ello debe tener un nombre que la identifique). Su funcionamiento es idéntico al de las restricciones de verificación. La única diferencia consiste en que las restricciones de verificación o CHECK tienen como ámbito una única relación. Continuando con nuestro ejemplo, se podría especificar una aserción para la restricción que establezca que "no hay ningún empleado que trabaje en el departamento de contabilidad que gane más de 200.000 ptas."

En las restricciones que involucran algún tipo de subconsulta en la condición (por ejemplo, si se utiliza alguna función de agregación sobre las tuplas de una relación) puede ocurrir que puedan implementarse como un CHECK. Sin embargo, es necesario estudiar que la restricción no pueda ser violada en ningún caso por cualquiera de las operaciones que puedan realizarse sobre el esquema. Por ello, a veces es necesario implementarlas como una Aserción, Ullman y Widom (1997).

g) *Disparadores* (TRIGGER): Por último, a veces puede interesar especificar una acción distinta del rechazo cuando no se cumple una determinada restricción semántica[16]. En ese caso, se recurre al uso de los disparadores o *triggers* que nos permiten además de indicar una condición, especificar la acción que queremos se lleve a cabo si la condición se hace verdadera. Los disparadores pueden interpretarse como reglas de tipo evento-condición-acción (ECA) que pueden interpretarse como reglas que especifican que cuando se produce un evento, si se cumple una condición, entonces se realiza una determinada acción. Los disparadores no están soportados en el SQL-92; sin embargo, existen productos relacionales que los soportan aunque con diferencias en el modo de funcionamiento. En nuestro ejemplo, se podría definir un disparador que "informará al administrador de la BD cuando haya un empleado que gane más de 450.000 ptas. y que trabaje en el departamento de Marketing". Como veremos más adelante también son convenientes para mantener algunas restricciones del esquema E/R que no tienen contrapartida en el modelo relacional.

Notación

Un esquema relacional se representará mediante un grafo, conocido como grafo relacional; básicamente, se trata de un grafo dirigido cuyos nodos son las relaciones de

[16] Al contrario que en las restricciones declarativas vistas hasta ahora en las que se rechaza la operación de inserción, borrado o modificación si NO se cumple la condición, en el caso de los disparadores la acción especificada en el cuerpo del disparador se ejecuta siempre y cuando se cumpla la condición especificada.

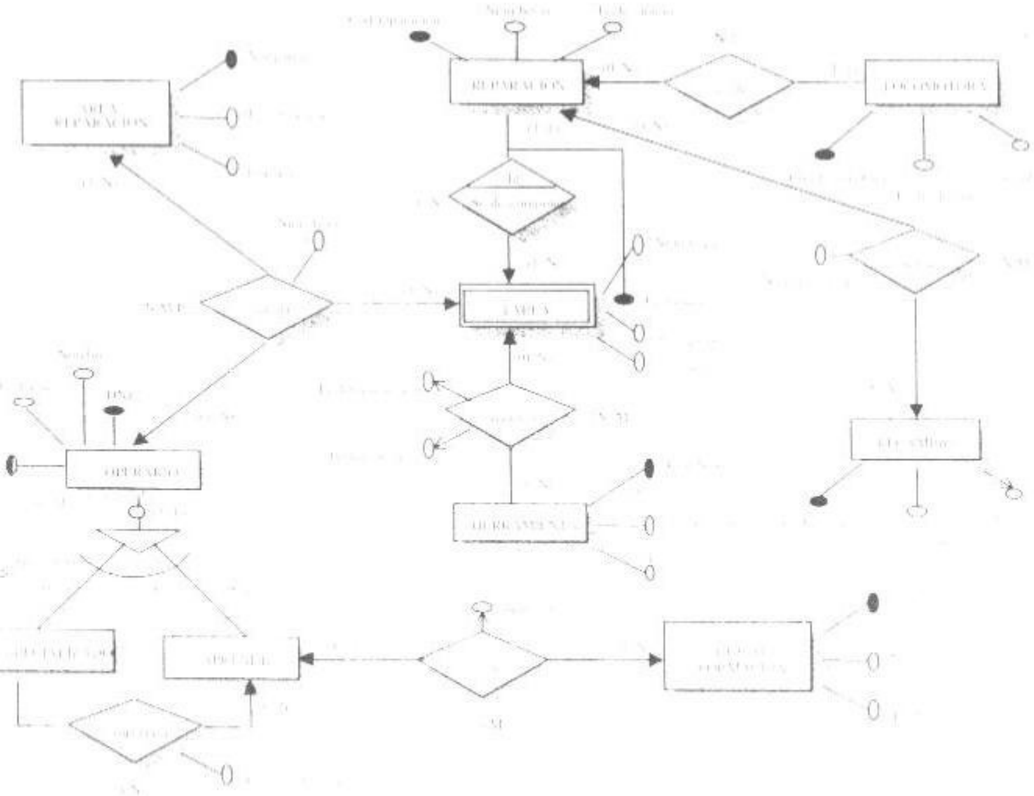


Figura 2.1. Esquema E/R

TRANSFORMACIÓN DE ENTIDADES, ATRIBUTOS Y DOMINIOS

Cada *Entidad* del esquema E/R dará lugar a una nueva relación cuya clave primaria es el Identificador Principal de la Entidad.

Cada *Atributo* de una Entidad se transforma en un atributo de la relación creada para la Entidad, aunque hay que tener en cuenta sus distintos tipos de restricciones semánticas:

Atributos Univaluados: Dan lugar a un atributo de la relación. En la figura 2.1, la entidad *LOCOMOTORA* da lugar a una relación con el mismo nombre y con tres columnas: "Cod_locomotora", "Fecha_inicio" y "Nombre".

Atributos Multivaluados: Dan lugar a una nueva relación cuya clave primaria es la concatenación de la clave primaria de la Entidad en la que se sitúa el atributo multivaluado más el nombre del atributo multivaluado. En ocasiones, si el atributo multivado no admite repeticiones, es suficiente éste como clave primaria. Por ejemplo, en la figura 2.1, la entidad *RECAMBIO* tiene el atributo multivaluado Proveedor; su transformación al modelo relacional daría lugar a una relación RECAMBIO cuya clave primaria es "Cod_recambio" y una relación

PROVEEDOR cuya clave primaria es la concatenación de "Cod_recambio" y "Proveedor".

- *Atributos Obligatorios*: Atributos con la restricción de NOT NULL.
- *Atributos Opcionales*: Atributos que pueden tomar valores nulos.
- *Identificador Principal*: Atributo(s) que forman la clave primaria.
- *Identificador Alternativo*: Atributo con la restricción de UNIQUE. El atributo "Nº_Seg Social" de la entidad *OPERARIO* (Figura 2.1) es un identificador alternativo de operario.
- *Atributos Derivados*: Atributos cuyos valores se obtienen como resultado de algún cálculo sobre otros atributos. En la figura 2.1 la entidad *REPARACIÓN* tiene el atributo derivado "Num_horas" que se obtiene de la suma de la duración de todas las tareas que tiene asociadas una determinada reparación.
- *Atributos compuestos*: Se transforman en los atributos que los componen (no hay forma de representar un atributo compuesto en el modelo relacional).
- Los *Dominios* se transforman en los dominios del modelo relacional.

TRANSFORMACIÓN DE INTERRELACIONES N:M

Las interrelaciones N:M dan lugar a una relación cuya clave será la concatenación de los Identificadores Principales de las Entidades que enlaza la interrelación. De esta forma los atributos que forman la clave primaria en esta nueva relación son claves ajenas respecto a las relaciones en las que estos atributos son clave primaria. Por ejemplo, en la figura 2.1 existe la interrelación N:M **Incluye** y su transformación al modelo relacional da lugar a una relación INCLUYE cuya clave primaria es la concatenación de las claves primarias de REPARACIÓN y RECAMBIO, es decir, "Cod_reparación" y "Cod_recambio".

Si la interrelación tiene atributos, éstos pasarán a formar parte de la relación creada para la interrelación. Por ejemplo, la interrelación **Incluye** tiene el atributo "Número_piezas"; la relación INCLUYE creada para esta interrelación contiene, además de los atributos que forman la clave primaria, una columna denominada "Número_piezas".

En el caso de que la interrelación contenga un atributo multivaluado que no denote una dimensión temporal, la clave de la relación deberá incluir también este atributo. Por ejemplo, en la figura 2.1 la interrelación **Asiste** entre *APRENDIZ* y *CURSO FORMACION* contiene el atributo multivaluado Puntuación que refleja las distintas calificaciones obtenidas por un aprendiz en cada uno de los cursos a los que

3D y los arcos representan las restricciones de clave ajena, y en el que aparecerán más de las distintas relaciones con sus atributos y las restricciones de clave ajena restricciones de clave primaria, unicidad y obligatoriedad. Las convenciones pleadas para la representación de este grafo son:

El nombre de las tablas está representado en mayúsculas (al igual que los butos), y en negrita. Primero aparece el nombre de la relación y a continuación sus butos entre paréntesis.

- Las claves primarias aparecen subrayadas.
- Las claves alternativas aparecen en negrita.
- Las claves ajenas están representadas en letra cursiva y referencian a la relación en la que son clave primaria mediante una flecha.
- Los atributos que pueden tomar valores nulos aparecen con un asterisco.
- Las opciones para la integridad referencial son:
  - B:C, Borrado en Cascada.
  - B:N, Borrado con puesta a Nulos.
  - B:D, Borrado con puesta a valor por defecto.
  - B:R, Borrado Restringido.
  - M:C, Modificación en Cascada.
  - M:N, Modificación con puesta a Nulos.
  - M:D, Modificación con puesta a valor por defecto.
  - M:R, Modificación Restringida.

glas de transformación de un esquema E/R a un esquema acional

Las tres reglas básicas empleadas para transformar un esquema conceptual E/R en esquema relacional son:

- Toda entidad se transforma en una relación
- Las interrelaciones N:M se transforman en una relación
- Las interrelaciones 1:N dan lugar o bien a una propagación de clave o bien a una relación.

A continuación, se describirán brevemente estas tres reglas, empleando para ello el esquema E/R que se muestra en la figura 2.1 y que se corresponde con la siguiente descripción:

> *Una compañía ferroviaria necesita una Base de Datos para gestionar su taller en el que se llevan a cabo las revisiones periódicas y reparaciones de las locomotoras. La información que debe contener la Base de Datos se describe en los siguientes supuestos semánticos.*
>
> *El taller dispone de varias áreas de reparación (motores diesel, motores eléctricos, transmisiones, etc.) caracterizadas por un número, una localización y un tamaño en metros cuadrados.*
>
> *Las reparaciones de las locomotoras (código de locomotora, fecha de comienzo de funcionamiento y nombre) se desglosan en un conjunto de tareas a las que se asignan diversas áreas de reparación (zonas del taller). Una tarea se caracteriza por su nombre (desmontaje, limpieza, engrase, montaje, etc.), su descripción y su duración, y se puede realizar en una o varias áreas de reparación. Cada área de reparación puede ocuparse para realizar distintas tareas de reparación.*
>
> *Las tareas de reparación son llevadas a cabo por varios operarios (tanto especializados como aprendices). Interesa conocer quiénes son los operarios asignados a una determinada tarea de reparación y área de reparación. De igual manera, hay que recoger las áreas de reparación ocupadas para llevar a cabo las tareas y los operarios involucrados en ellas (incluyendo las horas dedicadas por cada operario en cada tarea y área de reparación).*
>
> *De cada reparación se desea guardar un código, nombre, la duración en horas y la fecha de comienzo. Es importante, además, saber los recambios (tuercas, válvulas, etc.) que se han empleado en la reparación de una determinada locomotora, queriendo conocer de cada recambio el código, descripción, nombre de los proveedores que suministran cada recambio y número de unidades empleadas en cada reparación. Así mismo, se quiere almacenar las herramientas (código de herramienta, descripción y peso) empleadas en cada tarea de reparación junto con las fechas de inicio y fin de utilización de esas herramientas.*
>
> *Existen operarios especializados y aprendices; de los especializados se quiere conocer los años de experiencia, su DNI, nombre, fecha de nacimiento y nº de la Seguridad Social. Los operarios especializados supervisan a los aprendices, interesando almacenar esta información. Un operario aprendiz solo tiene un supervisor y se quiere guardar la fecha en que se inició este seguimiento. De los operarios aprendices se almacenará su DNI, nombre, fecha de nacimiento y nº de la Seguridad Social.*
>
> *Por último, los aprendices asisten a cursos de formación (código de curso, nombre y fecha de inicio) y en cada curso los aprendices son evaluados con varias puntuaciones.*

ha asistido. La transformación al modelo relacional de esta interrelación da lugar a una relación ASISTE cuya clave primaria contiene la concatenación de "DNI", "Cod_curso" y puntuación; sin embargo, puesto que un aprendiz podría obtener iguales calificaciones en un determinado curso, sería necesario añadir un atributo "Fecha_evaluación" para distinguirlas y que también formaría parte de la clave primaria.

Sin embargo, en el caso de atributos con una dimensión temporal (generalmente atributos que denotan fechas, horas o intervalos de tiempo), tanto si son multivaluados como univaluados, es necesario estudiar la semántica del universo del discurso con el fin de determinar cuáles van a ser los atributos que formen la clave primaria de la relación a la que da lugar la interrelación. Para ello habrá que estudiar si se trata de interrelaciones que representan información histórica o no. Por ejemplo, en la figura 2.1, la interrelación **Requiere** entre *TAREA* y *HERRAMIENTA* tiene dos atributos multivaluados ("Fecha_inicio_uso" y "Fecha_fin_uso") que nos indican que una herramienta puede utilizarse en una tarea de reparación en diferentes intervalos de tiempo. La transformación de esta interrelación da lugar a la relación REQUIERE cuya clave primaria está compuesta por los atributos "Cod_reparación" y "Num_tarea" procedentes de *TAREA*[17], "Cod_herramienta" procedente de *HERRAMIENTA* y el atributo "Fecha_inicio_uso", es decir, tarea, herramienta y fecha de inicio determinan unívocamente cada tupla de la relación REQUIERE. Si en una fecha determinada una herramienta sólo pudiera utilizarse en una tarea, entonces la clave de TAREA no formaría parte de la clave de la relación REQUIERE.

También es necesario tener en cuenta cuáles son las cardinalidades mínimas y máximas en cada lado de la interrelación para no perder semántica en la transformación. Estos casos se irán estudiando a medida que vayan surgiendo en los ejercicios propuestos.

TRANSFORMACIÓN DE INTERRELACIONES 1:N

En el caso de las interrelaciones 1:N existen dos posibilidades de transformación:

1. Crear una nueva relación para la interrelación, cuyo tratamiento sería igual que el de las interrelaciones N:M, con la salvedad de que la clave primaria de la nueva relación constará del Identificador Principal de la Entidad que se encuentra en el lado N de la interrelación. Si la interrelación tiene atributos propios, ésta es una solución razonable. Por ejemplo, en la figura 2.1, la interrelación **Supervisa** podría transformarse en una relación SUPERVISA cuya clave primaria es el atributo "DNI_aprendiz" (pues un operario aprendiz tiene un único supervisor) y que consta además de los atributos "DNI_especializado" y "Fecha_comienzo".

[17] La transformación de Dependencias en Identificación se estudiará en la sección dedicada a las extensiones del Modelo E/R.

2. Propagar el Identificador Principal desde la Entidad que se encuentra en el lado 1 a la Entidad que se encuentra en el lado N (propagación de clave); si existen atributos en la interrelación, éstos también se proparagarán. En ausencia de información relativa al volumen de valores nulos que se manejarán o de si existen posibilidades de que la interrelación evolucione o no a una interrelación N:M, esta solución de propagación de clave será la que se adopte en la resolución de los ejercicios propuestos en este libro. Por ejemplo, en la figura 2.1, la interrelación **Es_de** entre las entidades *REPARACIÓN* y *LOCOMOTORA* se transforma en una propagación de clave ("Cod_locomotora") desde la relación LOCOMOTORA hacia la relación REPARACIÓN que constaría de los siguientes atributos: "Cod_reparación" (clave primaria), "Núm_horas", "Fecha_inicio", "Cód_locomotora" (clave ajena).

No hay que olvidar cómo controlar las cardinalidades mínimas y máximas de la interrelación para no perder semántica. Estos casos se irán estudiando a medida que vayan surgiendo en los ejercicios propuestos.

Transformación de otros elementos del Modelo E/R extendido

A continuación se describen brevemente la transformación de las extensiones del modelo E/R. Un análisis más detallado se encuentra en De Miguel *et al.* (1999).

TRANSFORMACIÓN DE DEPENDENCIAS EN EXISTENCIA Y EN IDENTIFICACIÓN

Una interrelación 1:N de Dependencia en Existencia origina que la clave ajena, propagada desde la entidad fuerte a la entidad débil, deba tener la opción de borrado en Cascada, es decir, no puede existir ninguna ocurrencia de la entidad hijo que no esté relacionada con una ocurrencia de la entidad padre.

Una interrelación 1:N de Dependencia en Identificación[18] da lugar a una propagación de clave desde la entidad fuerte a la entidad débil de tal forma que la entidad débil requiere de la clave de la entidad fuerte para su identificación. En la figura 2.1, la dependencia en identificación denotada por la interrelación **Se_descompone** hace que la clave de REPARACIÓN ("Cod_reparación") se propague a la relación TAREA de tal forma que la clave primaria de esta última está formada por los atributos "Cod_reparación" y "Num_tarea". La relación TAREA consta, además, de los atributos "Descripción" y "Duración".

[18] Recuérdese que una Dependencia en Identificación siempre implica una Dependencia en Existencia; sin embargo, una Dependencia en Existencia no tiene por qué implicar una Dependencia en Identificación.

TRANSFORMACIÓN DE GENERALIZACIONES

Aunque existen tres posibilidades para la transformación de jerarquías al modelo relacional, en los casos propuestos en este libro se adoptará preferentemente la que se describe a continuación. Se creará una relación por cada Entidad participante en la jerarquía (una relación para el supertipo y otra para cada uno de los subtipos), de tal forma que el supertipo propaga su Identificador Principal a cada uno de los subtipos que pasan a identificarse por el mismo identificador (como clave ajena). La relación creada para el supertipo podrá contener el atributo discriminante de la jerarquía. Por ejemplo, en la figura 2.1, la generalización entre el supertipo *OPERARIO* y los subtipos *ESPECIALIZADO* y *APRENDIZ* da lugar a tres relaciones: la relación OPERARIO contiene los atributos "DNI" (clave primaria), "Nombre", "Fecha_nac", "N°_Seg_Social" y "Tipo". La relación ESPECIALIZADO tiene como clave primaria el atributo "DNI" (clave ajena) y, además, tiene el atributo propio "Años experiencia". Por último la relación APRENDIZ tiene como único atributo su clave primaria, "DNI", que también es clave ajena.

Las restricciones semánticas de totalidad/parcialidad y exclusividad/solapamiento también hay que controlarlas en la transformación al modelo relacional. La totalidad se controla prohibiendo las inserciones en el supertipo y con un disparador que ante una inserción en un subtipo inserte también en el supertipo (además el atributo discriminante no podría tomar valores nulos). La generalización de la figura 2.1 es total, pues no puede haber un operario que no sea ni especializado ni aprendiz; así, cuando se hace una inserción en la relación *OPERARIO* hay que forzar a una inserción en el subtipo correspondiente. En el caso de la parcialidad, el atributo discriminante puede tomar valores nulos y no sería necesario ningún disparador.

En cuanto a la exclusividad se requiere una aserción que compruebe que, si un ejemplar pertenece a uno de los subtipos, entonces no puede pertenecer a los demás. Si se permite solapamiento también es necesaria una aserción similar a la de la exclusividad pero que dé cabida a nuevos valores del atributo discriminante para los casos de solapamiento comprobando que las ocurrencias están en los subtipos adecuados. Por ejemplo, en la figura 2.1, la exclusividad entre *ESPECIALIZADO* y *APRENDIZ* puede controlarse con una aserción de la siguiente forma:

```
CREATE ASSERTION Controlar_exclusividad_generalización
Check ((Tipo='especializado' AND EXIST ESPECIALIZADO
AND NOT EXIST APRENDIZ)
  OR
  (Tipo='aprendiz' AND EXIST APRENDIZ
        AND NOT EXIST ESPECIALIZADO))
```

TRANSFORMACIÓN DE INTERRELACIONES DE GRADO SUPERIOR A DOS

Hasta el momento sólo se han tratado interrelaciones binarias. Si en el esquema E/R se encuentran interrelaciones de grado superior (ternarias, cuaternarias, etc.) se requiere hacer un estudio exhaustivo de las cardinalidades mínimas y máximas para llevar a cabo la transformación. Se mostrarán las reglas de transformación para el caso de las interrelaciones ternarias, aunque podría extenderse a las de grado superior a 3.

El caso general de transformación de interrelaciones ternarias N:M:P establece que la interrelación da lugar a una relación cuya clave primaria es la concatenación de los Identificadores Principales de las entidades que relaciona. En el caso de que las cardinalidades sean (1,N), (1,N) y (1,N) no es necesario ningún mecanismo adicional para no perder semántica en la transformación al relacional. La interrelación ternaria **Trabaja** de la figura 2.1 da lugar a una relación TRABAJA cuya clave primaria es la concatenación de las claves primarias de ÁREA REPARACIÓN, OPERARIO y TAREA y que son "Num área", "DNI", "Cod_reparación" y "Num_tarea". Además, esta relación contiene también el campo "Num_horas".

Sin embargo, es necesario estudiar las cardinalidades mínimas y máximas en los tres lados de la interrelación para realizar la transformación al relacional sin perder semántica, por ejemplo, en el caso de que alguna cardinalidad mínima sea 0 o en el caso de que algún lado de la interrelación tenga cardinalidad (1,1). Estas distintas posibilidades se estudiarán en la resolución de los problemas propuestos en este libro.

TRANSFORMACIÓN DE INTERRELACIONES EXCLUSIVAS

En el caso de interrelaciones exclusivas, éstas se transforman como se ha visto en este mismo apartado, pero es necesario añadir un *check* que compruebe que si un ejemplar de la entidad participa ya en una ocurrencia de una interrelación, entonces no puede participar en ninguna ocurrencia de la otra interrelación.

Algunos aspectos sobre la pérdida de semántica en la transformación al modelo relacional

Existen algunas restricciones que es necesario controlar con mecanismos externos al modelo relacional. Aquí es necesario establecer la diferencia entre el modelo relacional y los SGBD comerciales que no implementan el modelo relacional completo. Por ello, como se ha mencionado en los apartados anteriores dedicados a la transformación al modelo relacional, aunque aquí se hable de *checks* y aserciones como mecanismos del modelo relacional para implementar ciertas restricciones, estos mecanismos no siempre están disponibles en los sistemas gestores de BD, lo que implica que hay que recurrir a otros medios para recoger esas restricciones (por ejemplo, disparadores presentes en la BD, procedimientos almacenados, aplicaciones externas etc.). Se dará aquí enumeración de las restricciones de los esquemas E/R que es necesario contemplar en la transformación al modelo relacional mediante *checks*, aserciones o disparadores:

- Cardinalidades mínimas de 1 en interrelaciones N:M y 1:N (excluyendo aquellas que se controlan con la restricción NOT NULL cuando se realiza propagación de clave).

- Cardinalidades máximas conocidas en interrelaciones binarias N:M y 1:N e interrelaciones ternarias.
- Exclusividad en las generalizaciones.
- Inserción y borrado en las generalizaciones.
- Atributos derivados.
- Exclusividad entre interrelaciones.
- Atributos multivaluados obligatorios.

Por último, puede mejorarse la calidad del diseño resultante introduciendo otras icciones que no figuraban en el enunciado original pero que se consideran cuadas o convenientes (por ejemplo, restricciones que implican operadores de paración de fechas).

ámica del Modelo Relacional: Álgebra Relacional

En los apartados anteriores se ha estudiado la parte estática del modelo relacional. asaremos ahora brevemente la dinámica del mismo desde el punto de vista del bra relacional. No estudiaremos aquí la sintaxis del lenguaje SQL-92, pues en los blemas propuestos en este capítulo se mostrarán sentencias para la creación y ipulación de una BD en este lenguaje. Remitimos al lector a De Miguel y Piattini 99) para profundizar en estos aspectos.

Los operadores del álgebra relacional se aplican a relaciones dando como ltado nuevas relaciones. Originariamente Codd definió ocho operadores en dos pos:

Las operaciones tradicionales de conjuntos: unión, intersección, diferencia y producto cartesiano.

Las operaciones relacionales especiales: restricción, proyección, combinación y división.

La intersección, combinación y división se pueden derivar de otros operadores.

Una operación del álgebra relacional consiste en:

O - Operador del álgebra relacional (unión, combinación, etc.)
r - Relación(es)[19] a la(s) que se aplica el operador O
r' - Relación resultado de la operación

tal que O(r) = r'

[19] Se trataría de una o a lo sumo dos relaciones: en el primer caso se hablaría de operador unario y segundo de operador binario.

La figura 2.2 muestra los operadores primitivos y derivados del álgebra relacional y la figura 2.3 una representación gráfica de estos operadores.

OPERADORES DEL ÁLGEBRA RELACIONAL

Operadores primitivos

Binarios (propios de la teoría de conjuntos)
- Unión (UNION)
- Diferencia (DIFFERENCE)
- Producto cartesiano (CARTESIAN PRODUCT)

Unarios (nuevos para el álgebra relacional)
- Proyección (PROJECT)
- Selección o restricción (SELECTION o RESTRICTION)

Operadores derivados

Combinación (JOIN)
- Combinación **θ**
- Combinación Natural
- Combinación Externa
- Semicombinación

Intersección (INTERSECTION)
División (DIVISION)
Agrupación (GROUP BY)
Cierre transitivo (TRANSITIVE CLOSURE)

Figura 2.2. Operadores del álgebra relacional

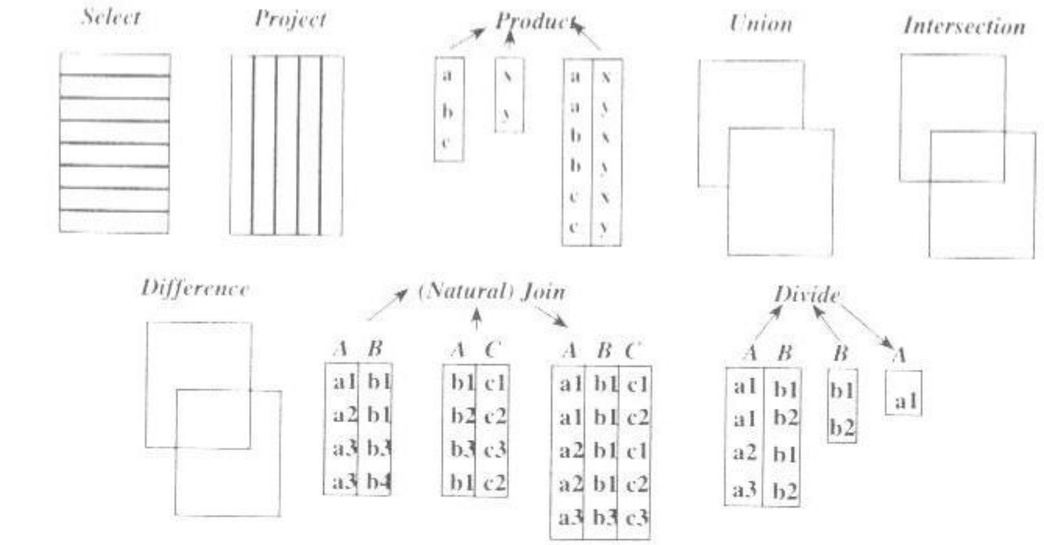


Figura 2.3. Representación gráfica de los operadores del álgebra relacional

Veamos en qué consiste cada uno de estos operadores.

Restricción o selección (σ): Extrae las tuplas de una relación dada que satisfagan una condición especificada. Por ejemplo, seleccionar los empleados cuya F_Nac < 01/01/1972 de la tabla EMPLEADO:

$$\sigma_{\text{F_nac} < 01/01/1970}(\text{EMPLEADO})$$

Proyección (Π): Extrae los atributos especificados de una relación dada minando las tuplas duplicadas. Por ejemplo, seleccionar los nombres de los pleados de la relación EMPLEADO:

$$\pi_{\text{nombre}}(\text{EMPLEADO})$$

Unión (U): Construye una relación con todas las tuplas que aparezcan en alquiera de las dos relaciones especificadas. Para usar este operador es necesario e las dos relaciones especificadas tengan los mismos atributos definidos sobre los smos dominios.

Diferencia (-): Construye una relación con las tuplas de la primera relación que aparecen en la segunda. Para utilizar este operador es necesario que las dos aciones especificadas tengan los mismos atributos definidos sobre los mismos minios.

Producto Cartesiano (x): A partir de dos relaciones especificadas, construye una ación concatenando cada tupla de la primera con cada una de las tuplas de la gunda. La relación resultante está definida sobre la unión de los atributos de las dos laciones especificadas.

Combinación (Join): A partir de las relaciones especificadas, construye una lación concatenando cada tupla de la primera relación con cada una de las tuplas de segunda, siempre que ambas tuplas satisfagan la condición dada. Si la condición es igualdad y en la relación resultante se elimina el atributo común, se denomina ombinación natural. En la combinación externa se incluyen las tuplas de una relación ue no están relacionadas con otra relación, concatenando con nulos.

La combinación (θ) es un producto cartesiano seguido de una restriccion. La ombinación natural (*) es un producto cartesiano seguido de restricción por igualdad proyección. Un ejemplo se muestra en la figura 2.4.

(Natural) Join

R1

A	B
a	u
b	v
c	w

*

R2

A	C
a	x
a	y
b	z

=

R

A	B	C
a	u	x
a	u	y
b	v	z

Figura 2.4. Combinación natural

Intersección (∩): Construye una relación con aquellas tuplas que aparezcan en las dos relaciones especificadas. Para usar este operador es necesario que las dos relaciones especificadas tengan los mismos atributos definidos sobre los mismos dominios. La intersección se puede definir en función de la unión y la diferencia.

División (:): Construye una relación con los valores de un atributo de la primera relación (dividendo) que concuerda con todos los valores de los atributos de la segunda relación (divisor). Se puede definir en función de la proyección, del producto cartesiano y de la diferencia.

Agrupación (GROUP BY): Agrupa tuplas por valores comunes de ciertos atributos y aplica una función de agregación (frecuencia, suma, media...) a cada subconjunto. La figura 2.5 muestra un ejemplo.

PERS

DNI	NOMBRE	SEXO	EDAD	EC
[illegible]	JUAN	V	[illegible]	S
[illegible]	PEDRO	V	[illegible]	S
[illegible]	ELENA	M	[illegible]	C
[illegible]	MARIA	M	[illegible]	C
[illegible]	LUISA	M	[illegible]	V
[illegible]	ANA	M	[illegible]	V
[illegible]	LUIS	V	[illegible]	C
[illegible]	JUAN	V	[illegible]	C

RES = PERS AGRUPACION POR SEXO Y EC, MEDIA (EDAD)

RES

SEXO	EC	MEDIA (EDAD)
V	S	[illegible]
V	C	[illegible]
M	C	[illegible]
M	V	[illegible]

Figura 2.5. Ejemplo de Agrupación

Cómo se estructuran los problemas

Los problemas de este capítulo se estructuran en dos grandes bloques: por un lado, los problemas cuyo objetivo es diseñar un esquema relacional a partir de una serie de supuestos semánticos y, por otro lado, los problemas cuyo objetivo es obtener un esquema relacional a partir de un esquema E/R (para los que se utilizarán las reglas de transformación vistas en el apartado *Reglas de Transformación de un Esquema E/R a un esquema relacional*).

Para el primer tipo de problemas (obtención de un esquema relacional a partir de enunciado) se seguirán los siguientes pasos:

. Identificación de las relaciones básicas en los supuestos semánticos.

. Analizar cada supuesto semántico identificando para las relaciones existentes, aves primarias de estas relaciones, las claves ajenas y sus correspondientes nes de borrado y modificación, así como las restricciones de obligatoriedad y ad.

. Construcción del grafo relacional completo.

. Realización del script[21] en el estándar SQL-92.

. Realización de un conjunto de consultas tanto en SQL-92 como en álgebra onal.

ara el segundo tipo de problemas se aplicarán las reglas de transformación de mas E/R a esquemas relacionales explicadas anteriormente y que se enumeran a uación junto con las restricciones semánticas que hay que contemplar en un ma relacional:

. Transformación de entidades, atributos y dominios.

. Transformación de interrelaciones N:M.

. Transformación de interrelaciones 1:N y de Dependencias en Existencia y en ificación.

. Transformación de Generalizaciones.

. Transformación de Interrelaciones de grado superior.

. Transformación de Interrelaciones exclusivas.

. Incorporar las opciones de borrado y modificación en las claves ajenas.

. Incorporar las restricciones de obligatoriedad y unicidad.

. Incorporar las restricciones de usuario que no podían recogerse en el esquema que ahora es necesario incluir en el esquema relacional (por ejemplo, mediante cciones de verificación).

[21] Código en SQL-92 que contiene las sentencias de creación de la BD (Lenguaje de Definición de .

2.1. MODELADO RELACIONAL

PROBLEMA 1: ALOJAMIENTOS RURALES

Enunciado

La Comunidad de Madrid desea guardar información sobre los alojamientos rurales que existen en dicha comunidad. Para ello decide crear una base de datos que recoja las siguientes consideraciones:

Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc..), tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del alojamiento.

En cada alojamiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el NIF. Aunque en un alojamiento trabajen varias personas, una persona sólo puede trabajar en un alojamiento.

os alojamientos se alquilan por habitaciones y se desea conocer cuántas iones componen el alojamiento, de qué tipo (individuales, dobles, tiples) es a de estas habitaciones, si poseen cuarto de baño y el precio.

algunos de estos alojamientos se realizan actividades multiaventura adas para huéspedes (senderismo, bicicleta de montaña, etc..). Estas ades se identifican por un código. Es de interés saber el nombre de la actividad, ripción y el nivel de dificultad de dicha actividad (1_10).

tas actividades se realizan un día a la semana, por ejemplo: en la casa "Villa " se practica senderismo los jueves y se desea guardar esta información. Pero haber algún día en el que no se practique ninguna actividad.

pide:

Diseñar el esquema relacional. Indicar en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas, especificar los modos de borrado y modificación. Marcar con un asterisco (*) los campos que acepten valores nulos. Y por último, indicar si existe algún atributo que exija definir alguna restricción a fin de mantener la integridad de la base de datos.

Escribir las siguientes consultas en SQL, y en álgebra relacional:

- Nombre y descripción de las actividades que se realizan en el alojamiento denominado "La Huerta".
- Nombre de los alojamientos que tienen habitaciones dobles y realizan actividades de senderismo (el nombre de la actividad es senderismo).

sión del enunciado

CIFICACIÓN DE LAS RELACIONES BÁSICAS

"Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc.), tiene una dirección, un teléfono y una persona de contacto".

ebemos construir una primera relación ALOJAMIENTO donde se almacenen os alojamientos rurales de la Comunidad de Madrid. Su clave primaria será el del alojamiento ("Nombre_Aloja"). Consideramos que el atributo persona de o ("Contacto") puede admitir valores nulos para permitir la posibilidad de que se dé de alta un nuevo alojamiento, la persona de contacto no haya sido da todavía.

LOJAMIENTO (Nombre_Aloja, Dirección, Teléfono, Contacto*)

"En cada alojamiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el NIF".

Consideramos una nueva relación PERSONAL cuya clave primaria es el código de personal ("Código_P") y cuya clave alternativa es el NIF.

PERSONAL (Código_P, Nombre_P, Dirección, **NIF**)

"Los alojamientos se alquilan por habitaciones y se desea conocer cuantas habitaciones componen el alojamiento, de qué tipo (individuales, dobles, tiples) es cada una de estas habitaciones, si poseen cuarto de baño y el precio".

Debemos considerar una relación HABITACIÓN. Podemos tomar como clave primaria el número de habitación ("N_Habita") junto con el nombre del alojamiento. Como puede darse el caso de que no se facilite información acerca de si una determinada habitación tiene baño o no y cuál es su precio, podemos considerar que los atributos "Baño" y "Precio" admiten nulos.

Se desea recoger el número de habitaciones de que dispone cada alojamiento, por lo que debemos introducir un nuevo atributo "Número Habita" en la relación ALOJAMIENTO, como se muestra en la figura.

HABITACIÓN (Nombre_Aloja, N_Habita, Tipo, Baño*, Precio*)

ALOJAMIENTO (Nombre_Aloja, Dirección, Teléfono, Contacto*, Número_Habita)

"En algunos de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc..). Estas actividades se identifican por un código. Es de interés saber el nombre de la actividad, la descripción y el nivel de dificultad de dicha actividad (1_10)".

De este último párrafo se deduce la necesidad de una nueva relación ACTIVIDAD, cuya clave primaria será el código de actividad ("Código_Activ").

ACTIVIDAD (Código_Activ, Nombre_Activ, Descripción, Nivel)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc.), ... y una persona de contacto que pertenece al personal del alojamiento.

En cada alojamiento trabajan una serie de personas, sin embargo, aunque en un niento trabajen varias personas, una persona sólo puede trabajar en un niento".

En un alojamiento pueden trabajar una serie de personas, pero una misma persona puede trabajar en un alojamiento, por tanto debemos introducir un atributo mbre_Aloja" en la relación PERSONAL, que será clave ajena de esta relación y enciará a ALOJAMIENTO.

Además todo alojamiento tiene una persona de contacto que pertenece al personal lojamiento, luego el atributo "Contacto" será clave ajena que referenciará a SONAL.

ALOJAMIENTO (Nombre_Aloja, Dirección, Teléfono, *Contacto* *, Número_Habita)

B:N,M:C

B:C,M:C

PERSONAL (Código_P, Nombre_P, Dirección, **NIF**, *Nombre_Aloja*)

Figura 2.1.1

Entre las relaciones ALOJAMIENTO y PERSONAL, existe un ciclo referencial mpide la creación de las mismas según las especificaciones del SQL_92. Este ema se puede solucionar creando las tablas sin las claves ajenas y posteriormente arlas, como se muestra en el script.

ones de borrado y modificación

Si se elimina un alojamiento de la base de datos, se elimina todo el personal que ja en él, en cascada (B:C). Lo mismo ocurre con las modificaciones (M:C).

Sin embargo, si se elimina una persona que resulta ser una persona de contacto de lojamiento, no se debe permitir que se elimine el alojamiento, pero como el uto "Contacto" admite valores nulos, podemos considerar el borrado con puesta a s (B:N). La modificación puede hacerse en cascada, pues si se modifica una ona de contacto en PERSONAL, se debe modificar en ALOJAMIENTO (M:C).

as ECA

Se deberían controlar mediante un disparador que en un alojamiento trabaje al os una persona.

Es necesario controlar que la persona de contacto sea una de las que trabaje en el miento.

"Los alojamientos se alquilan por habitaciones y se desea conocer cuántas habitaciones componen el alojamiento...".

Como los alojamientos disponen de una serie de habitaciones y las habitaciones dependen de los alojamientos en el sentido de que para hacer referencia a una habitación necesitamos el nombre del alojamiento donde está incluida, "Nombre_Aloja" es clave ajena de la relación HABITACIÓN que referencia a ALOJAMIENTO.

ALOJAMIENTO (Nombre_Aloja, Dirección, Teléfono, *Contacto* *, Número_Habita)

B:C,M:C

HABITACIÓN (*Nombre_Aloja*, N_Habita, Tipo, Baño*, Precio*)

Figura 2.1.2

Opciones de borrado y modificación

Si se elimina o se modifica un alojamiento se deben eliminar o modificar todas las habitaciones de dicho alojamiento, por lo que tanto el borrado como la modificación se deben hacer en cascada (B:C,M:C)

Reglas ECA

Se debería controlar mediante un disparador la cardinalidad mínima en ALOJAMIENTO.

Además, como se desea conocer el número de habitaciones de las que consta cada alojamiento, deberíamos tener un disparador que, cada vez que se dé de alta o se elimine una habitación, se incremente o decremente en uno el valor del atributo "Número_Habita".

"En algunos de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc..)....

Estas actividades se realizan un día a la semana, por ejemplo: en la casa "Villa Aurora" se practica senderismo los jueves y se desea guardar esta información. Pero puede haber algún día en el que no se practique ninguna actividad".

Cada alojamiento puede proporcionar ciertas actividades multiaventura, además una misma actividad puede ser propuesta por varios alojamientos, por tanto tenemos que crear una nueva relación REALIZA_ACTIVIDAD, cuya clave primaria estará compuesta por las claves primarias de las relaciones ALOJAMIENTO y ACTIVIDAD. Cada una de ellas será a su vez clave ajena que referenciará a la relación de la cual provienen.

Debemos introducir el atributo "Día_Semana" en la relación REALIZA_ACTI-AD, pues en el enunciado se nos dice que se desea guardar la información relativa de la semana en que se realiza cada actividad. Además podemos considerar que tributo admite valores nulos, para contemplar la posibilidad de que a una cierta dad propuesta por un determinado alojamiento todavía no se le haya asignado n día.

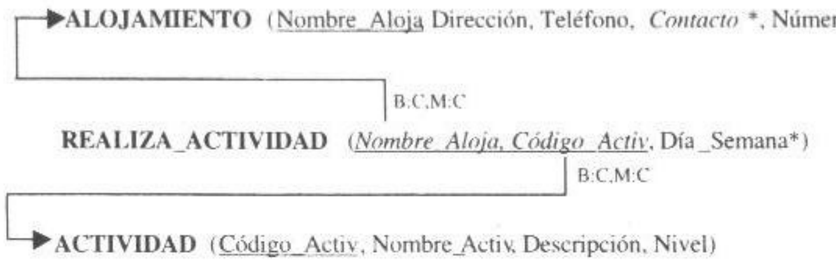


Figura 2.1.3

ones de borrado y modificación

Cada vez que se elimine o se modifique una actividad se debe eliminar o ficar la tupla correspondiente de la relación intermedia, por tanto el borrado y la ficación serán en cascada (B:C,M:C)

as ECA

Como una misma actividad es realizada por al menos un alojamiento, deberíamos un disparador para controlar esta cardinalidad mínima.

AFO RELACIONAL

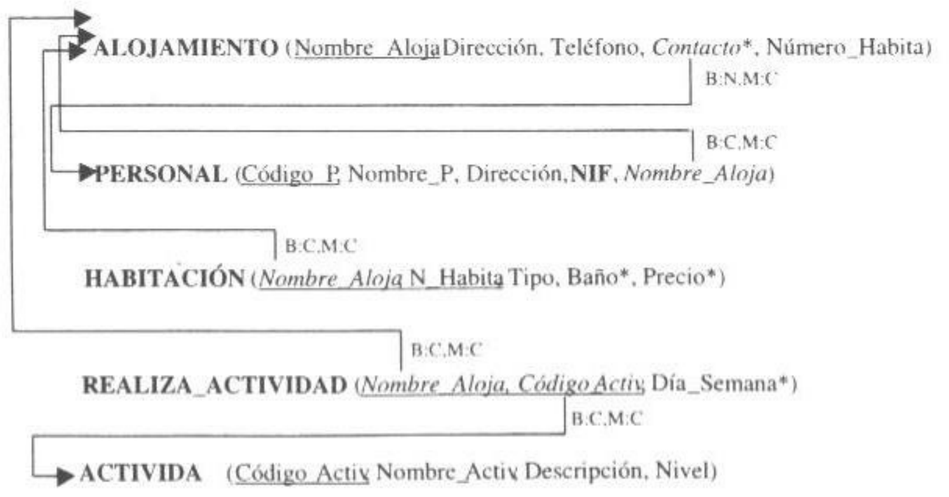


Figura 2.1.4

SCRIPT EN SQL_92

```
CREATE DOMAIN Nombre_Válido CHAR(30);
CREATE DOMAIN Tipo_Código CHAR(3);
CREATE DOMAIN Tipo_Baño CHAR(2)
CHECK (VALUE IN ("SI", "NO"));

CREATE TABLE Alojamiento (
      Nombre_Aloja Nombre_Válido,
      Dirección CHAR(25) NOT NULL,
      Teléfono CHAR (9) NOT NULL,
      Contacto Tipo_Código,
      Número_Habita INTEGER NOT NULL,
PRIMARY KEY (Nombre_Aloja)
);

CREATE TABLE Personal(
      Código_P Tipo_Código,
      Nombre_P Nombre_Válido NOT NULL,
      Dirección CHAR(25) NOT NULL,
      NIF CHAR(10) NOT NULL,
      Nombre_Aloja Nombre_Válido NOT NULL,
PRIMARY KEY (Código_P),
UNIQUE (NIF),
FOREIGN KEY (Nombre_Aloja) REFERENCES  Alojamiento
      ON UPDATE CASCADE
      ON DELETE  CASCADE
);

ALTER TABLE Alojamiento
      ADD FOREIGN KEY (Contacto) REFERENCES Personal
       ON DELETE SET NULL
       ON UPDATE CASCADE;

CREATE TABLE Habitación (
      Nombre_Aloja Nombre_Válido,
      N_Habita INTEGER(3),
      Tipo CHAR(1) NOT NULL,
      Baño Tipo_Baño,
      Precio INTEGER,
PRIMARY KEY (Nombre_Aloja, N_Habita),
FOREIGN KEY (Nombre_Aloja) REFERENCES  Alojamiento
      ON UPDATE CASCADE
      ON DELETE  CASCADE
);
```

```
TE TABLE Actividad (
Código_Acitv Tipo_Código,
Nombre_Activ Nombre_Válido NOT NULL,
Descripción CHAR(50) NOT NULL,
Nivel CHAR(2) NOT NULL,
ARY KEY (Código_Acitv),
K (Nivel BETWEEN 1 AND 10)


TE TABLE Realiza_Actividad (
Código_Activ Tipo_Código,
Nombre_Aloja Nombre_Válido,
Día_Semana CHAR(10),
ARY KEY (Código_Activ, Nombre_Aloja),
GN KEY (Código_Activ) REFERENCES  Actividad
ON UPDATE CASCADE
ON DELETE  CASCADE,
GN KEY (Nombre_Aloja) REFERENCES  Alojamiento
ON UPDATE CASCADE
ON DELETE  CASCADE
```

SULTAS

nbre y descripción de las actividades que se realizan en el alojamiento ominado "La Huerta".

```
SELECT A.Nombre_Activ, A.Descripción
FROM Actividad A, Realiza_Actividad B
WHERE Nombre_Aloja = "La Huerta" AND
      A.Código_Activ = B.Código_Activ;
```

ebra relacional

$$\Pi_{Nombre,Descripción}((ACTIVIDAD)*\Pi_{Código_Activ}(\sigma_{Nombre="La\ Huerta"}$$

$$(REALIZA_ACTIVIDAD))$$

nbre de los alojamientos que tienen habitaciones dobles y realizan actividades enderismo.

```
ELECT  Distinct B.Nombre_Aloja
ROM Actividad A, Realiza_Actividad B, Habitacion C
```



```
    WHERE A.Nombre_Activ = "Senderismo" AND C.Tipo = "Doble"
AND
          (B.Codigo_Activ = A. Codigo_Activ AND
           B.Nombre_Aloja = C.Nombre_Aloja);
```

Álgebra relacional

$$(\Pi_{Nombre_Aloja}(\sigma_{Tipo="Doble"}(HABITACION))*(\Pi_{Nombre_Aloja}$$
$$((REALIZA_ACTIVIDAD)*\Pi_{Codigo_Activ}(\sigma_{Nombre="Senderismo"}(ACTIVIDAD)))$$

PROBLEMA 2: AVES

Enunciado

Se quiere diseñar una base de datos que contenga información sobre las observaciones realizadas a distintas especies de aves en la Península Ibérica. Para ello se han de considerar las siguientes especificaciones:

Cada especie de ave se identifica por su nombre científico. Se desea conocer su nombre vulgar y una breve descripción de las características más importantes que la diferencian del grupo ornitológico al que pertenece.

Los grupos ornitológicos clasifican las especies según sus similitudes estructurales, por ejemplo, el grupo de las zancudas se caracteriza por tener patas largas y pico largo. Por lo tanto guardaremos información acerca de las patas, dedos y pico.

Cada especie de ave esta siempre asignada a un sólo grupo ornitológico.

Existen asociaciones ornitológicas de las cuales se quiere saber su nombre, la dirección y el teléfono.

Cada una de estas asociaciones consta de un grupo de personas, los observadores, que son los que realizan los avistamientos de las distintas especies. De los observadores se requiere su código de observador, el DNI, el nombre completo y la dirección. Además un observador sólo puede estar adscrito a un asociación.

Las zonas de observación tienen un código de zona que es único. También se quiere saber el nombre de la misma, la comunidad autónoma a la que pertenece, la provincia y el tipo (si es una laguna, un río, una zona costera, etc.). En cada zona se encuentra por lo menos un observador, y si existe más de uno en la misma zona pueden ser de distintas asociaciones ornitológicas. Cada observador está asignado a una determinada zona.

También se desea guardar información acerca de la fecha en la que el observador a una determinada especie, teniendo en cuenta que éstos realizan trabajo de po cada tres días y que además sólo anotan el primer avistamiento de una rminada especie por día. Esta información es muy importante para controlar las cies de aves que existen en la Península y, por tanto, tendremos que guardar las rvaciones aunque los observadores ya no se encuentren activos.

- Diseñar el esquema relacional. Indicar en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas especificar los modos de borrado y modificación. Marcar con un asterisco (*) los campos que acepten valores nulos.
- Escribir las siguientes consultas en SQL y en álgebra relacional:
  - Nombre científico y vulgar de las especies observadas por los observadores pertenecientes a la asociación ornitológica "El Petirrojo"
  - Se desea conocer el nombre y la zona a la que están asignados aquellos observadores activos que no han realizado ninguna observación.

cusión del enunciado

NTIFICACIÓN DE LAS RELACIONES BÁSICAS

"Cada especie de ave se identifica por su nombre científico. Se desea conocer su nombre vulgar y una breve descripción de las características más importantes que la diferencian del grupo ornitológico al que pertenece".

De este párrafo se deduce que podemos establecer una relación ESPECIE, cuya e primaria será "Nombre_Científico". Podemos tomar "Nombre_Vulgar" como e alternativa, pues podemos suponer que este nombre es único.

ESPECIE (Nombre_Científico, **Nombre_Vulgar**, Descripción)

"Los grupos ornitológicos clasifican las especies según sus similitudes estructurales, por ejemplo, el grupo de las zancudas se caracteriza por tener patas largas y pico largo. Por lo tanto guardaremos información acerca de las patas, dedos y pico".

Al tener los grupos ornitológicos características que se desean almacenar en la de datos, debemos introducir una nueva relación GRUPO_ORNITOLÓGICO. clave primaria consideraremos que es un código de grupo ("Código_G"). Los utos "Patas", "Dedos" y "Pico" pueden admitir valores nulos para recoger que nas de estas características no son comunes a todas las aves del grupo. oduciremos además un atributo "Nombre", que contendrá los nombres de los ntos grupos ornitológicos y que puede ser, por tanto, clave alternativa.

GRUPO_ORNITOLÓGICO (Código_G, **Nombre**, Patas*, Dedos*, Pico*)

"Existen asociaciones ornitológicas de las cuales se quiere saber su nombre, la dirección y el teléfono".

Tendremos una relación ASOCIACIÓN, donde podremos considerar como clave primaria el nombre de la asociación.

ASOCIACIÓN (Nombre_Aso, Dirección, Teléfono)

"Cada una de estas asociaciones consta de un grupo de personas, los observadores, que son los que realizan los avistamientos de las distintas especies. De los observadores se requiere su código de observador, el DNI, el nombre completo y la dirección".

Debemos almacenar los observadores en una nueva relación OBSERVADOR, en la cual el código del observador ("Código_Obs") será la clave primaria y el "DNI" la clave alternativa

OBSERVADOR (Código_Obs, **DNI**, Nombre, Dirección)

"Las zonas de observación tienen un código de zona que es único. También se quiere saber el nombre de la misma, la comunidad autónoma a la que pertenece, la provincia y el tipo (si es una laguna, un río, una zona costera, etc..)".

De la lectura de este párrafo se deduce una nueva relación ZONA_OBSERVACIÓN, cuya clave primaria será el código de zona ("Código_Z").

ZONA_OBSERVACIÓN (Código_Z, Nombre, Comunidad, Provincia, Tipo)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"Cada especie de ave está siempre asignada a un solo grupo ornitológico".

Aunque cada especie de ave esté asignada a un único grupo ornitológico, en cada grupo pueden estar incluidas varias especies de ave, por lo tanto el grupo ornitológico ("Código_G") será clave ajena de la relación ESPECIE, que referencia a GRUPO_ORNITOLÓGICO, como muestra la figura.

ESPECIE (Nombre_Científico, **Nombre_Vulgar**, Descripción, *Código_G*)

B:C,M:C

GRUPO_ORNITOLÓGICO (Código_G, **Nombre**, Patas*, Dedos*, Pico*)

Figura 2.1.5

ones de borrado y modificación

El borrado y la modificación de la clave ajena "Código_G" puede hacerse en da (B:C,M:C), pues si desaparece un grupo ornitológico desaparecerán todas las ies que pertenezcan a él.

as ECA

Si se exigiera que en nuestra base de datos todo grupo ornitológico tuviera al s una especie de ave, esta cardinalidad la deberíamos controlar mediante un rador.

"Cada una de estas asociaciones consta de un grupo de personas, los observadores, que son los que realizan los avistamientos de las distintas especies... Además un observador sólo puede estar adscrito a un asociación".

ASOCIACIÓN (Nombre_Aso, Dirección, Teléfono)

B:N,M:C

OBSERVADOR (Código_Obs **DNI**, Nombre, Dirección, *Nombre_Aso*)

Figura 2.1.6

Como se muestra en la figura, "Nombre_Aso" es clave ajena de la relación ERVADOR, que referencia a ASOCIACIÓN. Esta clave ajena puede admitir para recoger los casos de los observadores que no tienen asignada ninguna ación en un momento dado.

ones de borrado y modificación

La eliminación de una asociación ornitológica no debe implicar el borrado en da de los observadores pertenecientes a ella, ya que, como se nos dice más nte, estamos interesados en conservar las observaciones aunque desaparezcan los vadores. Por tanto al admitir la clave ajena nulos podemos optar por un borrado uesta a nulos (B:N). La modificación puede hacerse en cascada.

"En cada zona se encuentra por lo menos un observador y si existe más de uno en la misma zona pueden ser de distintas asociaciones ornitológicas. Cada observador está asignado a una determinada zona".

Como cada observador está asignado a una zona, el código de zona será clave de la relación OBSERVADOR, que referencia a ZONA_OBSERVACIÓN.

OBSERVADOR (Código_Obs **DNI**, Nombre, Dirección, *Nombre_Aso*, *Código_Z*)

B:R,M:C

ZONA_OBSERVACIÓN (Código_Z, Nombre, Comunidad, Provincia, Tipo)

Figura 2.1.7

El supuesto del enunciado que comenta que si existe más de un observador en la misma zona, pueden ser de distintas asociaciones ornitológicas, no es necesario contemplarlo, pues pueden ser de distintas zonas o de la misma.

Opciones de borrado y modificación

La eliminación debe ser restringida para evitar que se borren los observadores que pertenecen a una zona de observación eliminada (B:R). La modificación se puede hacer en cascada (M:C).

Reglas ECA

Como cada zona de observación tiene asignada al menos un observador, deberíamos controlar esta cardinalidad mínima con un disparador.

"También se desea guardar información acerca de la fecha en la que el observador divisa una determinada especie, teniendo en cuenta que estos realizan trabajo de campo cada tres días y que además sólo anotan el primer avistamiento de una determinada especie por día. Esta información es muy importante para controlar las especies de aves que existen en la Península y por tanto, tendremos que guardar las observaciones aunque los observadores ya no se encuentren activos".

Se desea guardar información acerca de la fecha en la que el observador divisa una determinada especie. Un observador puede divisar varias especies y una misma especie puede ser divisada por varios observadores, por lo tanto tendremos una nueva relación OBSERVACIÓN, cuya clave primaria será el código del observador ("Código_Obs"), el nombre científico de la especie ("Nombre_Científico") y la fecha de avistamiento ("Fecha"), pues un observador puede avistar la misma especie en diferentes fechas (los observadores sólo anotan el primer avistamiento de una especie por día).

El supuesto de que los observadores sólo realizan el trabajo de campo cada tres días no es relevante para el diseño de nuestra base de datos.

Como se desea guardar la información aunque los observadores ya no se encuentren activos, debemos introducir un nuevo atributo "Situación" en la relación

VADOR, que nos permita saber si un determinado observador está en activo o
r tanto solo podrá tomar dos valores 'Activo' o 'Inactivo'.

PECIE (Nombre_Científico, **Nombre_Vulgar** Descripción, *Código_G*)

ERVADOR (Código_Obs **DNI**, Nombre, Dirección, *Nombre_Aso**, *Código_Z*, Situación)

B:R,M:C B:R,M:C

RVACIÓN (*Código_Obs Nombre_Científico*, Fecha)

Figura 2.1.8

s de borrado y modificación

borrado y la modificación de cualquiera de las claves ajenas debe ser
do, pues en el enunciado se nos dice explícitamente que se desean recoger los
entos aunque el observador ya no esté activo, y esto implica que tampoco se
iminar especies con observaciones.

O RELACIONAL

SPECIE (Nombre_Científico, **Nombre_Vulgar** Descripción, *Código_G*)

B:C,M:C

RUPO_ORNITOLÓGICO (Código_G **Nombre**, Patas*, Dedos*, Pico*)

SOCIACIÓN (Nombre_Aso , Dirección, Teléfono)

B:N,M:C

BSERVADOR (Código_Obs **DNI**, Nombre, Dirección, *Nombre_Aso**, *Código_Z* , Situación)

B:R,M:C

ONA_OBSERVACIÓN (Código_Z, Nombre, Comunidad, Provincia, Tipo)

B:R,M:C B:R,M:C

BSERVACIÓN (*Código_Obs,Nombre_Científica* Fecha)

Figura 2.1.9

SCRIPT EN SQL_92

```
CREATE DOMAIN Nombre_Válido CHAR(60);
CREATE DOMAIN Tipo_Código  CHAR(5);
CREATE DOMAIN Tipo_Situación CHAR(8)
CHECK (VALUE IN ('Activo', 'Inactivo'));

CREATE TABLE Grupo_Ornitológico (
      Código_G Tipo_Código ,
      Patas CHAR(20),
      Dedos CHAR(20),
      Pico CHAR(20),
PRIMARY KEY (Código_G)
);

CREATE TABLE Especie (
      Nombre_Científico Nombre_Válido ,
      Nombre_Vulgar Nombre_Válido NOT NULL,
      Descripción CHAR(100) NOT NULL,
      Código_G Tipo_Código ,
PRIMARY KEY (Nombre_Científico),
UNIQUE (Nombre_Vulgar),
FOREIGN KEY (Código_G) REFERENCES  Grupo_Ornitológico
      ON UPDATE CASCADE
      ON DELETE  CASCADE
);

CREATE TABLE Asociación (
      Nombre_Aso Nombre_Válido ,
      Dirección CHAR(50) NOT NULL,
      Telefono CHAR(9) NOT NULL,
PRIMARY KEY (Nombre_Aso)
);

CREATE TABLE Zona_Observación (
      Código_Z Tipo_Código ,
      Nombre CHAR(15) NOT NULL,
      Ciudad CHAR(20) NOT NULL,
      Provincia CHAR(20) NOT NULL,
      Tipo CHAR(15) NOT NULL,
PRIMARY KEY (Código_Z)
);

CREATE TABLE Observador (
      Código_Obs Tipo_Código
      DNI CHAR(9) NOT NULL,
      Nombre CHAR(40) NOT NULL,
```

```
    Dirección CHAR(50) NOT NULL,
    Nombre_Aso Nombre_Válido ,
    Código_Z Tipo_Código  NOT NULL,
    Situación  Tipo_Situación NOT NULL,
IMARY KEY (Código_Obs),
NIQUE (DNI),
OREIGN KEY (Nombre_Aso) REFERENCES  Asociación
    ON UPDATE CASCADE
    ON DELETE  SET NULL,
OREIGN KEY (Código_Z) REFERENCES  Zona_Observación
    ON UPDATE CASCADE


REATE TABLE Observación (
    Código_Obs Tipo_Código ,
    Nombre_Científico Nombre_Válido ,
    Fecha DATE,
RIMARY KEY (Código_Obs, Nombre_Científico, Fecha),
OREIGN KEY (Código_Obs) REFERENCES  Observador
    ON UPDATE CASCADE,
OREIGN KEY (Nombre_Científico) REFERENCES  Especie
    ON UPDATE CASCADE
```

ONSULTAS

Nombre científico y vulgar de las especies observadas por los observadores pertenecientes a la asociación ornitológica "El Petirrojo".

Sql

```
SELECT DISTINCT E.Nombre_Científico, E.Nombre_Vulgar
FROM Especie E, Observación Ob, Observador O
WHERE Ob.Código_Obs =  O.Código_Obs AND Ob.Nombre_Científico =
        E. Nombre_Científico AND O.Nombre_Aso ="El Petirrojo"
```

Álgebra relacional

$$\Pi_{\text{Nombre_Científico, Nombre_Vulgar}}(\text{ESPECIE}) * \Pi_{\text{Nombre_Científico}}((\text{OBSERVACIÓN}) * (\pi_{\text{Código_Obs}}(\sigma_{\text{Nombre_Aso='El petirrojo'}}(\text{OBSERVADOR}))))$$

Se desea conocer el nombre y la zona a la que están asignados aquellos observadores activos que no han realizado ninguna observación.

Sql

```
SELECT Nombre, Código_Z
FROM Observador
WHERE Situación = "activo" AND
        Código_Obs NOT IN (SELECT DISTINCT Código_Obs
                           FROM Observación);
```

Álgebra relacional

$$\Pi_{\text{Nombre, Código_Z}}(\text{OBSERVADOR}) * (\Pi_{\text{Código_Obs}}(\sigma_{\text{Situación='activo'}}(\text{OBSERVADOR})) * \Pi_{\text{Código_Obs}}(\text{OBSERVACIÓN}))$$

PROBLEMA 3: PINACOTECAS

Enunciado

El Ministerio de Educación y Ciencia desea mantener información acerca de todos los cuadros que se encuentran en las pinacotecas españolas y toda la información relacionada con ellos.

De cada pinacoteca se desea saber el nombre (que se supone único), la ciudad en la que se encuentra, la dirección y los metros cuadrados que tiene.

Cada pinacoteca tiene una serie de cuadros de los que se quiere guardar su código, nombre, medidas, fecha en que fue pintado y técnica utilizada para pintarlo.

Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de fallecimiento). Un pintor puede tener a otro como maestro; a su vez, un maestro puede serlo de varios (o de ninguno).

Los pintores pueden pertenecer o no a una escuela de la que se desea saber su nombre y en qué país y fecha apareció.

Los pintores pueden tener también uno o varios mecenas que los protegen (nombre, país y ciudad de nacimiento, fecha de su fallecimiento, y fecha en que se inicia y termina el mecenazgo). A su vez un mismo mecenas puede serlo de varios pintores. Se desea recoger la relación que existe entre un pintor y su mecenas.

- Diseñe el esquema relacional. Indique en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas especifique los modos de borrado y modificación. Marque con un asterisco (*) los campos que acepten valores nulos.
- Efectuar en álgebra relacional y en SQL las siguientes consultas:

- Pintor que pintó el cuadro llamado "Saturno devorando a uno de sus hijos" y en qué pinacoteca se encuentra.
- Nombre y país de nacimiento de los mecenas de pintores pertenecientes a la Escuela de Florencia.

usión del enunciado

TIFICACIÓN DE LAS RELACIONES BÁSICAS

"De cada pinacoteca se desea saber el nombre (que se supone único), la ciudad en la que se encuentra, la dirección y los metros cuadrados que tiene."

De este primer párrafo se deduce una relación PINACOTECA en la que el re ("Nombre_P") es el atributo identificador principal:

PINACOTECA (Nombre_P, Ciudad_P, Dirección_P, Metros_P)

"Cada pinacoteca tiene una serie de cuadros de los que se quiere guardar su código, nombre, medidas, fecha en que fue pintado y técnica utilizada para pintarlo."

Tenemos la relación CUADRO en la cual, como no se nos dice nada, suponemos el atributo que identifica a cada cuadro es un código de cuadro ("Código_C""). nás podría ser que no se supiera la fecha exacta en la que fue pintado el cuadro. o que consideramos que el atributo "Fecha" puede admitir nulos:

CUADRO (Código_C, Nombre_C, Medidas, Fecha*, Técnica)

"Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de fallecimiento)."

Para conocer al artista que ha pintado cada cuadro debemos construir una nueva ión PINTOR, en la que, como no se nos dice nada en contra, podemos considerar lombre" como único y por tanto clave primaria de la relación.

Si consideramos que la Fecha de Nacimiento ("Fecha_N") y la Fecha de cimiento ("Fecha_F") de cada pintor pueden ser desconocidas, estos atributos ten valores nulos.

PINTOR (Nombre_Pt, País_Pt, Ciudad_Pt, Fecha_N*, Fecha_F*)

"Los pintores pueden pertenecer o no a una escuela de la que se desea saber su nombre y en qué país y fecha apareció."

Esta frase implica la creación de una nueva relación ESCUELA, cuya clave primaria es el nombre de la escuela ("Nombre_E"). Podría pensarse en considerar la clave primaria formada por Nombre y País, pues por ejemplo el Impresionismo puede ser francés o inglés, dando lugar al Impresionismo Francés y al Impresionismo Inglés, pero supondremos que en el nombre de la escuela ya almacenamos estas variantes.

ESCUELA (Nombre_E, País_E, Fecha_Aparición)

"Los pintores pueden tener también uno o varios mecenas que los protegen (nombre, país y ciudad de nacimiento, fecha de su fallecimiento, y fecha en que se inicia y termina el mecenazgo)...."

Se desea recoger todas las características de los mecenas que protegen a los artistas de nuestra base de datos, por lo que debemos construir la relación MECENAS, con clave primaria el nombre ("Nombre_M") pues, como no se nos dice nada, podemos pensar que es único. Además, y al igual que en la relación PINTOR, la fecha de comienzo ("Fecha_Ini") y la fecha de fin ("Fecha_Fin"), pueden admitir valores nulos.

MECENAS (Nombre_M, País_M, Ciudad_M, Fecha_Fallecimiento, Fecha_Ini*, Fecha_Fin*)

Tanto PINTOR como ESCUELA y MECENAS hacen referencia a países, para evitar anomalías de inserción y modificación, podría pensarse en crear una relación PAIS, con lo que se tendría un esquema más normalizado, algo parecido sucedería con las ciudades de las cuales habría de registrarse además su dependencia de los países.

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"Cada pinacoteca tiene una serie de cuadros de los que se quiere guardar su código, nombre, medidas, fecha en que fue pintado y técnica utilizada para pintarlo."

Podemos suponer que un cuadro no puede estar en varias pinacotecas a la vez, de esta forma podemos introducir "Nombre_P", como clave ajena en la relación CUADRO y evitar los grupos repetitivos que se producirían en el caso de introducir el código de cuadro como clave ajena de la relación PINACOTECA.

Figura 2.1.10

pciones de borrado y modificación

No se permite eliminar una pinacoteca mientras existan cuadros en ella (B:R). demás si modificamos (bien sea por inserción o por actualización) los datos de una inacoteca, deben poderse modificar en cascada todos los cuadros de la orrespondiente pinacoteca modificada (M:C).

eglas ECA

Deberíamos controlar mediante disparadores, que todo cuadro pertenezca siempre una pinacoteca.

"Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de fallecimiento)."

Del análisis de esta frase se deduce que un cuadro no puede ser pintado por más e un pintor, por lo que "Nombre_Pt" es clave ajena de la relación CUADRO, que eferencia a la relación PINTOR.

CUADRO (Código_C Nombre_C, Medidas, Fecha*, Técnica, *Nombre_P, Nombre_Pt*)
B:R,M:C

PINTOR (Nombre_Pt, País_Pt, Ciudad_Pt, Fecha_N*, Fecha_F*)

Figura 2.1.11

Opciones de borrado y modificación

No se debe permitir eliminar un pintor de nuestra base de datos, mientras existan uadros de ese pintor en alguna de las pinacotecas almacenadas, por tanto el borrado ebe ser restringido (B:R). Sin embargo la modificación debe realizarse en cascada M:C).

Reglas ECA

Al igual que en el caso anterior, necesitaríamos disparadores para controlar que odo pintor tenga al menos un cuadro.

Restricciones de rechazo

Se debería controlar mediante un CHECK, que la fecha de nacimiento de los intores siempre sea anterior a la fecha de fallecimiento.

"Un pintor puede tener a otro como maestro; a su vez, un maestro puede serlo de varios (o de ninguno)."

Todo maestro es un pintor y por tanto sus características (atributos) son los mismos que los del pintor. Estamos ante una relación reflexiva. Introducimos un atributo "Nombre_Maestro", cuyo contenido es el mismo que el del atributo "Nombre_Pt", como clave ajena de la relación PINTOR, que se referencia a sí misma. Además, como cada pintor puede o no tener a otro como maestro, el atributo "Nombre_Maestro" puede tomar valores nulos.

PINTOR (Nombre_Pt, País_Pt, Ciudad_Pt, Fecha_N*, Fecha_F*, *Nombre_Maestro**)
B:N,M:C

Figura 2.1.12

Opciones de borrado y modificación

Como un pintor puede ser maestro de otro pintor, si eliminamos un pintor, no debemos permitir que se borre al maestro, por lo tanto el borrado es con puesta a nulos (B:N)[21].

"Los pintores pueden pertenecer o no a una escuela de la que se desea saber su nombre y en qué país y fecha apareció."

Los pintores pueden pertenecer o no a una escuela y no se nos dice nada acerca de que un pintor pueda pertenecer a varias escuelas, por lo tanto podemos considerar "Nombre_E" como clave ajena, que admite valores nulos, de la relación pintor, que referencia a ESCUELA.

PINTOR (Nombre_Pt, País_Pt, Ciudad_Pt, Fecha_N*, Fecha_F*, Nombre_Maestro*, *Nombre_E**)
B:N,M:C

ESCUELA (Nombre_E País_E, Fecha_Aparición)

Figura 2.1.13

Opciones de borrado y modificación

No se debe permitir eliminar los pintores pertenecientes a una escuela cuando ésta se elimine de la base de datos, por tanto el borrado es restringido (B:R); en cambio la modificación de una escuela debe permitir la modificación en cascada de los pintores pertenecientes a ella (M:C).

[21] Puede elegirse esta opción, pues la clave ajena admite valores nulos.

stricciones de rechazo

Deberíamos controlar mediante una ASERCIÓN que la fecha de aparición de una uela fuese anterior a la fecha de fallecimiento de los pintores pertenecientes a ella.

> *"Los pintores pueden tener también uno o varios mecenas que los protegen (nombre, país y ciudad de nacimiento, fecha de su fallecimiento, y fecha en que se inicia y termina el mecenazgo). A su vez un mismo mecenas puede serlo de varios pintores. Se desea recoger la relación que existe entre un pintor y su mecenas."*

Como un pintor puede tener a varios como mecenas y además un mecenas puede lo de varios pintores, debemos introducir una nueva relación (relación intermedia) ECENAZGO, donde la clave primaria estará formada por las claves primarias de las aciones PINTOR y MECENAS ("Nombre_Pt", "Nombre_M"), y de esta forma itar grupos repetitivos en las dos relaciones principales. Cada uno de estos dos ributos son clave ajena de sus respectivas relaciones.

Como se desea guardar la relación existente entre cada pintor y su mecenas. bemos crear un nuevo atributo "Relación" en la relación MECENAZGO.

Además, en un principio se consideró las fechas de inicio y finalización del ecenazgo como atributos propios de la relación MECENAS, lo que significa que un ecenas sólo puede serlo en un determinado período. Pero si quisiésemos que n mecenas lo pudiese ser en distintos períodos de tiempo, deberíamos considerar Fecha_Ini", "Fecha_Fin" como atributos de la relación MECENAZGO. Es más, si uponemos que un mecenas puede serlo de un mismo pintor en distintas fechas. Fecha_Ini", "Fecha_Fin" deberían formar parte de la clave primaria.

Nuestra opción ha sido considerar que un mecenas puede serlo en distintos eríodos ("Fecha_Ini", "Fecha_Fin" atributos de la relación MECENAZGO), pero no e un mismo pintor (clave primaria formada por "Nombre_Pt" y "Nombre_M").

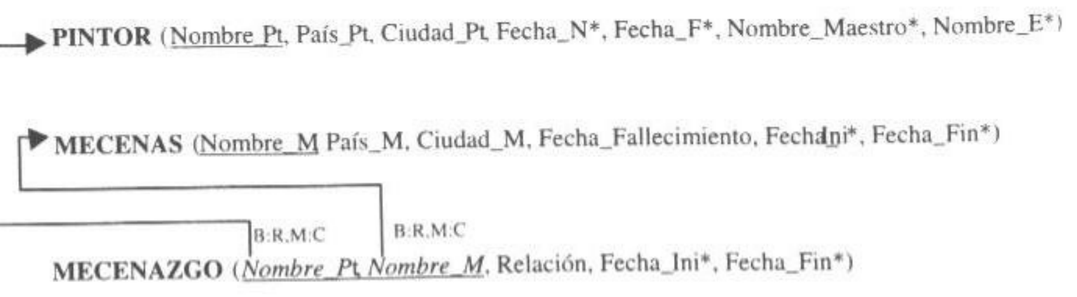


Figura 2.1.14

Opciones de borrado y modificación

Como se desea guardar la relación existente entre el pintor y su mecenas, no debemos permitir el borrado en cascada en la relación MECENAZGO, es decir, el borrado debe ser restringido (B:R). Al no decir el enunciado nada en contra, la modificación debe hacerse en cascada (M:C).

Restricciones de rechazo

Deberíamos controlar mediante una ASERCIÓN que las fechas de inicio y fin del mecenazgo sean anteriores a la fecha de fallecimiento del mecenas.

También deberíamos controlar con un CHECK que la fecha de inicio del mecenazgo es anterior a la fecha de terminación del mismo.

GRAFO RELACIONAL

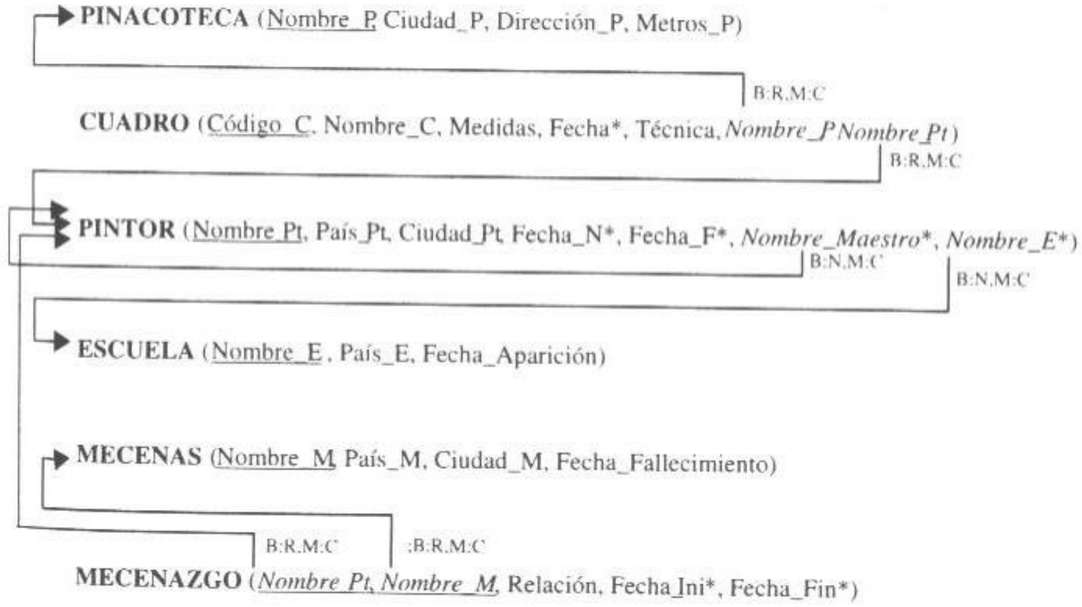


Figura 2.1. 15

SCRIPT EN SQL_92

```
CREATE DOMAIN Nombre_Válido CHAR(40);
CREATE DOMAIN Tipo_Código CHAR(5);

CREATE TABLE Pinacoteca (
    Nombre_P Nombre_Válido ,
    Ciudad_P CHAR(20) NOT NULL,
    Dirección_P CHAR(20) NOT NULL,
```

```
    Metros_P INTEGER NOT NULL,
IMARY KEY (Nombre_P)


EATE TABLE Escuela (
    Nombre_E Nombre_Válido ,
    País_E CHAR(15) NOT NULL,
    Fecha DATE NOT NULL,
IMARY KEY (Nombre_E)


EATE TABLE Mecenas (
    Nombre_M Nombre_Válido ,
    País_M CHAR(15) NOT NULL,
    Ciudad_M CHAR(20) NOT NULL,
    Fecha_Fallecimiento DATE,
IMARY KEY (Nombre_M)


EATE TABLE Pintor (
    Nombre_Pt Nombre_Válido ,
    Pais_pt CHAR(15) NOT NULL,
    Ciudad_Pt CHAR(20) NOT NULL,
    Fecha_N DATE NOT NULL,
    Fecha_F DATE NOT NULL,
    Nombre_M Nombre_Válido ,
    Nombre_E Nombre_Válido ,
IMARY KEY (Nombre_Pt),
ECK ( Fecha_N<Fecha_F),
REIGN KEY (Nombre_M) REFERENCES Pintor
    ON UPDATE CASCADE
    ON DELETE SET NULL,
REIGN KEY (Nombre_E) REFERENCES Escuela
    ON UPDATE CASCADE


EATE ASSERTION Escuela_válida
ECK (Not exists)
    ( SELECT * FROM pintor NATURAL JOIN Escuela
      WHERE Nombre_E<> NULL AND Fecha > Fecha_N));

EATE TABLE Cuadro (
    Código_C Tipo_Código,
    Nombre_C Nombre_Válido NOT NULL,
    Medidas INTEGER NOT NULL,
    Fecha DATE NOT NULL,
    Técnica CHAR(20) NOT NULL,
```



```
    Nombre_P Nombre_Válido ,
    Nombre_Pt Nombre_Válido ,
PRIMARY KEY (Código_C),
FOREIGN KEY (Nombre_P) REFERENCES Pinacoteca
    ON UPDATE CASCADE,
FOREIGN KEY (Nombre_Pt) REFERENCES Pintor
    ON UPDATE CASCADE
);

CREATE TABLE Mecenazgo (
    Nombre_M Nombre_Válido ,
    Nombre_Pt Nombre_Válido ,
    Relación CHAR(10) NOT NULL,
    Fecha_Ini DATE,
    Fecha_Fin DATE,
PRIMARY KEY (Nombre_M, Nombre_Pt),
CHECK (Fecha_Ini.< Fecha_Fin),
FOREIGN KEY (Nombre_M) REFERENCES TO Mecenas
    ON UPDATE CASCADE,
FOREIGN KEY (Nombre_Pt) REFERENCES TO Pintor
    ON UPDATE CASCADE
);

CREATE Assertion Fecha_Mecenazgo
    CHECK (NOT EXISTS
    SELECT * FROM Mecenazgo NATURAL JOIN Mecenas
    WHERE fecha_fallecimiento> Fecha_Ini));
```

CONSULTAS

1. Pintor que pintó el cuadro llamado "Saturno devorando a uno de sus hijos" y en qué pinacoteca se encuentra..

   Sql

   ```
   SELECT Nombre_Pt, Nombre_P
   FROM Cuadro
   WHERE Nombre_C = 'Saturno devorando a uno de sus hijos';
   ```

   Álgebra relacional

   $$\Pi_{\text{Nombre_Pt, Nombre_P}}(\sigma_{\text{Nombre_C = 'Saturno devorando a uno de sus hijos'}}(\text{CUADRO}))$$

2. Nombre y país de nacimiento de los mecenas de pintores pertenecientes a la Escuela de Florencia.

```
SELECT Nombre_M, País_M
FROM Mecenas
WHERE Nombre_M IN
(SELECT Nombre_M FROM Mecenazgo
 WHERE Nombre_Pt IN (SELECT Nombre_Pt
                     FROM Pintor
                     WHERE Nombre_E[22] = 'Florencia');
```

gebra relacional:

$$(\Pi_{Nombre_M,\ País_M}(\sigma_{Nombre_E\ =\ 'Florencia'}\ (MECENAS))\ *\ \Pi_{Nombre_M}((MECENAZGO)\ *\Pi_{Nombre_Pt}\ (\sigma_{Nombre_E\ =\ 'Florencia'}(PINTOR))))$$

ROBLEMA 4: CASOS DE CORRUPCIÓN

nunciado

La asociación internacional "Voto Blanco", tras una larga labor de investigación, logrado obtener valiosa información acerca de los numerosos casos de corrupción e se están dando en un país imaginario. Para tenerla toda ordenada y accesible va a eparar una base de datos que contenga todos estos datos.

De cada caso de corrupción se desea saber su código, nombre por el que se le noce, una breve descripción y una estimación de los millones que se han desviado.

Cada caso es investigado por un juez del que se desea saber su nombre, dirección, cha de nacimiento y fecha en que comenzó a ejercer. Una vez concluida la vestigación del caso emiten un dictamen que se registrará.

En cada caso hay una serie de ciudadanos implicados, cada uno de ellos con un rgo principal determinado en el momento en que se produjo el caso. De cada uno de stos se desea conocer su DNI, nombre, dirección y patrimonio.

Estos ciudadanos pueden o no pertenecer a un partido político determinado y, a eces, desempeñan un puesto en él. De cada partido quiere conocerse su nombre, irección de la sede central y teléfonos.

Cada caso de corrupción es descubierto por un periódico (nombre, dirección y rada) en una fecha determinada, que se desea saber. Cada periódico puede tener o no na afinidad con un partido político (por otro lado un partido político puede tener finidad con varios periódicos o con ninguno).

[22] Nótese que hemos supuesto "Florencia" como nombre de escuela

- Diseñe el esquema relacional. Indique en él las claves primarias, claves alternativas y las claves ajenas indicando las opciones de borrado y modificación. Marque con un asterisco (*) los campos que acepten valores nulos.
- Efectuar en álgebra relacional y en SQL las siguientes consultas:

  Implicados (y partidos en que militan) en el caso "XXX".

  Casos descubiertos por los periódicos afines al partido "YYY".

  Jueces que investigan casos en los que hay implicados miembros del partido "YYY".

Discusión del enunciado

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"De cada caso de corrupción se desea saber su código, nombre por el que se le conoce, una breve descripción y una estimación de los millones que se han desviado."*

Para recoger la información acerca de los distintos casos de corrupción, se creará la relación CASO que tendrá los atributos "Nombre" (nombre del caso de corrupción), "Descripción" (descripción del caso) y "Millones" (estimación de millones desviados).

La clave primaria de esta relación es el atributo ("Cód"). El atributo "Nombre" puede tomar valores nulos en el caso de que aún no se le conozca públicamente por ninguno, también podría tomar valores nulos el atributo "Millones" en el caso de que la estimación no se haya podido realizar, por último, supondremos que el atributo "Descripción" no tomará valores nulos. Tampoco podrá tomar valores nulos el atributo "Cód" dado que constituye la clave primaria de la relación.

CASO (Cód, Nombre*, Descripción, Millones*)

> *"Cada caso es investigado por un juez del que se desea saber su nombre, dirección, fecha de nacimiento y fecha en que comenzó a ejercer. Una vez concluida la investigación del caso emiten un dictamen que se registrará."*

Se precisa una relación JUEZ que recoja la información de los jueces encargados de investigar los casos de corrupción, los atributos de esta relación serán "Nombre", "Dirección", "Fecha_Nac" y "Fecha_Ejer". Si consideramos que la dirección, fecha de nacimiento y fecha de ejercicio de un juez pueden ser desconocidos, los correspondientes atributos podrán tomar valores nulos.

Estamos considerando que el atributo "Nombre" contiene no sólo el nombre sino también los apellidos del juez, y suponemos que no hay jueces con exactamente el

no nombre y apellidos, por lo que la clave primaria podrá ser este atributo nbre" que bajo la anterior suposición identificará unívocamente a cada uno de los es. Si no estableciéramos esta hipótesis, deberíamos proporcionar un medio de tificación alternativo para cada uno de los jueces, asignándoles un código, ya sea rado arbitrariamente o fijado por alguna organización (por ejemplo, con su DNI o go de colegiación en el colegio de abogados), o bien, por último, mediante una binación de campos cuya coincidencia para diferentes jueces sea prácticamente osible (por ejemplo, con el nombre y la fecha en que comenzó a ejercer).

JUEZ (Nombre, Dirección*, Fecha_Nac*, Fecha_Ejer*)

Los dictámenes que emite cada juez no podrían recogerse en esta relación, pues necesitaría un atributo que admitiera varios valores, lo que en el modelo relacional es posible. Más adelante daremos solución a este problema.

"En cada caso hay una serie de ciudadanos implicados, cada uno de ellos con un cargo principal determinado en el momento en que se produjo el caso. De cada uno de estos personajes se desea conocer su DNI, nombre, dirección y patrimonio."

Para recoger la información de los ciudadanos implicados se introduce una nueva ación CIUDADANO que contendrá los atributos "DNI", "Nombre", "Direcci ón" y trimonio", la clave primaria será "DNI". Puede suponerse que tanto la dirección no el patrimonio admiten valores nulos por ser éstos desconocidos.

CIUDADANO (DNI, Nombre, Dirección*, Patrimonio*)

"Estos ciudadanos pueden o no pertenecer a un partido político determinado y, a veces, desempeñan un puesto en él. De cada partido quiere conocerse su nombre, dirección de la sede central y teléfonos."

Se precisa la creación de una relación PARTIDO que permita recoger la formación acerca de los partidos políticos. Los atributos que incluiremos en esta ación serán "Nombre" y "Sede_Central" (dirección de la sede central que se supone ica), dado que un partido podrá tener varios números de teléfono, éstos no podrán cogerse introduciendo un atributo en esta relación, para recogerlos se precisará cluir una nueva relación que se considerará posteriormente cuando analicemos los puestos semánticos. La clave primaria de la relación PARTIDO será el atributo Nombre", supondremos además que la "Sede_Cen tral" no admite valores nulos.

PARTIDO (Nombre, Sede_Central)

"Cada caso de corrupción es descubierto por un periódico (nombre, dirección y tirada) en una fecha determinada que se desea saber. Cada periódico puede tener o no una afinidad con un partido político (por otro lado un partido político puede tener afinidad con varios periódicos o con ninguno)."

Para recoger la información sobre los periódicos, se requiere una nueva relación PERIÓDICO, sus atributos son "Nombre", "Dirección", y "Tirada", su clave p rimaria es el atributo "Nombre" y el atributo "Tirada" es el único que admite nulos para poder reflejar los casos en los que se desconozca su valor.

PERIÓDICO (Nombre, Dirección, Tirada*)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"Cada caso es investigado por un juez del que se desea saber su nombre, dirección, fecha de nacimiento y fecha en que comenzó a ejercer. Una vez concluida la investigación del caso emiten un dictamen que se registrará."

Es preciso relacionar los casos de corrupción con los jueces que los investigan. Para ello es suficiente informar en cada caso de corrupción del juez que lo investiga, lo que se logra incluyendo en la relación CASO un atributo "Nombre_Juez" que será una clave ajena que referencia a la relación JUEZ. Puesto que al descubrirse el caso puede que no haya sido asignado a ningún juez, este atributo podrá tomar valores nulos. Para registrar el resultado de la investigación de cada caso habrá de añadirse a la relación CASO otro atributo "Dictamen" que podrá admitir valores nulos en tanto y cuanto no haya concluido la investigación.

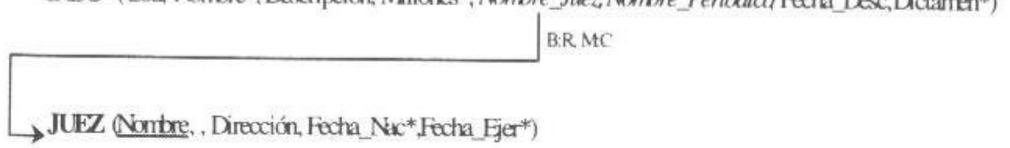


Figura 2.1.16

Opciones de borrado y modificación

Las opciones de integridad referencial podrán ser en el caso del borrado cualquiera salvo en cascada (no tiene sentido que al borrarse un juez se borren los casos de corrupción que éste investigaba) y en el caso de la modificación será en cascada.

Reglas ECA

En principio, no serían necesarias con las hipótesis consideradas, ahora bien, si por conocimiento del universo de discurso para el que se desarrolla el sistema de información pudiera afirmarse que todo caso que se registra en el mismo tiene

gnado necesariamente un juez que lo investigue (afirmación no necesariamente erta en el mundo real) deberían considerarse las reglas precisas para reflejar ese **cesariamente**, al tiempo, esta condición haría imposible otra opción de borrado que fuera el restringido.

Podrían realizarse también comprobaciones deducidas de nuestro conocimiento l discurso como, por ejemplo, una comprobación de la corrección de las fechas echa de nacimiento + 18 años <= fecha de ejercicio).

> *"En cada caso hay una serie de ciudadanos implicados, cada uno de ellos con un cargo principal determinado en el momento en que se produjo el caso. De cada uno de estos personajes se desea conocer su DNI, nombre, dirección y patrimonio."*

Los ciudadanos están implicados en casos, por conocimiento del dominio abemos que en el mismo caso pueden estar implicados diversos ciudadanos y que el ismo ciudadano puede estar implicado en múltiples casos de corrupción. Esta tuación no puede recogerse simplemente añadiendo atributos a la relación CASO ni mpoco a la relación CIUDADANO, pues ello limitaría el número de casos que se odría recoger por cada ciudadano.

Precisamos por ello una nueva relación IMPLICADO, que recoja espectivamente para cada ciudadano todos aquellos casos de corrupción en los que stá implicado y para cada caso de corrupción todos aquellos ciudadanos que están mplicados en él, en esta relación podrá recogerse también el cargo principal del iudadano en el momento en que se produjo el caso, que suponemos puede ser iferente en cada uno de los casos.

Los atributos de la relación IMPLICADO serán "Cód_Caso", "DNI_Ciudadano" "Cargo", la clave primaria estará compuesta por los dos primeros atributos que al iempo serán claves ajenas, la primera referencia a la relación CASO y la segunda eferencia a la relación CIUDADANO. El único atributo que podrá tomar valores ulos será "Cargo", pues un implicado no tiene por qué tener un determinado cargo.

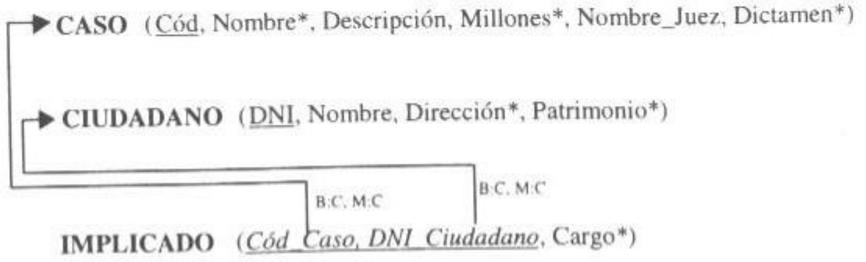


Figura 2.1.17

Opciones de borrado y modificación

Las opciones de integridad referencial de ambas claves ajenas podrán ser las de borrado, tanto en cascada como restringido dependiendo de la semántica pretendida (no podrán ser ni nulo ni por defecto pues ello podría violar las condiciones de la clave primaria en la relación IMPLICADO), por su parte la de modificación en ambos casos será en cascada.

Reglas ECA

Si se exigiera la condición de que en todo caso hay ciudadanos implicados, esta condición debería controlarse de alguna manera, sin embargo, los implicados en un caso podrían no ser inicialmente conocidos, por lo que no es preciso asegurar esta semántica. Por el contrario, cualquier ciudadano en la BD lo estará como consecuencia de haberse visto implicado en un caso, por lo que todo ciudadano[23] está implicado necesariamente en al menos un caso. Esta semántica (o al menos, parte de la misma) se puede recoger mediante disparadores dentro de la propia BD, por ejemplo, debería impedirse el borrado del último caso en que se ha visto implicado un determinado ciudadano.

> *"Estos ciudadanos pueden o no pertenecer a un partido político determinado y, a veces, desempeñan un puesto en él. De cada partido quiere conocerse su nombre, dirección de la sede central y teléfonos."*

Existe una relación de pertenencia entre los ciudadanos y los partidos políticos, para reflejar esta relación, recogeremos para cada ciudadano el partido al que pertenece (único, en caso de que pertenezca a alguno) y el puesto que ocupa en el mismo (único, en caso de que ocupe alguno), lo que supone añadir a la relación CIUDADANO dos nuevos atributos "Nombre_Partido" y "Puesto", ambos admitirán valores nulos y "Nombre_Partido" será una clave ajena que referencia a la relación PARTIDO.

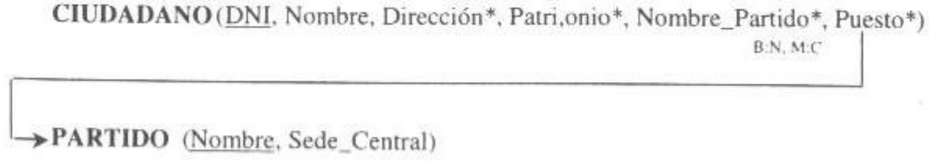


Figura 2.1.18

[23] Si bien no todo ciudadano, sí al menos todo ciudadano desde la perspectiva del dominio de interés al sistema de información que se está desarrollando.

pciones de borrado y modificación

La opción de borrado de la clave ajena "Nombre_Partido" podrá ser tanto stringida como puesta a nulos (no se consideran adecuadas para la semántica del ercicio ni la opción de cascada ni la asignación de un valor por defecto), por el ntrario, la opción de modificación será en cascada.

La relación TELÉFONO_PARTIDO recogerá todos los teléfonos de cada partido, ndrá los atributos "Nombre" (nombre del partido) y "Teléfono". Lo habitual en estos sos (relaciones introducidas para recoger los diversos valores que puede adoptar un ibuto de la misma tupla) sería que la clave primaria estuviera formada componiendo mbos atributos, sin embargo, dado que se sabe que los teléfonos no pueden repetirse, staría el uso del atributo "teléfono" como clave primaria de la relación. El atributo ombre" será una clave ajena que referencia a la relación PARTIDO.

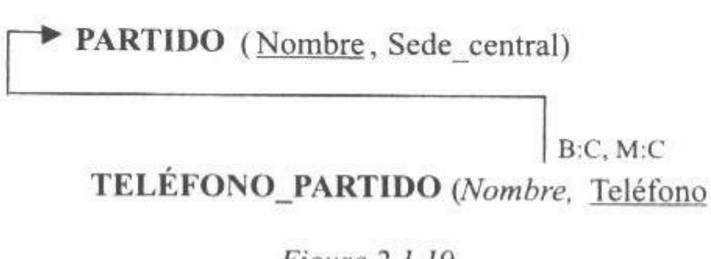


Figura 2.1.19

pciones de borrado y modificación

Tanto la opción de borrado como la de modificación de la clave ajena "nombre" rán en cascada.

eglas ECA

Si se exigiera que todo partido tuviera al menos un teléfono, deberíamos asegurar ediante disparadores que frente a cualquiera de las operaciones posibles sobre la BD. cualquier caso, por cada tupla en la relación PARTIDO, hay al menos una tupla en relación TELÉFONO_PARTIDO.

"Cada caso de corrupción es descubierto por un periódico (nombre, dirección y tirada) en una fecha determinada, que se desea saber. Cada periódico puede tener o no una afinidad con un partido político (por otro lado un partido político puede tener afinidad con varios periódicos o con ninguno)."

Para reflejar el hecho de que cada caso de corrupción es descubierto por un riódico, se incluirá en la relación CASO una referencia al periódico que lo ha scubierto y la fecha en que lo ha descubierto. Se añaden por tanto a la relación ASO los atributos "Nombre_Periódico" y "Fecha_Desc", ninguno de ellos admitirá

valores nulos, pues suponemos que todo caso de corrupción lo descubre un periódico (no se hace mención a casos no descubiertos por periódicos). El atributo "Nombre_Periódico" es una clave ajena que referencia a la relación PERIÓDICO.

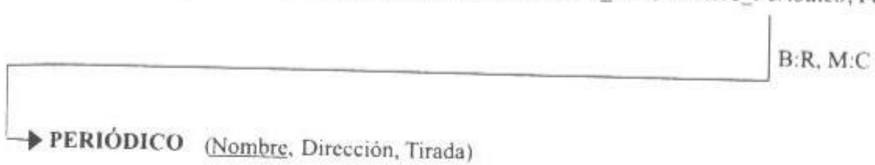


Figura 2.1.20

Opciones de borrado y modificación

La opción de borrado de la clave ajena "Nombre_Periódico" será restringida (no podría ser puesta a nulos y las opciones de cascada y puesta a un valor por defecto no se consideran convenientes para la semántica del enunciado). La opción de modificación será en cascada.

Reglas ECA

Debería asegurarse mediante los correspondientes disparadores que cada tupla en la relación CASO está relacionada con una tupla en la relación PERIÓDICO.

Para reflejar la afinidad política de los periódicos, se informará para cada periódico el partido político al que es afín. Se añade para ello a la relación PERIÓDICO el atributo "Nombre_Partido" que será optativo, pues no todos los periódicos tienen por qué ser afines a un partido político. Este atributo será por tanto una clave ajena que estará referenciando a la relación PARTIDO.

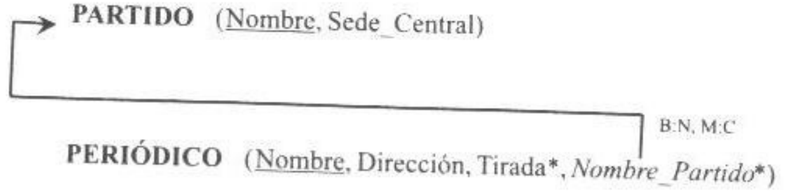


Figura 2.1.21

Opciones de borrado y modificación

Las opciones de integridad referencial de esta clave ajena serán la de borrado restringida o con puesta a nulos y la de modificación en cascada.

las ECA

No serían precisas, pues toda la semántica precisa se estaría recogiendo mente con opciones propias del modelo relacional.

AFO RELACIONAL

En el siguiente grafo relacional se refleja todo lo obtenido:

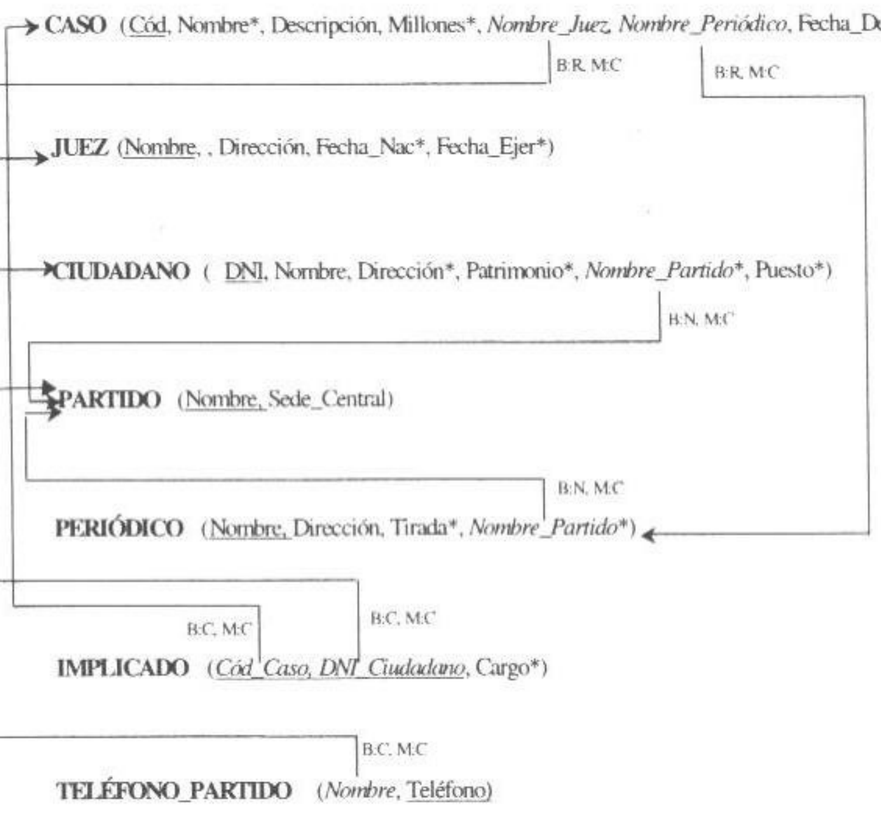


Figura 2.1.22

RIPT EN SQL_92

```
REATE DOMAIN Nombre_Válido CHAR(40);
REATE DOMAIN Dirección_Válida CHAR(40);
REATE DOMAIN Tipo_Código CHAR(5);
REATE DOMAIN Tipo_DNI CHAR(9);

REATE TABLE Juez (
    Nombre Nombre_Válido,
    Dirección Dirección_Válida,
```



```
    Fecha_Nac DATE,
    Fecha_Ejer DATE,
PRIMARY KEY (Nombre)
);

CREATE TABLE Partido (
    Nombre Nombre_Válido,
    Sede_Central Dirección_Válida UNIQUE,
PRIMARY KEY (Nombre));

CREATE TABLE Periódico (
    Nombre Nombre_Válido,
    Dirección Dirección_Válida,
    Tirada INTEGER NOT NULL,
    Nombre_Partido Nombre_Válido NOT NULL,
PRIMARY KEY (Nombre),
FOREIGN KEY (Nombre_Partido) REFERENCES Partido
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE Caso (
    Cód Tipo_Código,
    Nombre Nombre_Válido NOT NULL,
    Descripción CHAR(2000),
    Millones INTEGER NOT NULL,
    Nombre_Juez Nombre_Válido NOT NULL,
    Dictamen CHAR(2000),
    Nombre_Periódico Nombre_Válido,
    Fecha_Desc DATE,
PRIMARY KEY (Cód),
FOREIGN KEY (Nombre_Juez) REFERENCES JUEZ
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
FOREIGN KEY (Nombre_Periódico) REFERENCES Periódico
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Ciudadano (
    DNI Tipo_DNI,
    Nombre Nombre_Válido,
    Dirección Dirección_Válida NOT NULL,
    Patrimonio INTEGER NOT NULL,
    Nombre_Partido Nombre_Válido NOT NULL,
    Puesto CHAR(40) NOT NULL,
PRIMARY KEY (DNI),
```

```
OREIGN KEY (Nombre_Partido) REFERENCES Partido
      ON DELETE SET NULL
      ON UPDATE CASCADE


REATE TABLE Implicado (
      Cód_Caso Tipo_Código,
      DNI_Ciudadano Tipo_DNI,
      Cargo CHAR(40) NOT NULL,
RIMARY KEY (Cód_Caso, DNI_Ciudadano)
OREIGN KEY (Cód_Caso) REFERENCES Caso
      ON DELETE CASCADE
      ON UPDATE CASCADE,
OREIGN KEY (DNI_Ciudadano) REFERENCES Ciudadano
      ON DELETE CASCADE
      ON UPDATE CASCADE
;

REATE TABLE Teléfonos_Partido (
      Nombre Nombre_Válido,
      Teléfono CHAR(12),
RIMARY KEY (Teléfono),
OREIGN KEY (Nombre) REFERENCES Partido
      ON DELETE CASCADE
      ON UPDATE CASCADE
);
```

CONSULTAS

1. Implicados (y partidos en que militan) en el caso "XXX".

Sql

```
SELECT Ciudadano.DNI, Ciudadano.Nombre_Partido
FROM Implicado, Ciudadano
WHERE Implicado.Cód_Caso='XXX'
        AND Implicado.DNI_Ciudadano=Ciudadano.DNI;
```

Álgebra relacional

$$\Pi_{DNI,\ Nombre_Partido}\ (CIUDADANO) * \Pi_{DNI_Ciudadano}(\sigma_{Cód_Caso='XXX'}(IMPLICADO))$$

2. Casos descubiertos por los periódicos afines al partido "YYY".

Sql

```
SELECT Caso.*
FROM Caso, Periódico
```



```
WHERE Periódico.Nombre_Partido = 'YYY'
        AND Caso.Nombre_Periódico = Periódico.Nombre;
```

Álgebra relacional

$$\Pi_{Cód,\ Nombre,\ Descripción,\ Millones,\ Nombre_Juez,\ Nombre_Periódico,\ Fecha_Dese,\ Dictamen}((CASO) * \Pi_{Nombre}(\sigma_{Nombre_Partido='YYY'}(PERIÓDICO)))$$

3. Jueces que investigan casos en los que hay implicados miembros del partido "YYY".

Sql

```
SELECT DISTINCT Caso.Nombre_Juez
FROM Caso, Ciudadano, Implicado
WHERE Ciudadano.Nombre_Partido='YYY'
        AND Ciudadano.DNI=Implicado.DNI_Ciudadano
        AND Implicado.Cód_Caso=Caso.Cód);
```

Álgebra relacional

$$\Pi_{Nombre_Juez}(\Pi_{Cód,\ Nombre_Juez}(CASO) * (\Pi_{DNI,\ Cód.Caso}(IMPLICADO) * (\Pi_{DNI}(\sigma_{Nombre_Partido='YYY'}(CIUDADANO)))))$$

PROBLEMA 5: COMISARÍA

Enunciado

Se desea recoger en una base de datos información acerca del funcionamiento interno de una comisaría de policía.

Se consideran los siguientes supuestos:

En la comisaría trabajan una serie de policías (DNI, nombre, categoría), que pueden desempeñar funciones distintas: administrativos, agentes, etc.

Cada policía tiene un único jefe, aunque un policía puede ser jefe de varios.

En la comisaría existe un arsenal de armas. Cada arma está identificada por un código único, pertenece a una clase y tiene un nombre determinado.

Cada policía puede utilizar una o varias armas en un momento determinado. Es importante conocer el grado de habilidad (puntuación de 1 a 10) de cada policía con cada una de las armas que utiliza.

Un delincuente (DNI, nombre, teléfono) es arrestado por uno o varios policías.

A cada delincuente que permanezca en la comisaría se le encierra en un calabozo (código y ubicación). En el calabozo pueden estar encerrados varios delincuentes.

Los delincuentes están involucrados en casos (código de caso y juzgado que lo instruye); interesa saber cuál es el principal cargo (robo, homicidio, etc.) que se le imputa a un delincuente en cada delito en que está involucrado.

Uno o varios policías investigan cada uno de los casos.

Se pide:

- Diseñar el esquema relacional indicando claves primarias y ajenas, nulos, opciones de borrado y modificación, etc.
- Sobre este esquema relacional, elaborar las siguientes consultas en SQL y en álgebra relacional:
  - Nombres de los policías que han investigado algún caso en el que esté inculpado el delincuente "El mono".
  - Todos los datos del jefe del policía que utiliza el arma W1234.
  - Códigos de las armas que utilizan los policías que detuvieron a los delincuentes que están en el calabozo número 13.

Esquema relacional

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"En la comisaría trabajan una serie de policías (DNI, nombre, categoría), que pueden desempeñar funciones distintas: administrativos, agentes, etc.,...En la comisaría existe un arsenal de armas. Cada arma está identificada por un código único, pertenece a una clase y tiene un nombre determinado".*

Tras una primera lectura del enunciado, se detectan algunas relaciones de forma sencilla. Así, parece evidente que POLICÍA (identificado por su DNI), será una relación. Tendrá como atributos el nombre, la categoría y la función que realiza. La función que puede realizar un policía la modelamos en principio como un atributo. Aunque también se podría haber modelado como una relación si suponemos que un policía puede realizar varias funciones.

ARMA (identificada, según nos especifican, por un código) será otra relación, y tendrá como atributos, de momento, la clase y el nombre de la misma. No lo será, sin embargo, ARSENAL, ni COMISARÍA. Se podría pensar en ARSENAL y

COMISARÍA como relaciones, pero hemos de tener en cuenta que estamos modelando la base de datos de UNA COMISARÍA y ésta tiene UN ARSENAL, cuyo único fin es servir de almacén a las armas. Una posible situación en la que deberíamos almacenar los datos del arsenal sería aquella en la que las armas pudieran almacenarse en distintos arsenales. De forma análoga, si existieran distintas comisarías deberíamos plantearnos crear una relación para almacenar información referente a éstas. Sin embargo, no ocurre ni lo uno ni lo otro, y tanto la comisaría como el arsenal son únicos (o al menos no es significativo que no lo fueran).

ARMA (Cód_Arma, Clase, Nombre_Arma)

POLICÍA (DNI_Poli, Nombre_Poli, Categoría, Función)

> *"Un delincuente (DNI, nombre, teléfono) es arrestado por uno o varios policías.*
>
> *A cada delincuente que permanezca en la comisaría se le encierra en un calabozo (código y ubicación).*
>
> *Los delincuentes están involucrados en casos (código de caso y juzgado que lo instruye)".*

Podemos añadir a nuestra lista inicial de relaciones detectadas DELINCUENTE, que se identificará por su DNI y tendrá un nombre y un teléfono (el cual suponemos que podríamos no conocer, por lo que admite valores nulos), CALABOZO, que tendrá un código y una ubicación, y CASO, identificado por un código de caso, y para el que almacenaremos el juzgado que lo instruye. Podríamos habernos planteado haber diseñado un juzgado como una relación. Sin embargo, puesto que cada caso lo lleva un único juzgado, y no disponemos de ninguna otra información acerca de los mismos (como, por ejemplo, su dirección, teléfono, etc.), hemos decidido modelarlo como atributo.

CASO (Cód_Caso, Juzgado)

DELINCUENTE (DNI_Deli, Nombre, Teléfono*)

CALABOZO (Cód_Cala, Ubicación)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

A continuación pasaremos a modelar los conceptos que se refieren a "enlaces" entre las relaciones detectadas, y otros conceptos todavía no recogidos. Así, deberemos decidir cómo almacenar los policías que son jefes de otros, qué armas utilizan los policías y con qué nivel de habilidad, qué policías investigan los casos y arrestan a los delincuentes, en qué casos están implicados los delincuentes, y en qué calabozos son encerrados.

"Cada policía tiene un único jefe, aunque un policía puede ser jefe de varios".

El enunciado nos dice que cada policía tiene un único jefe. Esto lo modelaremos incluyendo en la relación POLICÍA un nuevo atributo que identifique a su jefe, que será clave ajena que referencie a la propia relación POLICÍA. Llamaremos a este atributo, "Jefe". Supondremos, además, que todo policía debe tener un jefe, por lo que no admitiremos nulos en dicho atributo.

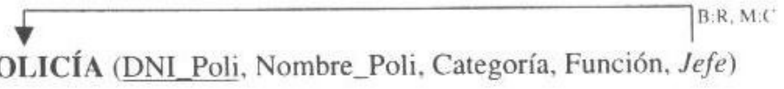


Figura 2.1.23

Opciones de borrado y modificación

La opción de modificación será en cascada, y la de borrado restringido, prohibiendo eliminar un policía que todavía tenga subordinados asignados.

"Cada policía puede utilizar una o varias armas en un momento determinado. Es importante conocer el grado de habilidad (puntuación de 1 a 10) de cada policía con cada una de las armas que utiliza".

Debido a que los policías pueden utilizar varias armas en su trabajo, no podemos almacenar esta información en la relación Policía. Si, además, cada arma pudiera ser utilizada por distintos policías, necesitaríamos una nueva relación que almacenara esta asociación. Supondremos, sin embargo, que cada arma es utilizada por un único policía. Propagaremos, por tanto, la clave de la relación Policía a la relación Arma, como clave ajena y sin admitir valores nulos (esto es, toda arma ha de estar asignada a uno y sólo un policía). En esta misma relación incluiremos un atributo que almacene el grado de habilidad que tiene con dicho arma el policía que la utiliza.

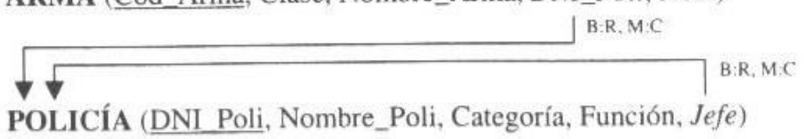


Figura 2.1.24

Opciones de borrado y modificación

No permitiremos borrar a un policía que tenga asignadas armas (Borrado restringido), y la modificación de aquél se propagará a éstas en cascada.

Reglas ECA

Se debería implementar un disparador que obligue a que, en caso de que se modifique el policía al que se le asigne un arma, se revise también la nota o nivel de habilidad del mismo.

"Uno o varios policías investigan cada uno de los casos".

En los requisitos se nos especifica que uno o varios policías investigan cada uno de los casos. Supondremos, además, que un caso puede estar siendo investigado por más de un policía. Por tanto, necesitamos una relación intermedia cuya clave primaria esté formada por la concatenación de las claves ajenas de las relaciones POLICÍA y CASO.

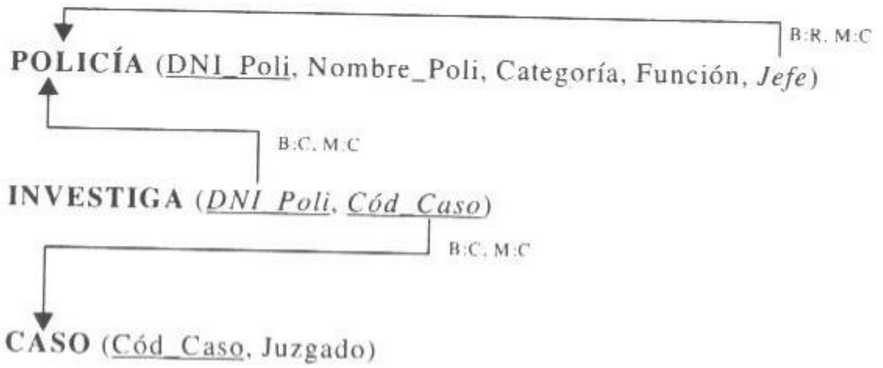


Figura 2.1.25

Opciones de borrado y modificación

Las opciones de borrado y modificación serán en cascada

Reglas ECA

El hecho de que cada caso debe estar siendo investigado como mínimo por un policía deberemos recogerlo por medio de disparadores. Como no nos indican otra cosa, supondremos que un policía puede no estar asignado a ningún caso, y por tanto no serán necesarios más disparadores.

Debemos tener cuidado a la hora de borrar un policía, ya que puede ocurrir que dejemos un caso sin policía asignado. Este hecho también se controlaría mediante un disparador que no dejaría borrar ni al policía ni a la tupla que lo relaciona con su caso en la relación INVESTIGA.

"Un delincuente (DNI, nombre, teléfono) es arrestado por uno o varios policías".

Es necesaria una nueva relación análoga a la anterior para el caso de los arrestos delincuentes que llevan a cabo los policías. El enunciado nos especifica claramente todo delincuente es arrestado por uno o varios policías, es decir, que todo ncuente es arrestado. Esto no significa, evidentemente, que estemos modelando la pica situación de un mundo sin delincuencia, sino que únicamente guardaremos rmación de los delincuentes arrestados. Sin embargo, puesto que no se nos dice a en contra, puede haber policías que no hayan arrestado a ningún delincuente. aremos, por tanto, una relación ARRESTA cuya clave primaria estará formada por oncatenación de las claves ajenas de las relaciones POLICÍA y DELINCUENTE.

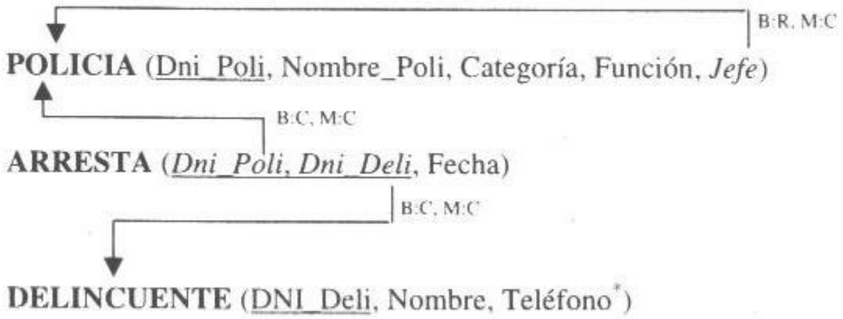


Figura 2.1.26

ciones de borrado y modificación

Las opciones de borrado y modificación serán en cascada.

glas ECA

Tendremos que controlar mediante disparadores que todo delincuente haya sido estado por, al menos, un policía (o, lo que es lo mismo, que toda ocurrencia de la ación DELINCUENTE está incluida en la relación ARRESTA).

> *"Los delincuentes están involucrados en casos (código de caso y juzgado que lo instruye); interesa saber cuál es el principal cargo (robo, homicidio, etc.) que se le imputa a un delincuente en cada delito en que está involucrado"*

Al igual que los policías investigaban casos, lo que nos llevó a la creación de la ación INVESTIGA, los delincuentes están INVOLUCRADOS en dichos casos. La ántica que consideramos es muy similar, es decir, todo delincuente sobre el que acenamos información está involucrado en al menos un caso (de hecho, como mos antes, está arrestado). En los casos, están involucrados varios o ningún incuentes (de esta manera, se permite tener información acerca de los casos entras estén siendo investigados). Crearemos por tanto la relación VOLUCRADO, cuya clave será la concatenación de las claves ajenas de CASO y

DELINCUENTE. Esta relación tendrá además otro atributo que almacenará el cargo que se le imputa a dicho delincuente en el caso.

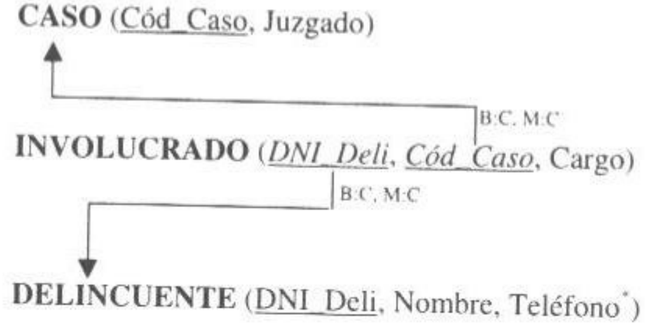


Figura 2.1.27

Opciones de borrado y modificación

Las opciones de borrado y modificación serán en cascada.

Reglas ECA

Mediante disparadores controlaremos que no pueda haber delincuentes sin estar involucrados en ningún caso.

> *"A cada delincuente que permanezca en la comisaría se le encierra en un calabozo (código y ubicación). En el calabozo pueden estar encerrados varios delincuentes".*

Por último, sólo nos faltaría por recoger en qué calabozo está encerrado cada delincuente mientras permanece en la comisaría. En principio, un calabozo podría estar vacío, pero consideraremos (de forma bastante razonable), que todo delincuente arrestado (si recordamos, todos nuestros delincuentes lo están) está encerrado en uno y sólo un calabozo. Por tanto, no será necesario crear ninguna relación auxiliar, sino que podremos usar la relación DELINCUENTE para almacenar el calabozo en que se encuentra propagando el código del calabozo a dicha relación como clave ajena.

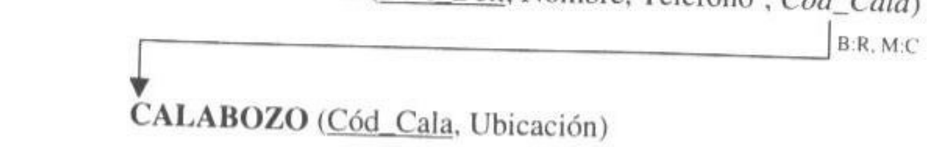


Figura 2.1.28

iones de borrado y modificación

Las modificaciones en la relación CALABOZO se propagarán en cascada, y ideraremos que no se permite eliminar un calabozo mientras haya algún acuente dentro (cosa por otra parte bastante recomendable), es decir, los borrados n restringidos.

afo relacional

El grafo relacional completo se muestra a continuación.

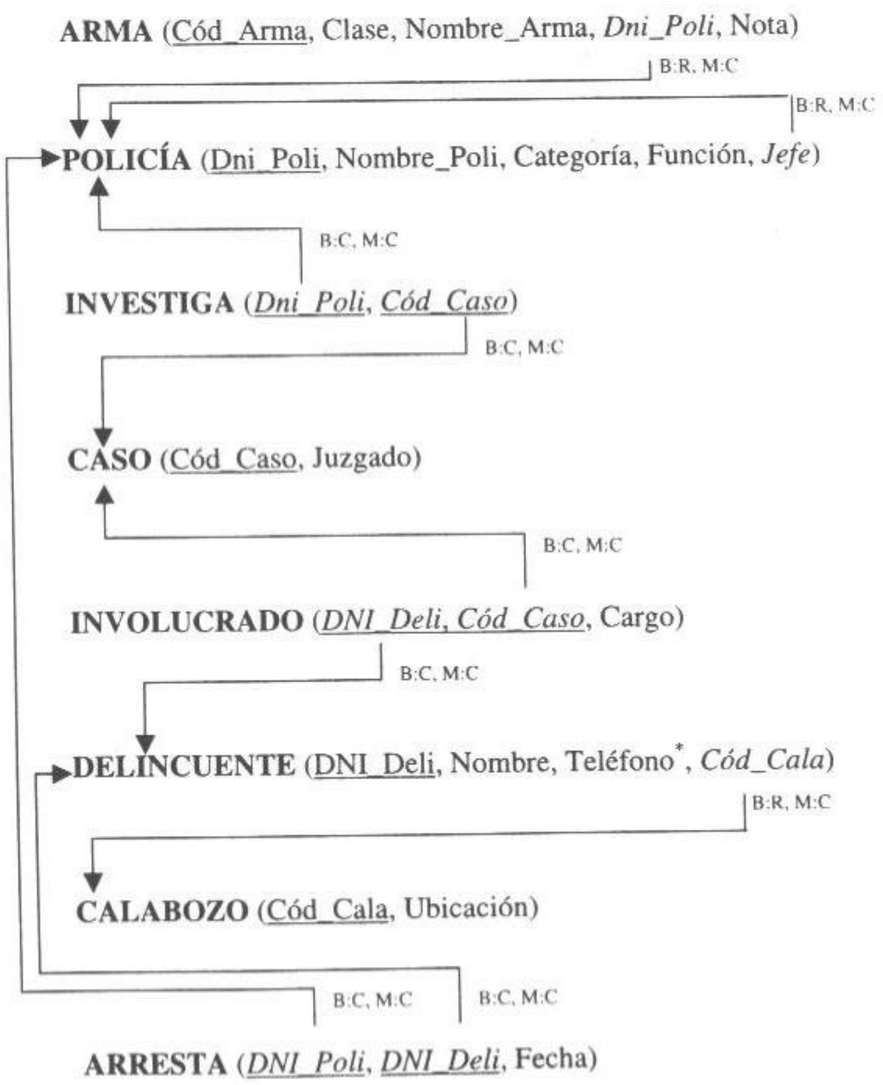


Figura 2.1.29

SCRIPT EN SQL-92

```
CREATE DOMAIN Nombre_Válido CHAR(40);
CREATE DOMAIN Tipo_Código CHAR(5);
CREATE DOMAIN Tipo_DNI  CHAR(9);

CREATE TABLE Policia (
      DNI_Poli Tipo_DNI,
      Nombre_Poli Nombre_Válido NOT NULL,
      Categoría CHAR(15) NOT NULL,
      Función CHAR(15) NOT NULL,
      Jefe Tipo_DNI NOT NULL,
PRIMARY KEY (DNI_Poli),
FOREIGN KEY (Jefe) REFERENCES Policía
      ON UPDATE CASCADE
);

CREATE TABLE Arma (
      Cód_Arma Tipo_Código
      Clase CHAR(20) NOT NULL,
      Nombre_Arma Nombre_Válido NOT NULL,
      DNI_Poli Tipo_DNI NOT NULL,
      Nota INTEGER NOT NULL,
PRIMARY KEY (Cód_Arma),
CHECK ( Nota>0 AND Nota<11)[24],
FOREIGN KEY (DNI_Poli) REFERENCES Policia
      ON UPDATE CASCADE
);

CREATE TABLE Caso (
      Cód_Caso Tipo_Código,
      Juzgado CHAR(20) NOT NULL,
PRIMARY KEY (Cód_Caso)
);

CREATE TABLE Investiga (
      DNI_Poli Tipo_DNI,
      Cód_Caso Tipo_Código,
PRIMARY KEY (DNI_Poli, Cód_Caso),
FOREIGN KEY (DNI_Poli) REFERENCES Policía
      ON UPDATE CASCADE
      ON DELETE CASCADE,
FOREIGN KEY (Cód_Caso) REFERENCES Caso
      ON UPDATE CASCADE
      ON DELETE CASCADE
```

[24] Esta restricción también podría recogerse mediante un dominio.

```
EATE TABLE Calabozo (
    Cód_Cala Tipo_Código,
    Ubicación CHAR(20) NOT NULL,
MARY KEY (Cód_Cala)

EATE TABLE Delincuente (
    DNI_Deli Tipo_DNI,
    Nombre Nombre_Válido NOT NULL,
    Teléfono CHAR(15),
    Cód_Cala Tipo_Código NOT NULL,
IMARY KEY (DNI_Deli),
REIGN KEY (Cód_Cala) REFERENCES Calabozo
    ON UPDATE CASCADE

EATE TABLE Involucrado (
    DNI_Deli Tipo_DNI,
    Cód_Caso Tipo_Código,
    Cargo CHAR(20) NOT NULL,
IMARY KEY (DNI_Deli, Cód_Caso),
REIGN KEY (DNI_Deli) REFERENCES Delincuente
    ON UPDATE CASCADE
    ON DELETE CASCADE,
REIGN KEY (Cód_Caso) REFERENCES Caso
    ON UPDATE CASCADE
    ON DELETE CASCADE);

EATE TABLE Arresta (
    DNI_Poli Tipo_DNI,
    DNI_Deli Tipo_DNI,
    Fecha DATE NOT NULL,
IMARY KEY (DNI_Poli, DNI_Deli),
REIGN KEY (DNI_Poli) REFERENCES Policía
    ON UPDATE CASCADE
    ON DELETE CASCADE,
REIGN KEY (DNI_Deli) REFERENCES Delincuente
    ON UPDATE CASCADE
    ON DELETE CASCADE);
```

ONSULTAS

1. Nombres de los policías que han investigado algún caso en el que esté inculpado el delincuente "El mono".

Sql

```
SELECT Nombre_Poli
FROM Policia, Investiga, Involucrado, Delincuente
WHERE Policia.DNI_Poli=Investiga.DNI_Poli
        AND Investiga.Cód_Caso=Involucrado.Cód_Caso
        AND Involucra.DNI_Deli=Delincuente.DNI_Deli
        AND Delincuente.Nombre='El Mono'
```

Álgebra relacional

$$\Pi_{\text{Nombre_Poli}}((\Pi_{\text{DNI_Poli, Nombre_Poli}}(\text{POLICIA})) * (\Pi_{\text{Cód_Caso, DNI_Poli}}(\text{INVESTIGA})) * (\Pi_{\text{DNI_Deli, Cód_Caso}}(\text{INVOLUCRADO})) * (\Pi_{\text{DNI_Deli}}(\sigma_{\text{Nombre='El mono'}}(\text{DELINCUENTE}))))$$

2. Todos los datos del jefe del policía que utiliza el arma W1234.

Sql

```
SELECT DNI_Poli, Nombre_Poli, Categoría, Función
FROM Policia
WHERE DNI_Poli=
        (SELECT Jefe
         FROM Policia
         WHERE DNI_Poli=
                (SELECT DNI_Poli
                 FROM Arma
                 WHERE Cód_Arma = 'W1234'))
```

Álgebra relacional

$$\Pi_{\text{DNI_Poli, Nombre_Poli, Categoría, Función}}(\text{POLICIA}) * (\Pi_{\text{DNI_Poli, Jefe}}(\text{POLICIA})) * (\Pi_{\text{DNI_Poli}}(\sigma_{\text{Cód_Arma='W1234'}}(\text{ARMA})))$$

3. Códigos de las armas que utilizan los policías que detuvieron a los delincuentes que están en el calabozo número 13.

Sql

```
SELECT Cód_Arma
FROM Arma
WHERE DNI_poli IN
        (SELECT DNI_Poli
         FROM Arresta
         WHERE DNI_Deli IN
                (SELECT DNI_Deli
                 FROM Delincuente
                 WHERE Cód_Cala = '13'))
```

gebra relacional

$\Pi_{Cód_Arma}((\Pi_{DNI_poli, Cód_Arma}(ARMA)) * (\Pi_{DNI_Poli}(\Pi_{DNI_Poli, DNI_Deli}(ARRESTA)) * (\Pi_{DNI_Deli}(\sigma_{Cód_cala='13'}(DELINCUENTE)))))$

OBLEMA 6: CAMINO DE SANTIAGO

nciado

La Consejería de Cultura de la Comunidad Gallega ha decidido guardar mación referente al Camino de Santiago en una Base de Datos. La información esea almacenar es la siguiente:

Teniendo en cuenta que la peregrinación a Santiago se puede realizar por tos caminos (el camino francés, el aragonés, etc.), se quiere guardar información a de éstos. Esta información se refiere al nombre (que los identifica), el número lómetros totales y el tiempo estimado para la realización del camino.

Cada camino se compone de distintas etapas que se identifican por un número lativo dentro de cada camino, y para cada una de ellas se desea saber el número lómetros, el tiempo estimado y las distintas localidades por las que pasa. Además iere recoger la localidad de salida y de llegada de la correspondiente etapa.

Se recogerán las distintas localidades por las que pasa cada camino. La mación que se recogerá de cada localidad será: nombre de la misma, Comunidad noma a la que pertenece y código postal. Se debe tener en cuenta que pueden ir localidades comunes a distintos caminos.

Se desea guardar información sobre los albergues para peregrinos que existen en nas de las localidades que pertenecen al camino. Esta información consta de: ore del albergue, capacidad y precio (si lo tuvieran).

Por último, se quiere registrar los peregrinos que realizan el camino. Para llevar control cada uno de ellos lleva un carnet que consta de un número de tificación, el nombre completo del peregrino, su dirección y las localidades por las ha ido pasando a lo largo del recorrido junto con el día que llegaron a dicha idad.

- Diseñar el esquema relacional indicando claves primarias y ajenas, nulos, opciones de borrado y modificación, etc.
- Efectuar en álgebra relacional y en SQL las siguientes consultas:

- Nombre de los peregrinos que han realizado el Camino de Santiago (se cuentan solo aquellos que han llegado a la localidad de Santiago).
- Nombre y Comunidad Autónoma de aquellas localidades que no posean albergues para peregrinos.
- Número de albergues por localidad pertenecientes a la primera etapa del Camino Aragonés.

Discusión del enunciado

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"Teniendo en cuenta que la peregrinación a Santiago se puede realizar por distintos caminos (el camino francés, el aragonés, etc.), se quiere guardar información acerca de éstos. Esta información se refiere al nombre (que los identifica), el número de kilómetros totales y el tiempo estimado para la realización del camino."*

Para recoger la información acerca de los distintos caminos de peregrinaje, se requiere una relación que los contenga, se crea así la relación CAMINO que tendrá los atributos "Nombre" (nombre del camino), "Kilómetros_T" (kilómetros del camino) y "Tiempo_T" (tiempo estimado de realización del camino). La clave primaria de esta relación será el atributo "Nombre", pues identifica de manera unívoca a cada uno de los posibles caminos. Ninguno de los atributos de esta relación podrá contener valores nulos.

CAMINO (Nombre, Kilómetros_T, Tiempo_T)

> *"Cada camino se compone de distintas etapas que se identifican por un número correlativo dentro de cada camino, y para cada una de ellas se desea saber el número de kilómetros, el tiempo estimado y las distintas localidades por las que pasa. Además se quiere recoger la localidad de salida y de llegada de la correspondiente etapa."*

La información acerca de las distintas etapas que componen un camino de peregrinación, se recogerá en una nueva relación ETAPA que contendrá los atributos "Nombre_C" (nombre del camino), "Número" (número de orden de la etapa dentro del camino), "Kilómetros_P" (kilómetros de la etapa), "Tiempo_P" (tiempo estimado de realización de la etapa), "Ciudad_s" (ciudad de salida) y "Ciudad_LL" (ciudad de llegada). Ninguno de los atributos de esta relación podrá contener valores nulos.

La clave primaria de esta relación estará formada conjuntamente por los atributos "Nombre_C" y "Número", pues identifican de manera unívoca a cada una de las posibles etapas. De este esquema de identificación se deduce que no puede haber

apas comunes a los diferentes caminos, por lo que si hubiera etapas de diferentes minos con el mismo recorrido, se considerarían como diferentes.

ETAPA (Nombre_C, Número, Kilómetros_P, Tiempo_P, Ciudad_S, Ciudad_LL)

"Se recogerán las distintas localidades por las que pasa cada camino. La información que se recogerá de cada localidad será: nombre de la misma, Comunidad Autónoma a la que pertenece y código postal. Se debe tener en cuenta que pueden existir localidades comunes a distintos caminos."

Se precisa una relación CIUDAD que recoja la información de las distintas calidades por las que pasan los caminos, los atributos de esta relación serán Nombre" (nombre de la ciudad), "Comunidad_Aut" (comunidad autónoma a la que ertenece la ciudad) y "Código_P" (código postal). La clave primaria es el atributo Nombre" pues identifica unívocamente a cada una de las localidades; podría haberse ensado en elegir el atributo "Código_P", sin embargo para ello, debería asegurarse ue no hubiera localidades que compartieran el código postal, por este motivo no lo onsideramos clave (ni primaria ni alternativa). Ninguno de los atributos de esta elación podrá tomar valores nulos.

CIUDAD (Nombre, Comunidad_Aut, Código_P)

"Se desea guardar información sobre los albergues para peregrinos que existen en algunas de las localidades que pertenecen al camino. Esta información consta de: nombre del albergue, capacidad y precio (si lo tuvieran)."

Para recoger la información sobre los albergues, se requiere una nueva relación ALBERGUE, sus atributos son "Nombre_A", "Capacidad", "Precio" y un atributo Ciudad". La clave primaria de ALBERGUE será "Nombre_A" mientras que el único atributo que puede tomar valores nulos será "Precio", pues éste puede no siempre ser conocido.

ALBERGUE (Nombre_A, Capacidad, Precio*, Ciudad)

"Por último, se quiere registrar los peregrinos que realizan el camino. Para llevar este control cada uno de ellos lleva un carnet que consta de un número de identificación, el nombre completo del peregrino, su dirección y las localidades por las que ha ido pasando a lo largo del recorrido junto con el día que llegaron a dicha localidad."

La información de los peregrinos se depositará en una relación PEREGRINO que contendrá los atributos "Número_I" (número de identificación), "Nombre_Completo" y "Dirección". El número de identificación será la clave primaria de la relación, y podría suponerse que el nombre completo identifica a los peregrinos, por lo que se

trataría de una clave alternativa, por último si se supone que podría no conocerse la dirección de un determinado peregrino, el atributo "Dirección" admitiría nulos.

PEREGRINO (Número_I, Nombre_Completo, Dirección*)

Además, como dice el enunciado, se debe registrar las ciudades por las que va pasando cada peregrino, esta información no se podrá recoger sólo con atributos ni en la relación PEREGRINO ni en la relación CIUDAD (pues como máximo se podría recoger una ciudad en cada peregrino o un peregrino en cada ciudad), se precisará una nueva relación que se comentará en el siguiente apartado.

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"Cada camino se compone de distintas etapas que se identifican por un número correlativo dentro de cada camino y para cada una de ellas se desea saber el número de kilómetros, el tiempo estimado y las distintas localidades por las que pasa. Además se quiere recoger la localidad de salida y de llegada de la correspondiente etapa."

Un camino tendrá varias etapas, pero cada etapa corresponde a un único camino, se sabe además que la identificación de la etapa está formada a partir de la del camino, así se tiene por tanto que el atributo "Nombre_C" de la relación ETAPA es una clave ajena que referencia a la relación CAMINO.

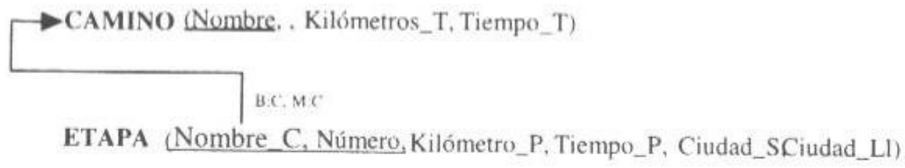


Figura 2.1.30

Opciones de borrado y modificación

Veamos qué sucede con las opciones de integridad referencial de la clave ajena, en el caso en que se borre un camino se habrán de borrar todas las etapas que le componen, pues éstas no tienen sentido de manera aislada sin pertenecer a un camino, por el mismo motivo, cuando se modifique el nombre de un camino habrá que realizar la misma modificación en todas las etapas que le componen. En ambos casos, tanto en el borrado como en la modificación, las opciones de integridad referencial serán en cascada.

Reglas ECA

Puede recogerse mediante disparadores la restricción que se deduce al razonar sobre el universo de discurso y darse cuenta de que un camino tiene que tener al

enos una etapa. Así, por ejemplo, no podría modificarse en la relación ETAPA la ave ajena "Nombre_C" cuando la tupla que se va a modificar sea la última de algún mino.

Es interesante además observar cómo el cálculo, tanto de los kilómetros del mino como del tiempo estimado de realización del camino, podría realizarse a partir e los correspondientes datos de las etapas que componen el camino, por lo que podría ensarse en la inclusión de disparadores que realizaran estos cálculos.

> *"Se recogerán las distintas localidades por las que pasa cada camino. La información que se recogerá de cada localidad será: nombre de la misma, Comunidad Autónoma a la que pertenece y código postal. Se debe tener en cuenta que pueden existir localidades comunes a distintos caminos."*

Para reflejar el recorrido de los distintos caminos podríamos pensar en recoger on dos atributos "Nombre_C" y "Número" en la relación CIUDAD la etapa a que orresponde esa ciudad, sin embargo esta solución asume que por esa localidad sólo uede pasar un camino y el enunciado afirma que puede haber distintas localidades omunes a varias ciudades. Por ello se precisa introducir una nueva relación RECORRIDO que mantenga tantas veces la ciudad como etapas de camino a la que sta pertenezca.

Los atributos de la relación RECORRIDO son "Nombre_C" (nombre del amino), "Número" (número de la etapa dentro del camino) y "Ciudad"; los dos rimeros atributos determinan la etapa del camino. Todos estos atributos serán no ulos y la clave primaria será el conjunto de los tres atributos, además, se tendrá que os dos primeros son una clave ajena que referencia a la relación ETAPA, mientras que el tercero es una clave ajena que referencia a la relación CIUDAD, pues tanto la tapa como la ciudad tienen que existir previamente.

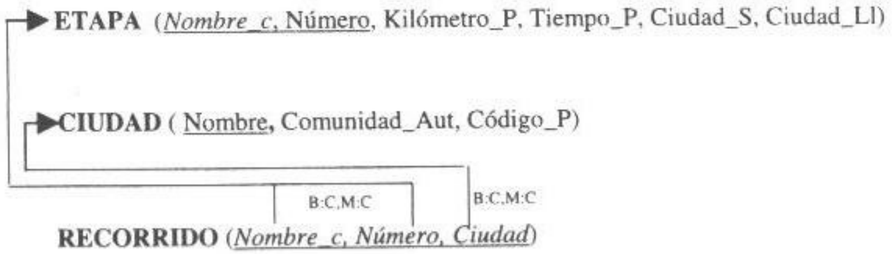


Figura 2.1.31

Opciones de borrado y modificación

Como opciones de integridad referencial serán admisibles en ambas claves ajenas tanto el borrado en cascada como el restringido (no permitir el borrado de una ciudad, ni de una etapa, mientras haya ejemplares que estén registrados como participantes en

un recorrido) ello dependerá del tipo de tratamiento que se prefiera realizar; por otra parte, la modificación en ambas claves ajenas será en cascada.

Reglas ECA

De manera similar a lo que ocurría entre los caminos y las etapas, puede recogerse mediante disparadores la restricción deducida de nuestro conocimiento del dominio de discurso de que toda etapa ha de recorrer como mínimo dos ciudades[25] (origen y destino). Así, por ejemplo, para poder crear una nueva etapa debería asegurarse previamente la presencia de las correspondientes ciudades de salida y llegada. Si la restricción se asegurara mediante programa, la creación en caso de que no existieran las ciudades debería realizarse de manera simultanea en la misma transacción.

Por otra parte, si en la BD se van a recoger solamente ciudades que pertenezcan a algún recorrido, debería asegurarse igualmente mediante disparadores la restricción de que toda ciudad en la relación CIUDAD pertenece a algún recorrido.

Por último, ha de observarse que en la relación CIUDAD, en particular, deberían aparecer como ejemplares tanto la ciudad de salida como la de llegada de cada una de las etapas, debería forzarse restricciones de integridad referencial entre las relaciones CIUDAD y ETAPA para asegurar que no puedan recogerse en ETAPA ciudades inexistentes. Se tiene por tanto que los atributos "Ciudad_s" y "Ciudad_LL" de la relación ETAPA son claves ajenas que referencian a la relación CIUDAD.

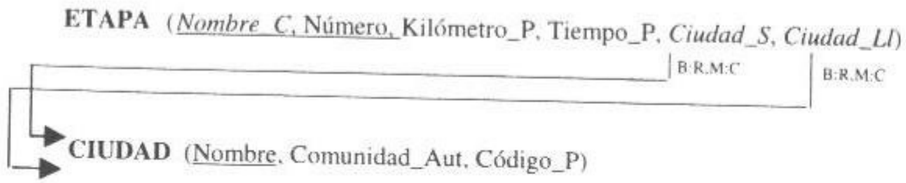


Figura 2.1.32

Opciones de borrado y modificación

En ambos casos, las opciones de integridad referencial serán borrado restringido (no se permitirá borrar una ciudad mientras que haya una etapa que se inicie o finalice en ella) y modificación en cascada.

Podría considerarse una alternativa que elimina la necesidad de las dos claves ajenas, "Ciudad_S" y "Ciudad_LL", si asumimos que en cada recorrido una ciudad se recorrerá siempre en el mismo orden e introducimos en la relación RECORRIDO un atributo "orden" que lo refleje, en ese caso, la ciudad de salida en cada etapa será

[25] Salvo que se admitieran etapas circulares, claro está.

aquella que tenga orden 1, mientras que la ciudad de llegada será aquella que tenga el mayor valor de orden en cada etapa, por lo que no sería preciso mantener su valor explícitamente en la relación ETAPA, pues ambos atributos podrían calcularse mediante disparadores a partir del atributo "Orden"[26]. Dado que no se hace mención explícita en el enunciado a este orden, nos quedamos con la solución inicialmente propuesta.

> *"Se desea guardar información sobre los albergues para peregrinos que existen en algunas de las localidades que pertenecen al camino. Esta información consta de: nombre del albergue, capacidad y precio (si lo tuvieran)."*

El atributo ciudad en la relación ALBERGUE determinaría para cada albergue la ciudad a la que pertenece, se trata por tanto de una clave ajena que referencia a la relación CIUDAD.

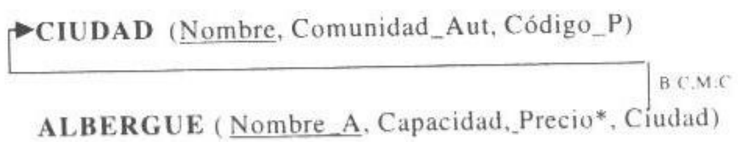


Figura 2.1.33

Opciones de borrado y modificación

La anterior restricción de integridad referencial podría tener opción de borrado en cascada o restringido dependiendo de la semántica pretendida, mientras que la modificación se hará en cascada.

Reglas ECA

Si suponemos que no puede darse la posibilidad de ciudades sin albergues, podría controlarse con disparadores la existencia de al menos un albergue en cualquier ciudad.

> *"Por último, se quieren registrar los peregrinos que realizan el camino. Para llevar este control cada uno de ellos lleva un carnet que consta de un número de identificación, el nombre completo del peregrino, su dirección y las localidades por las que ha ido pasando a lo largo del recorrido junto con el día que llegaron a dicha localidad."*

Para registrar las ciudades por las que va pasando cada peregrino se crea una nueva relación CAMINO_PEREGRINO que recoja por cada peregrino todas aquellas ciudades por las que éste ha pasado y respectivamente, para cada ciudad, todos

[26] Naturalmente "orden" no debería admitir nulos.

aquellos peregrinos que han pasado por ella. En esta relación se podrá además recoger la fecha en que este paso ha sucedido.

La relación CAMINO_PEREGRINO tendrá los atributos "Número_I" (número de identificación, clave ajena que referencia a PEREGRINO), "Nombre_C" (nombre de ciudad, clave ajena que referencia a CIUDAD) y "Fecha_Paso" (fecha en que el peregrino identificado mediante el atributo "Número_I" pasa por la ciudad identificada en "Nombre_C"). La clave primaria de esta relación la constituirán el conjunto de los tres atributos, por lo que ninguno de ellos podrá ser nulo. Si no se hubiera incluido el atributo "Fecha_Paso" como parte de la clave primaria, podría haberse dado la situación de que el mismo peregrino realice varias veces el mismo camino (e incluso caminos distintos que compartan la misma ciudad) y no se podría recoger más que uno de sus pasos por esa ciudad.

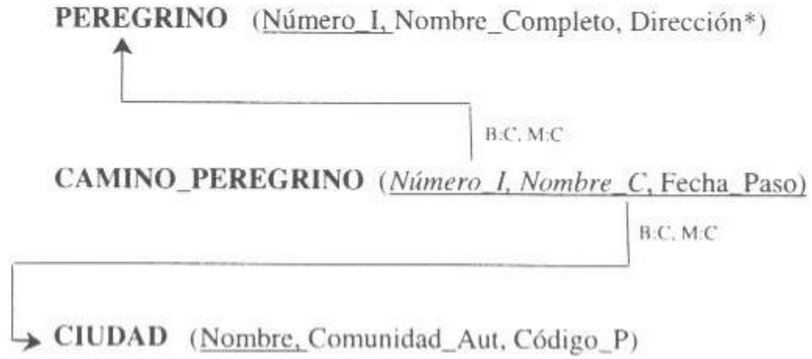


Figura 2.1.34

Nótese cómo aunque se han recogido las ciudades por las que cada peregrino ha pasado, no se ha recogido explícitamente el camino realizado por cada peregrino, si se hubiera querido registrar los caminos que un peregrino ha realizado (podría realizar más de uno) y los peregrinos que han realizado un camino, habría de incluirse una nueva relación con esta información y que tendría claves ajenas a las relaciones PEREGRINO y CAMINO.

Opciones de borrado y modificación

Las opciones de integridad referencial en ambas claves ajenas podrán ser: la de borrado tanto en cascada como restringido, mientras que la de modificación será en cascada.

Reglas ECA

Podría controlarse mediante disparadores la restricción de que todo peregrino debería haber pasado por al menos una ciudad, es decir, que por cada tupla en la

ción PEREGRINO haya al menos una tupla en la relación CAMINO_
REGRINO. En el caso de que esta hipótesis no fuera aplicable, por ejemplo si los
egrinos se registraran antes de iniciar el camino, no sería preciso, pues serían
nisibles los peregrinos que no han pasado por ninguna ciudad.

RAFO RELACIONAL

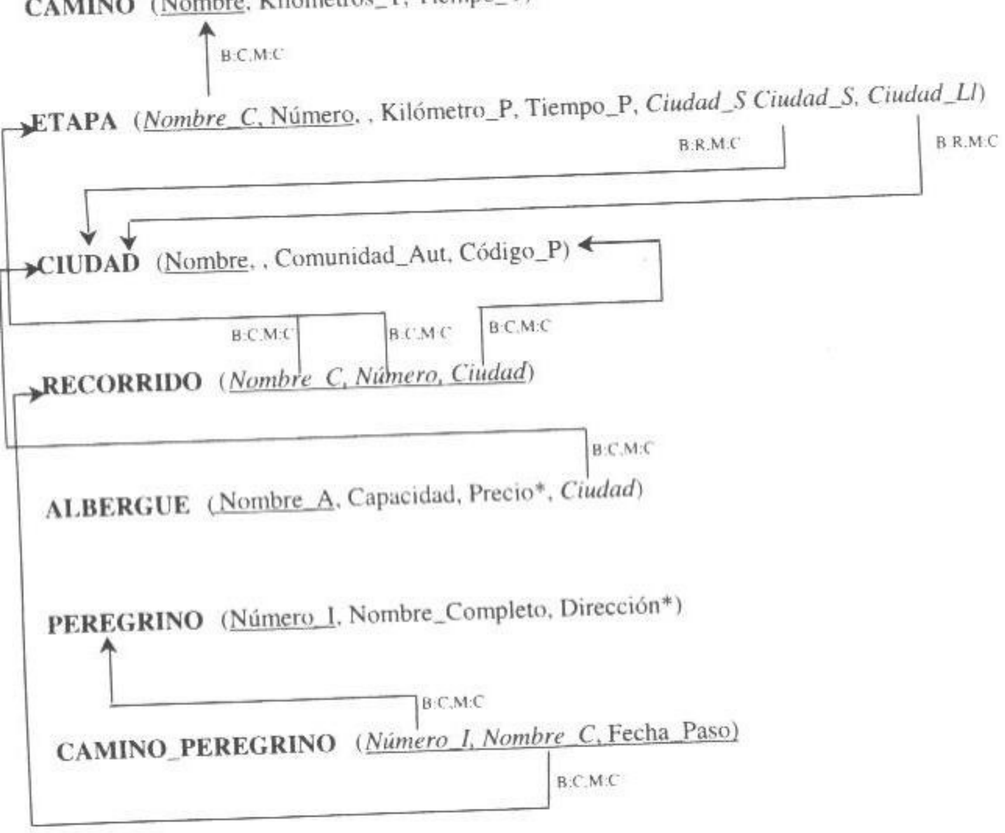


Figura 2.1.35

SCRIPT EN SQL 92

```
CREATE DOMAIN Nombre_Válido CHAR(40);

CREATE TABLE Camino (
        Nombre Nombre_Válido,
        Kilómetros_T INTEGER NOT NULL,
        Tiempo_T INTEGER NOT NULL,
PRIMARY KEY (Nombre)
);
```



```
CREATE TABLE Ciudad (
        Nombre Nombre_Válido,
        Comunidad_Aut Nombre_Válido
        Código_P CHAR(5) NOT NULL,
PRIMARY KEY (Nombre)
);

CREATE TABLE Etapa (
        Nombre_C Nombre_Válido,
        Número INTEGER,
        Kilómetros_P INTEGER NOT NULL,
        Tiempo_P INTEGER NOT NULL,
        Ciudad_S Nombre_Válido NOT NULL,
        Ciudad_LL Nombre_Válido NOT NULL,
PRIMARY KEY (Nombre_C, Número),
FOREIGN KEY (Nombre_C) REFERENCES Camino
        ON UPDATE CASCADE
        ON DELETE CASCADE,
FOREIGN KEY (Ciudad_S) REFERENCES Ciudad
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
FOREIGN KEY (Ciudad_S) REFERENCES Ciudad
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);

CREATE TABLE Recorrido (
        Nombre_C Nombre_Válido,
        Número INTEGER,
        Ciudad Nombre_Válido,
PRIMARY KEY (Nombre_C, Número, Ciudad),
FOREIGN KEY (Nombre_C, Número) REFERENCES Etapa
        ON UPDATE CASCADE
        ON DELETE CASCADE,
FOREIGN KEY (Ciudad) REFERENCES Ciudad
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE Albergue (
        Nombre_A Nombre_Válido,
        Capacidad INTEGER NOT NULL,
        Precio INTEGER,
        Ciudad Nombre_Válido NOT NULL,
PRIMARY KEY (Nombre_A),
FOREIGN KEY (Ciudad) REFERENCES Ciudad
        ON UPDATE CASCADE
```

```
    ON DELETE CASCADE

REATE TABLE Peregrino (
    Número_I INTEGER,
    Nombre_Completo Nombre_Válido NOT NULL,
    Dirección CHAR(100),
RIMARY KEY (Número_I)

CREATE TABLE Camino_Peregrino (
    Número_I INTEGER,
    Nombre_C Nombre_Válido,
    Fecha_Paso DATE,
PRIMARY KEY (Número_I, Nombre_C, Fecha_Paso),
FOREIGN KEY (Número_I) REFERENCES Peregrino
    ON UPDATE CASCADE
    ON DELETE CASCADE
FOREIGN KEY (Nombre_C) REFERENCES Ciudad
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
```

CONSULTAS

1. Nombre de los peregrinos que han realizado el Camino de Santigo (se cuentan sólo aquellos que han llegado a la localidad de Santiago).

Sql

```
SELECT Nombre_Completo
FROM Peregrino, Camino_Peregrino
WHERE Peregrino.Número_I = Camino_Peregrino.Número_I
        AND Camino_Peregrino.Nombre_C='Santiago'
```

Álgebra relacional

$$\Pi_{Nombre_Completo}((PEREGRINO) * (\sigma_{Nombre_C='Santiago'}(CAMINO_PEREGRINO)))$$

2. Nombre y Comunidad Autónoma de aquellas localidades que no posea N albergues para peregrinos.

Sql

```
SELECT Nombre
FROM Ciudad
WHERE Nombre NOT IN
```



```
(SELECT DISTINCT Ciudad FROM Albergue)
```

Álgebra relacional

$$\Pi_{Nombre}(CIUDAD) - \Pi_{Ciudad}(ALBERGUE)$$

3. Número de albergues por localidad pertenecientes a la primera etapa del Camino Aragonés.

Sql

```
SELECT Nombre, Count(Nombre_A)
FROM Ciudad, Albergue
WHERE Ciudad.Nombre=Albergue.Ciudad
GROUP BY Nombre
```

Álgebra relacional

$$(\Pi_{Nombre,\ Nombre_A}((CIUDAD) * (ALBERGUE)) \text{ AGRUPACIÓN POR (Nombre), CUENTA(Nombre_A)}$$

PROBLEMA 7: PROYECTOS DE INVESTIGACIÓN

Enunciado

En la Universidad Carlos III se desea llevar un control sobre los proyectos de investigación que se desarrollan. Para ello se decide emplear una base de datos que contenga toda la información sobre los proyectos, departamentos, grupos de investigación y profesores. Esta información se detalla a continuación:

Un departamento se identifica por su nombre (Informática, Ingeniería, etc). Tiene una sede situada en un determinado campus, un teléfono de contacto y un Director, que ha de ser un profesor de esa Universidad.

Dentro de un departamento se crean grupos de investigación. Cada grupo tiene un nombre único dentro del departamento (pero que puede ser el mismo en distintos departamentos) y está asociado a un área de conocimiento (bases de datos, inteligencia artificial, sistemas y comunicaciones, etc). Cada grupo tiene un líder que ha de ser profesor de la Universidad.

Un profesor está identificado por su DNI. De él se desea saber el nombre, titulación, años de experiencia en investigación, grupo de investigación en el que desarrolla su labor y proyectos en los que trabaja.

Cada proyecto de investigación tiene un nombre, un código único, un presupuesto, fechas de inicio y terminación y un grupo que lo desarrolla. Por otro lado, puede estar financiado por varios programas. Dentro de cada programa cada proyecto tiene un número asociado y una cantidad de dinero financiada (por ejemplo, el proyecto "BDC - Bases de Datos en Colores" tiene el número 3113 dentro del programa "Ayudas al Pobre Investigador" que le financia con 30.000 pesetas).

Un profesor puede participar en varios proyectos. En cada proyecto se incorpora en una determinada fecha y cesa en otra, teniendo una determinada dedicación (en horas a la semana) durante ese período.

- Diseñar el esquema relacional indicando claves primarias y ajenas, nulos, opciones de borrado y modificación, etc.
- Efectuar además en álgebra relacional y en SQL las siguientes consultas:
  - Profesores que trabajan en el grupo del prestigioso científico "Juan Pérez".
  - Programas que financian los proyectos del grupo que dirige ese famoso profesor.

Discusión del enunciado

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"Un departamento se identifica por su nombre (Informática, Ingeniería, etc). Tiene una sede situada en un determinado campus, un teléfono de contacto y un Director, que ha de ser un profesor de esa Universidad."*

Para almacenar todos los datos anteriores necesitamos la relación DEPARTAMENTO, cuya clave primaria será el nombre de departamento. Además debemos incluir una nueva relación SEDE que almacene los datos sobre las distintas sedes de la Universidad ("Campus" y "Nombre_Sede"),

DEPARTAMENTO (Código_D, Teléfono, Director*)

SEDE (Nombre_Sede, Campus)

Podemos considerar que el atributo "Director" puede admitir nulos, y de esta forma permitir introducir un departamento de nueva creación al que todavía no se le haya asignado ningún director.

> *"Dentro de un departamento se crean grupos de investigación. Cada grupo tiene un nombre único dentro del departamento (pero que puede ser el mismo en distintos departamentos) y está asociado a un área de conocimiento (bases de datos, inteligencia artificial, sistemas y comunicaciones, etc.). Cada grupo tiene un líder que ha de ser profesor de la Universidad."*

Del análisis de este párrafo se deduce la existencia de la relación GRUPO. Como el nombre del grupo es único sólo dentro de cada departamento, la identificación de cada grupo requiere los atributos "Nombre_G" y "Código_D".

GRUPO (Nombre_G, Código_D, Área, Líder*)

Por los mismos motivos que en el caso anterior podemos admitir que el atributo Lider tome valores nulos

> *"Un profesor está identificado por su DNI. De él se desea saber el nombre, titulación, años de experiencia en investigación, grupo de investigación en el que desarrolla su labor y proyectos en los que trabaja."*

Tenemos una nueva relación PROFESOR, cuya clave primaria es "DNI", pues es el que identifica unívocamente cada ocurrencia de la relación.

PROFESOR (DNI, Nombre_Prof , Titulación, Experiencia*, Grupo*)

No debemos incluir Proyectos como atributo de la relación PROFESOR, puesto que un profesor puede trabajar en varios proyectos a la vez y por tanto sería un atributo multivaluado que no está permitido en el modelo relacional.

Además podemos considerar que el atributo "Experiencia" puede admitir valores nulos, al igual que el atributo Grupo, pues podemos considerar que cuando entra un profesor nuevo, todavía no se le ha asignado grupo.

> *"Cada proyecto de investigación tiene un nombre, un código único, un presupuesto, fechas de inicio y terminación y un grupo que lo desarrolla".*

Tenemos una relación PROYECTO donde la clave primaria es "Código_P" y al igual que la anterior, no podemos incluir el atributo Programas, pues éste sería multivaluado, lo que iría en contra de los elementos permitidos del modelo relacional.

Además, podemos considerar el nombre del proyecto como clave alternativa, pues en principio se puede considerar único. Los atributos "Presupuesto" y Fecha de terminación ("F_Fin"), a priori pueden ser desconocidos, por lo que pueden admitir valores nulos.

PROYECTO (Código_P, **Nombre_P**, Presupuesto*, F_Ini, F_Fin*, Grupo)

> *"Dentro de cada programa cada proyecto tiene un número asociado y una cantidad de dinero financiada (por ejemplo, el proyecto "BDC _ Bases de Datos en Colores" tiene el número 3113 dentro del programa "Ayudas al Pobre Investigador" que le financia con 30.000 pesetas)."*

Tenemos la relación PROGRAMA, donde en principio, podemos considerar el número de proyecto ("Número_P"), junto con el nombre del programa ("Nombre_Prog") como clave primaria, puesto que un proyecto puede ser financiado por varios programas.

PROGRAMA (Nombre_Prog, Número_P, Financiación)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

> *"Un departamento se identifica por su nombre (Informática, Ingeniería, etc). Tiene una sede situada en un determinado campus...".*

Un departamento puede estar repartido en varias sedes, pero en cada sede existen varios departamentos, por ejemplo en el campus de Leganés, la Escuela Politécnica Superior tiene, entre otros, los departamentos de Informática y Matemáticas. Esto nos obliga a introducir una nueva relación UBICACIÓN cuya clave primaria será la concatenación de las claves primarias de las relaciones DEPARTAMENTO y SEDE, además cada una de ellas será clave ajena que referencia a la relación de la cual proviene.

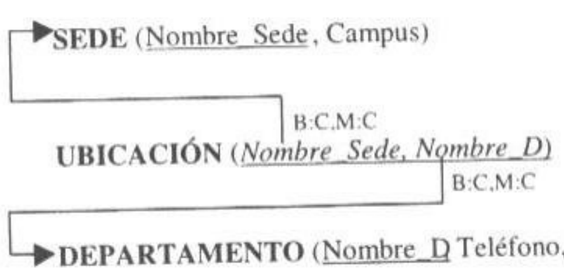


Figura 2.1.36

Opciones de borrado y modificación

Las opciones de integridad referencial de ambas claves ajenas pueden ser en cascada.

Reglas ECA

Si consideramos que todo departamento tiene que tener al menos una sede y que en cualquier sede al menos hay un departamento, debemos asegurar estas cardinalidades mediante disparadores.

> *"... todo departamento tiene un director que ha de ser profesor de la universidad."*

Existe una interrelación entre DEPARTAMENTO y PROFESOR, que se efectúa a través de la clave ajena "Director", de la relación DEPARTAMENTO, como se muestra en la figura.

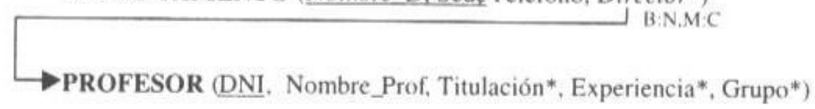


Figura 2.1.37

Opciones de borrado y modificación

Si se elimina un profesor director de algún departamento, se debe controlar que no se elimine el departamento, por tanto el borrado debe ser con puesta a nulos (BN) o restringido, sin embargo la modificación puede hacerse en cascada.

> *"Dentro de cada departamento se crean grupos de investigación..., además cada grupo tiene un líder que es un profesor... Se desea conocer el grupo al que pertenece cada profesor ".*

Por un lado debemos interrelacionar DEPARTAMENTO y GRUPO. Esto se hace a través del atributo "Código_D", de la relación GRUPO, que es clave ajena que referencia a DEPARTAMENTO.

Por otro lado debemos relacionar GRUPO con PROFESOR, a través del atributo Líder, clave ajena de la relación GRUPO, que referencia a PROFESOR.

Por último, y puesto que todo profesor pertenece a un grupo, debemos relacionar PROFESOR con GRUPO a través del atributo "Grupo" de la relación PROFESOR. Este atributo es un atributo compuesto que contiene el nombre del grupo ("Nombre_G") y el departamento al que pertenece el grupo ("Código_D").

De esta forma el subesquema relacional queda de la siguiente forma:

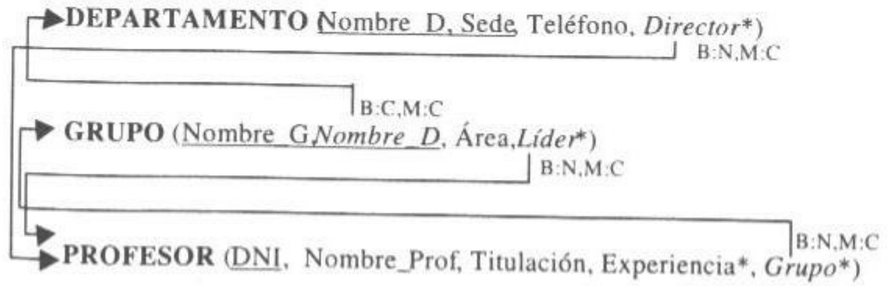


Figura 2.1.38

Entre las relaciones DEPARTAMENTO, GRUPO y PROFESOR, existe un ciclo referencial que impide la creación de las mismas según las especificaciones del SQL_92. Este problema se puede solucionar creando las tablas sin las claves ajenas y posteriormente modificarlas añadiendo las restricciones de integridad referencial, como se muestra en el script.

Opciones de borrado y modificación

Si se elimina un departamento no tiene sentido dejar los grupos pertenecientes al mismo, por tanto el borrado se hace en cascada (BC), al igual que la modificación (MC).

Además, si se elimina un grupo se debe poner a nulos (BN) la clave ajena "Grupo" de la tabla profesor, porque de no hacerlo eliminaríamos a todos los profesores pertenecientes al grupo y esto no se puede admitir. En cuanto a la modificación, debe hacerse en cascada (MC).

En cuanto a la clave ajena de GRUPO, "Líder", su eliminación sería con puesta a nulos (BN), porque en caso contrario, si cambia el líder de un grupo lo eliminaríamos de la relación PROFESOR, caso que no está admitido.

Reglas ECA

Deberíamos controlar mediante un disparador que, cuando se inserte un nuevo grupo, exista el departamento correspondiente y, en caso contrario insertar primero el departamento.

"Cada proyecto de investigación tiene un nombre..., y un grupo que lo desarrolla. Por otro lado, puede estar financiado por varios programas".

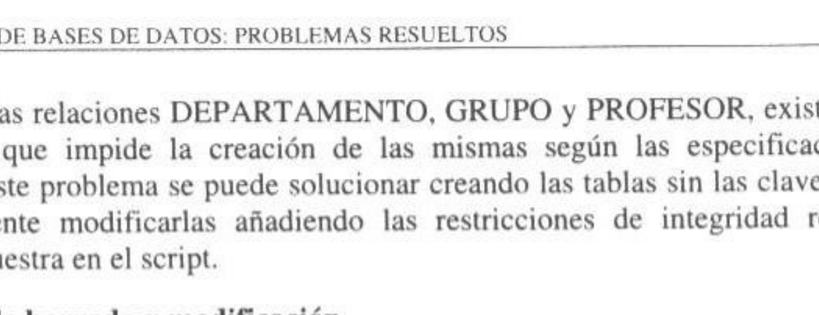


Figura 2.1.39

Como no se nos dice nada en contra, podemos suponer que un mismo programa puede financiar a varios proyectos, pero dentro de cada programa los proyectos llevan asociado un código, por tanto no es necesario construir una nueva relación FINANCIA (relación intermedia) cuya clave primaria será la concatenación de las claves primarias de PROYECTO y PROGRAMA, que además serán respectivamente, claves ajenas de las relaciones de las que provienen. Por otro lado debemos incluir también en esta nueva relación los atributos "Número_P" y "Financiación" que en un principio habíamos considerado dentro de la relación PROGRAMA.

Cada proyecto de investigación es desarrollado por un grupo, por otra parte, un mismo grupo puede llevar a cabo varios proyectos. La manera de hacer constar este hecho es considerar el atributo "Grupo" de la relación PROYECTO como clave ajena que referencia a GRUPO.

Opciones de borrado y modificación

El borrado y la modificación de la clave ajena grupo puede ser en cascada (BC/MC), pues si se elimina o se modifica un grupo, es lógico que se eliminen o modifiquen todos los proyectos que llevaba dicho grupo.

Lo mismo sucede con las claves ajenas "Código_P" y "Nombre_Prog" de la relación FINANCIA.

Restricciones de rechazo

Deberíamos comprobar mediante un CHECK que la fecha de inicio del proyecto es anterior a la fecha de finalización.

Se debería controlar mediante una aserción que el presupuesto de cada proyecto sea siempre mayor o igual que la suma de las financiaciones otorgadas por cada programa a cada uno de los proyectos.

"Un profesor puede participar en varios proyectos. En cada proyecto se incorpora en una determinada fecha y cesa en otra, teniendo una determinada dedicación (en horas a la semana) durante ese período".

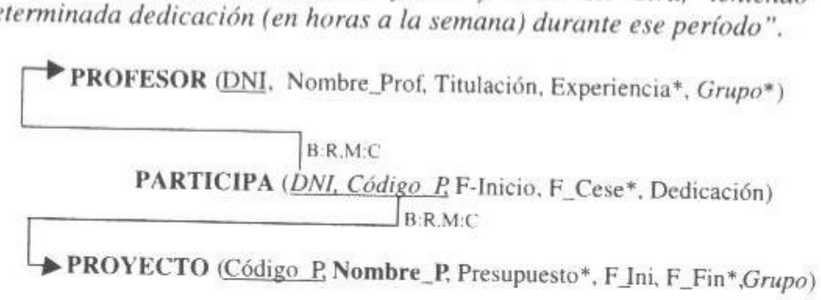


Figura 2.1.40

Lógicamente en un proyecto pueden participar varios profesores, por lo tanto necesitamos incluir en el esquema una relación intermedia PARTICIPA, cuya clave primaria estará formada por las claves primarias de las relaciones PROFESOR y PROYECTO, que a su vez serán claves ajenas de sus respectivas relaciones. Además, esta nueva relación contendrá las fechas de inicio y cese de cada investigador en el proyecto y la dedicación; la fecha de cese puede admitir valores nulos, pues puede darse el caso de profesores que continúen participando en un determinado proyecto. Si pudiera darse el caso de que el mismo profesor participase en el mismo proyecto en diferentes períodos de tiempo, se habría de incluir la fecha de inicio como parte de la clave primaria de PARTICIPA.

pciones de borrado y modificación

Los borrados de las dos claves ajenas de la relación PARTICIPA deben ser stringidos para poder guardar la información acerca de la fechas y la dedicación a s proyectos, de cada profesor. La modificación en cualquiera de los dos casos es en scada.

eglas ECA

Cuando se introduzca un nuevo proyecto, en él deberá participar al menos un ofesor, lo que podríamos controlar con un disparador que, por ejemplo, cada vez que sertemos un nuevo proyecto, se le asigne al menos un profesor de los existentes y en so contrario rechazar la inserción.

Restricciones de rechazo

Se debería controlar con un CHECK que las fechas de inicio de participación en da proyecto sean anteriores a las fechas de cese.

RAFO RELACIONAL

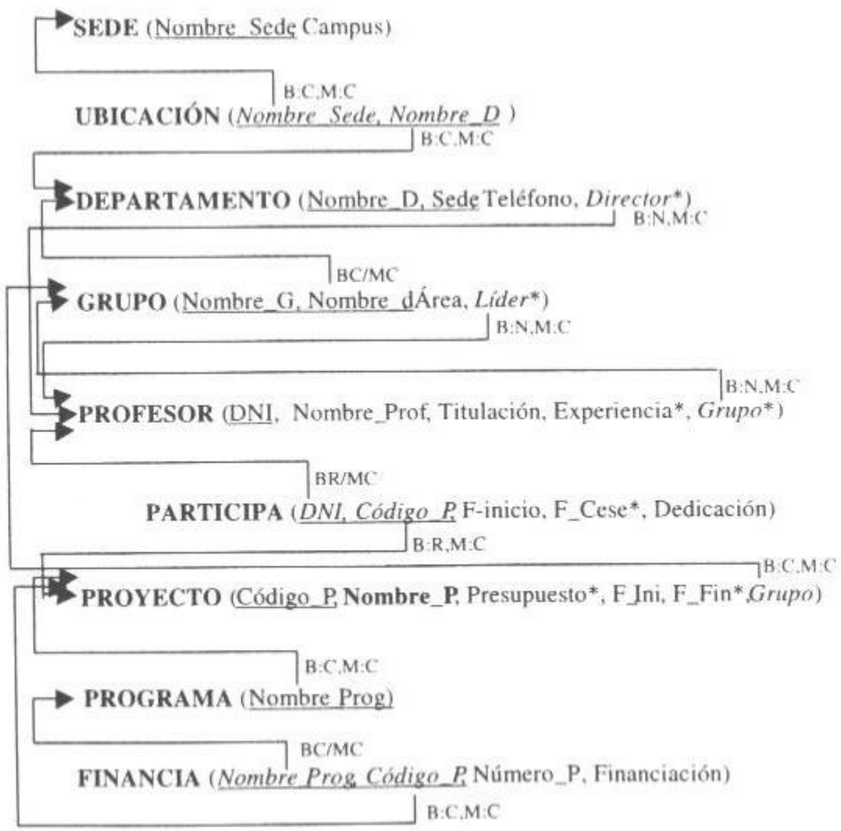


Figura 2.1.41

SCRIPT EN SQL_92

```
CREATE DOMAIN Nombre_Válido CHAR(30);
CREATE DOMAIN Tipo_Código CHAR(5);
CREATE DOMAIN Tipo_DNI CHAR(9);

CREATE TABLE Sede (
      Nombre_Sede Nombre_Válido,
      Campus Nombre_Válido,
PRIMARY KEY (Nombre_Sede)
);

CREATE TABLE Departamento (
      Código_D Nombre_Válido,
      Sede Nombre_Válido,
      Teléfono CHAR(9) NOT NULL,
      Director Nombre_Válido,
PRIMARY KEY (Nombre_P, Sede)
);

CREATE TABLE Ubicación (
      Código_D Nombre_Válido,
      Nombre_Sede Nombre_Válido,
PRIMARY KEY (Nombre_P, Sede)
FOREIGN KEY (Código_D) REFERENCES Departamento
      ON UPDATE CASCADE,
FOREIGN KEY (Nombre_Sede) REFERENCES Sede
      ON UPDATE CASCADE
);

CREATE TABLE Grupo (
      Nombre_G Nombre_Válido,
      Código_D Nombre_Válido,
      Área Nombre_Válido,
      Líder Nombre_Válido,
PRIMARY KEY (Nombre_G, Código_D)
);

CREATE TABLE Profesor (
      DNI Tipo_DNI,
      Nombre_Prof  Nombre_Válido,
      Titulación CHAR(20) NOT NULL,
      Experiencia INTEGER,
      Grupo Nombre_Válido,
PRIMARY KEY (DNI)
);
```

```
ALTER TABLE Departamento
      ADD FOREIGN KEY (Director) REFERENCES Profesor
       ON DELETE SET NULL
       ON UPDATE CASCADE;

ALTER TABLE Grupo
      ADD FOREIGN KEY (Código_D) REFERENCES Departamento
       ON DELETE CASCADE
       ON UPDATE CASCADE,
      ADD FOREIGN KEY (Líder) REFERENCES Profesor
       ON DELETE SET NULL
       ON UPDATE CASCADE;

      ALTER TABLE Profesor
      ADD FOREIGN KEY (Grupo) REFERENCES Grupo
       ON DELETE SET NULL
       ON UPDATE CASCADE;

CREATE TABLE Proyecto (
      Código_P Tipo_Código,
      Nombre_P Nombre_Válido,
      Presupuesto INTEGER NOT NULL,
      F_Ini DATE NOT NULL,
      F_Fin DATE,
      Grupo Nombre_Válido,
PRIMARY KEY (Código_P),
UNIQUE (Nombre_P),
CHECK ( F_Ini<F_Fin),
FOREIGN KEY (Grupo) REFERENCES Grupo
      ON UPDATE CASCADE
      ON DELETE CASCADE
);

CREATE TABLE Participa (
      DNI Tipo_DNI,
      Código_P Tipo_Código,
      F_Inicio INTEGER NOT NULL,
      F_Cese DATE,
      Dedicación INTEGER NOT NULL,
PRIMARY KEY (DNI, Código_P),
CHECK (F_Inicio < F_Cese),
FOREIGN KEY (DNI) REFERENCES Profesor
      ON UPDATE CASCADE,
FOREIGN KEY (Código_P) REFERENCES Proyecto
      ON UPDATE CASCADE
);
```



```
CREATE TABLE Programa (
      Nombre_Prog Nombre_Válido,
PRIMARY KEY (Nombre_Prog));

CREATE TABLE Financia (
      Nombre_Prog Nombre_Válido,
      Código_P Tipo_Código,
      Número_P Tipo_Código,
      Financiación INTEGER,
PRIMARY KEY (Nombre_Prog, Código_P),
FOREIGN KEY (Código_P) REFERENCES Proyecto
      ON UPDATE CASCADE
      ON DELETE CASCADE,
FOREIGN KEY (Nombre_Prog) REFERENCES Programa
      ON UPDATE CASCADE
      ON DELETE CASCADE);


CREATE ASSERTION Subvención
      CHECK ((SELECT Presupuesto
            FROM Proyecto NATURAL JOIN Financia) <
                  (SELECT SUM (Financiación)
                  FROM Financia NATURAL JOIN Programa));
```

CONSULTAS

1. Profesores trabajan en el grupo del prestigioso científico "Juan Pérez".

   Sql

   ```
   SELECT Nombre_Prof
   FROM Profesor, Grupo
   WHERE Líder='Juan Perez' AND Profesor.Grupo=Grupo.Nombre_G;
   ```

   Álgebra relacional

   $$\Pi_{Nombre_Prof}(\Pi_{Nombre_Prof\ ,\ Grupo}(PROFESOR) * \Pi_{Nombre_G}(\sigma_{Líder='Juan\ Pérez'}(GRUPO)))$$

2. Programas financian los proyectos del grupo que dirige ese famoso profesor.

   Sql

   ```
   SELECT Nombre_Prog
   FROM Programa, Financia, Proyecto, Grupo
   WHERE Programa.Nombre_Prog= Financia.Nombre_Prog
         AND Financia.Código_P = Proyecto.Código_P
         AND Proyecto.Grupo=Grupo.Nombre_G AND Líder='Juan Pérez';
   ```

Álgebra relacional

$$\Pi_{Nombre_Prog}((PROGRAMA)^* \Pi_{Nombre_Prog}((FINANCIA)^*\Pi_{Código_P}((\Pi_{Código_P,grupo}(PROYECTO))^*(\Pi_{Nombre_G}(\sigma_{Nombre='Juan\ Pérez'}(GRUPO)))))$$

ROBLEMA 8: NAVES ESPACIALES RELACIONAL

nunciado

El Ministerio de Defensa español desea desarrollar un Sistema de Información ara la nave espacial Casiopea 2000 que próximamente se lanzará al espacio. La Base e Datos deberá contener la siguiente información:

La nave espacial se compone de distintas dependencias, y cada una de ellas tiene n nombre, un código (único para cada dependencia), una función y una localización. ada dependencia está bajo el control de un determinado Servicio, identificado por un ombre y una clave. Todo servicio de la nave (Servicio de Operaciones, Comando y ontrol, Seguridad, etc.) ha de estar asignado por lo menos a una dependencia.

Se quiere llevar al día una relación de la tripulación de la nave. Esta información ontiene el nombre, código, categoría, antigüedad, procedencia y situación administrativa (en servicio, de baja...). Cada tripulante está asignado a una dependencia que se esea conocer, así como la cámara en la que se aloja. Una cámara es una dependencia ue posee dos características propias, su categoría y su capacidad.

Por otra parte, se desean conocer los planetas que ha visitado cada miembro de la ripulación y el tiempo que permanecieron en ellos para saber las personas con quien e puede contar a la hora de realizar una exploración interplanetaria. De cada planeta e conoce su nombre y código, la galaxia y coordenadas en las que se encuentra. Algunos planetas se encuentran poblados por diversas razas, cada una en una cierta antidad de individuos. De cada raza se almacena información sobre el nombre. oblación total y dimensiones medias (altura, anchura, peso).

- Diseñe el esquema relacional. Indique en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas, especifique los modos de borrado y modificación. Marque con un asterisco (*) los campos que acepten valores nulos. Y por último, indique si existe algún atributo que exija definir alguna restricción a fin de mantener la integridad de la base de datos.
- Escriba las siguientes consultas en SQL y en álgebra relacional:
  - Razas que habitan en los planetas visitados por Flash Gordon
  - Cámaras que han llegado a su capacidad máxima.
  - Nombre de los tripulantes asignados a alguna dependencia de seguridad.

Discusión del enunciado

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"La nave espacial se compone de distintas dependencias, y cada una de ellas tiene un nombre, un código (único para cada dependencia), una función y una localización. Cada dependencia está bajo el control de un determinado Servicio, identificado por un nombre y una clave."*

Tras la lectura del primer párrafo, parece claro que DEPENDENCIA va a ser una relación, que tendrá como atributos un código (que será clave), un nombre de la dependencia (el cual supondremos que es único para cada dependencia, y por tanto será clave alternativa), una función, y una localización. Supondremos que toda dependencia tendrá obligatoriamente un nombre (de hecho, la hemos considerado como clave alternativa), pero permitiremos que tanto la función como la localización puedan tomar valores nulos. SERVICIO será también una relación, y sus dos únicos atributos, el nombre y la clave, formarán la clave de la relación.

SERVICIO (Clave, Nombre_S)

DEPENDENCIA (Código_D, **Nombre_D**, Función*, Localización*)

> *"Se quiere llevar al día una relación de la tripulación de la nave. Esta información contiene el nombre, código, categoría, antigüedad, procedencia y situación administrativa (en servicio, de baja,...),...Una cámara es una dependencia que posee dos características propias, su categoría y su capacidad".*

Para almacenar los tripulantes de la nave, utilizaremos una tabla TRIPULACIÓN, que tendrá como atributos el nombre, el código (que será clave primaria), la categoría, antigüedad, procedencia y situación administrativa de cada tripulante. Consideraremos que tanto la antigüedad como la procedencia de cada tripulante pueden ser datos desconocidos, por lo que admitirán valores nulos. El enunciado también nos habla de un tipo de dependencia especial, las cámaras, cuya función es la de servir de alojamiento a los tripulantes de la nave. Aunque posteriormente resolveremos cómo tratar el hecho de que una cámara es un tipo especial de dependencia, de momento simplemente crearemos una relación que tendrá como atributos el código de la cámara (que será clave), la categoría y la capacidad.

CÁMARA (Código_C, Categoría, Capacidad)

TRIPULACIÓN (Código_T, Nombre_T, Categoría, Antigüedad*, Procedencia*, Adm)

> *"Por otra parte, se desean conocer los planetas que ha visitado cada miembro de la tripulación y el tiempo que permanecieron en ellos para saber*

las personas con quien se puede contar a la hora de realizar una exploración interplanetaria. De cada planeta se conoce su nombre y código, la galaxia y coordenadas en las que se encuentra. Algunos planetas se encuentran poblados por diversas razas, cada una en una cierta cantidad de individuos. De cada raza se almacena información sobre el nombre, población total y dimensiones medias (altura, anchura, peso)".

En este último párrafo detectamos la presencia de las últimas dos relaciones ásicas: PLANETA y RAZA. Con respecto a los planetas, el enunciado nos dice que endrá un código (que será clave primaria), un nombre, la galaxia en la que se ncuentra y sus coordenadas. Cada uno de éstos será un atributo, aunque onsideraremos que tanto el nombre como las coordenadas serán únicos para cada laneta, y por tanto podremos considerarlos como claves alternativas. En cuanto a la alaxia, podría considerarse la posibilidad de crear una nueva relación (por ejemplo, guardar información adicional acerca de cada galaxia), sin embargo, a falta de más nformación, lo trataremos simplemente como un atributo. En cuanto a la relación RAZA, tendrá un nombre, que supondremos único y por tanto será la clave primaria de la relación, y una serie de atributos: la altura, anchura y peso medio de la raza, y la población total de dicha raza en todo el universo. El enunciado también nos exige almacenar la población de una raza presente en un planeta. Esta información no podemos guardarla en ninguna de estas dos relaciones, ya que un planeta puede estar habitado por muchas razas, cada una de ellas con sus poblaciones parciales, y una raza puede vivir en distintos planetas. Posteriormente veremos dónde situar este atributo.

PLANETA (Código_P, **Nombre_P**, Galaxia, **Coordenadas**)

RAZA (Nombre_R, Altura, Anchura, Peso, Población_Total)

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"...Cada dependencia está bajo el control de un determinado Servicio, identificado por un nombre y una clave. Todo servicio de la nave (Servicio de Operaciones, Comando y Control, Seguridad, etc.) ha de estar asignado por lo menos a una dependencia"

Comenzaremos tratando el hecho de que cada dependencia esté bajo el control de un determinado servicio y que cada uno de éstos controle, al menos, una dependencia. Para controlar la primera restricción, puesto que toda dependencia está bajo el control de un determinado Servicio (y sólo uno), necesitaremos propagar a esta tabla como clave ajena los atributos que son clave primaria de la tabla SERVICIO (esto es, la clave y el nombre). Estos dos atributos, por lo tanto, serán clave ajena en la tabla DEPENDENCIA. Además, no podrán tomar valores nulos, ya que en los requisitos se nos ha exigido que toda dependencia está obligatoriamente bajo el control de un servicio.

Con respecto a la tabla SERVICIO, es necesario un pequeño comentario: los únicos atributos de esta tabla son los que forman parte de la clave primaria. Podríamos pensar en eliminarla, y únicamente guardar los servicios asociados a cada dependencia en la tabla DEPENDENCIA (que dejaría de ser clave ajena pero seguiría sin poder tomar valores nulos). Sin embargo, nos sería más complicado controlar la segunda restricción anterior (que todo servicio estuviera asociado a una dependencia):

- Mediante una tabla SERVICIO tendríamos suficiente con un disparador que obligara a que cada vez que se inserta un nuevo servicio se asociara a una dependencia, y que no se permitiera borrar una dependencia o modificar el servicio que la controla si era la única controlada por determinado servicio.

- Si almacenamos los servicios sólo en la tabla DEPENDENCIAS, cada vez que se borre o modifique una dependencia tendremos que controlar que todos los posibles valores que pudiera tomar un servicio aparecieran en alguna de las dependencias, de forma similar al caso anterior. Sin embargo, tendríamos bastantes más problemas si deseamos añadir un nuevo servicio, ya que tendríamos que modificar para ello la propia base de datos (para cambiar los dominios o para modificar el código que comprobara que todos los servicios controlan alguna dependencia).

SERVICIO (Clave, Nombre_S)

B:C,M:C

DEPENDENCIA (Código_D, **Nombre_D**, Función*, Localización*, *Clave_S*, *Nombre_S*)

Figura 2.1.42

Opciones de borrado y modificación

Tanto los borrados como las modificaciones de cualquier servicio se propagarían en cascada a la dependencia que controle.

"...Cada tripulante está asignado a una dependencia que se desea conocer, así como la cámara en la que se aloja. Una cámara es una dependencia que posee dos características propias, su categoría y su capacidad.

En este segundo párrafo nos encontramos con las siguientes restricciones a controlar: el hecho de que una cámara es un tipo especial de dependencia, y la asignación de cada tripulante tanto a la dependencia en que trabaja como a la cámara en las que se aloja. La existencia de este tipo especial de dependencias nos podría hacer plantearnos la posibilidad de crear suposiciones adicionales, como por ejemplo que las cámaras se

icen únicamente como alojamiento, y que por tanto la dependencia a la que esté gnado un tripulante no pueda ser una cámara. Esto podríamos resolverlo, suponiendo que existe una jerarquía en las dependencias, existiendo unas que son cámaras y as que no lo son, que podríamos reflejar de distintas formas (ver capítulo de delado E/R), por ejemplo, mediante tres relaciones (DEPENDENCIA, CÁMARA y _CÁMARA). A falta de más información, sin embargo, supondremos que la dendencia a la que esté asignado un tripulante pueda ser cualquiera, incluso la cámara la que él mismo se aloja.

La restricción de que una cámara es un tipo especial de dependencia la haremos ciendo que el código de la cámara sea también clave ajena de la dependencia

Para recoger la dependencia a la que está asignado un tripulante y la cámara en la ue se aloja, deberemos propagar a la relación TRIPULACIÓN la clave primaria de ada una de las relaciones DEPENDENCIA y CÁMARA, que serán claves ajenas que o podrán tomar valor nulo (ya que, como hemos dicho, todo tripulante trabaja en una sólo una dependencia y se aloja en una y sólo una cámara).

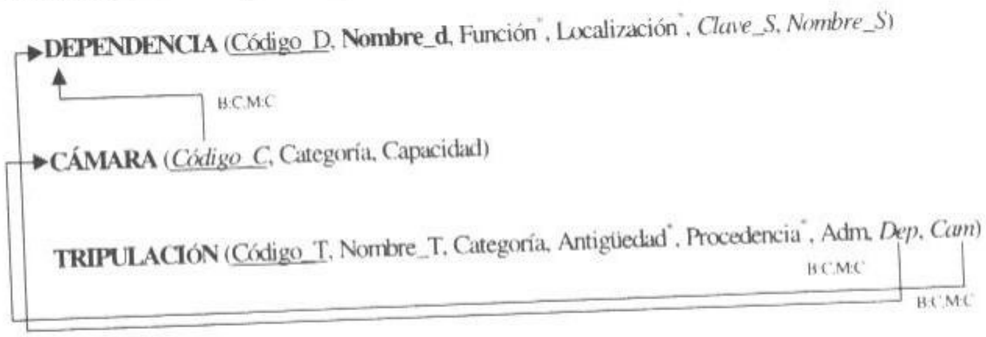


Figura 2.1.43

Opciones de borrado y modificación

Las opciones de borrado y modificación en el caso de "Código_C" de la relación CÁMARA serán claramente en cascada. En el caso de las claves ajenas de TRIPULACIÓN la modificación sería en cascada, y el borrado podría ser restringido o en cascada. En la solución propuesta se ha optado por este último.

Reglas ECA

Restricciones de rechazo

Una restricción adicional que no aparece en el enunciado pero que parece razonable es la de que no puedan estar alojados en una cámara más tripulantes de los que admite su capacidad. Esta restricción se recogería mediante una aserción.

> *"Por otra parte, se desean conocer los planetas que ha visitado cada miembro de la tripulación y el tiempo que permanecieron en ellos para saber las personas con quien se puede contar a la hora de realizar una exploración interplanetaria,...Algunos planetas se encuentran poblados por diversas razas, cada una en una cierta cantidad de individuos. De cada raza se almacena información sobre el nombre, población total y dimensiones medias (altura, anchura, peso).*

La últimas restricciones son las siguientes: se desea recoger tanto los planetas visitados por cada miembro de la tripulación como las razas que habitan en cada planeta. Para ambas necesitaremos crear una relación intermedia, ya que, por una parte, un miembro de la tripulación ha podido visitar más de un planeta, y cada planeta puede haberlo visitado más de un tripulante y, por otra, una raza puede habitar en distintos planetas y en cada planeta pueden convivir varias razas.

La relación VISITA recogerá las estancias de los tripulantes en los planetas. Formarán parte de su clave, por tanto, las claves ajenas de las relaciones TRIPULACIÓN y PLANETA, además de un atributo que contenga la duración de dicha visita. Además, si suponemos que un mismo tripulante ha podido visitar un planeta más de una vez, necesitaremos incrementar la clave para evitar que ésta se repita con cada nueva visita. Para ello añadiremos el atributo "Fecha_Visita" que recoja la fecha de la visita, y que formará también parte de la clave primaria. Esto implicará que, aunque un tripulante ha podido visitar un planeta más de una vez, estas visitas habrán tenido que producirse en distintas fechas (lo cual es una suposición razonable).

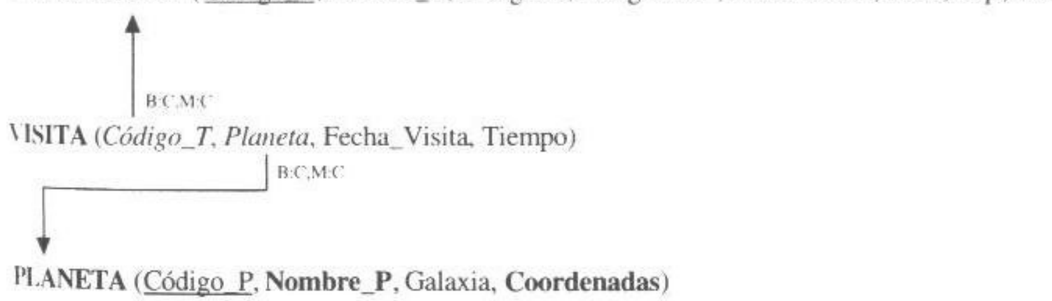


Figura 2.1 44

Opciones de borrado y modificación

Los borrados y modificaciones de las claves ajenas de la relación VISITA serán en cascada.

Para recoger las razas que habitan los planetas, crearemos la relación HABITA, que contendrá las razas que viven en cada planeta. Sus atributos serán las claves primarias de las relaciones RAZA y PLANETA (que formarán la clave primaria de la relación), junto a un atributo que contendrá el número de individuos de dicha raza que habitan en el planeta.

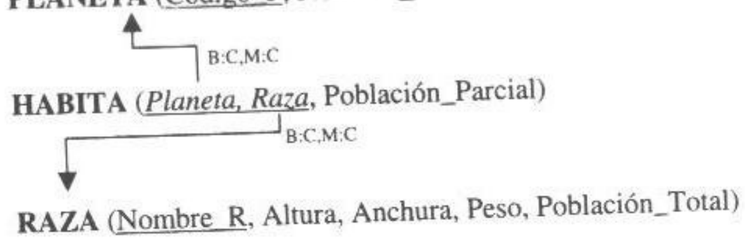


Figura 2.1.45

Opciones de borrado y modificación

Los borrados y modificaciones de las claves ajenas de las relación HABITA serán en cascada.

Reglas ECA

Si suponemos que toda raza habita en al menos un planeta deberemos controlarlo mediante disparadores.

Restricciones de rechazo

Será necesaria una restricción adicional que obligue a que la población total de una raza (contenida en la relación RAZA) coincida con la suma de las poblaciones parciales de la misma en todos los planetas en los que habita (contenidas en la relación HABITA), por ejemplo haciendo que el atributo POBLACIÓN_TOTAL sea un atributo calculado a partir de los valores del atributo POBLACIÓN_PARCIAL.

Grafo relacional

El grafo relacional completo se muestra a continuación.

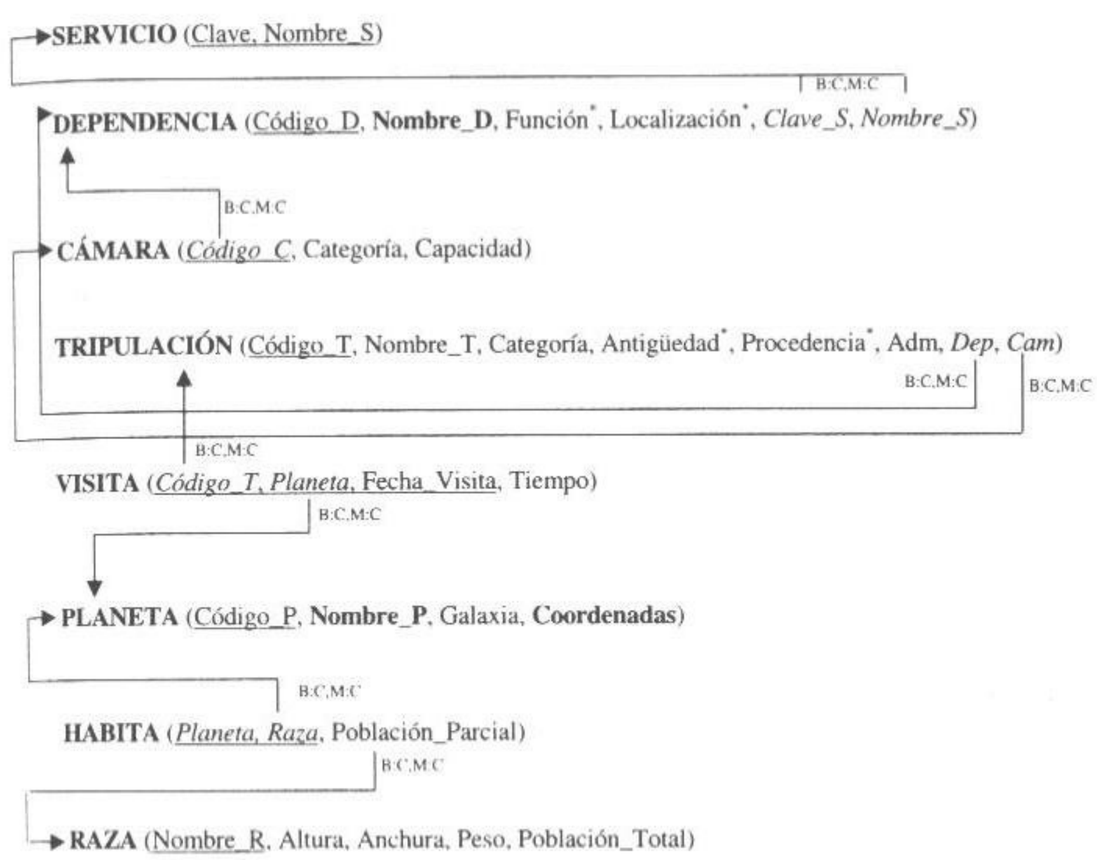


Figura 2.1.46

SCRIPT EN SQL-92

```
CREATE DOMAIN Nombre_Válido CHAR(40);
CREATE DOMAIN Tipo_Código CHAR(5);

CREATE TABLE Servicio (
      Clave Tipo_Codigo,
      Nombre_s Nombre_Válido,
PRIMARY KEY (Clave, Nombre_s)
);

CREATE TABLE Dependencia (
      Codigo_c Tipo_Código,
      Nombre_d Nombre_Válido NOT NULL,
      Función CHAR(20),
      Localización CHAR(20),
      Clave_s  Tipo_Código NOT NULL,
      Nombre_s Nombre_Válido NOT NULL,
PRIMARY KEY (Código_c),
```

```
NIQUE (Nombre_s),
OREIGN KEY (Clave_s, Nombre_s) REFERENCES Servicio
      ON UPDATE CASCADE
      ON DELETE CASCADE


REATE TABLE Camara (
      Código_c Tipo_Código,
      Categoría Nombre_Válido NOT NULL,
      Capacidad INTEGER NOT NULL,
RIMARY KEY (Codigo_c),
OREIGN KEY (Codigo_c) REFERENCES Dependencia
      ON UPDATE CASCADE
      ON DELETE CASCADE
;

CREATE TABLE Tripulación (
      Código_T Tipo_Código,
      Nombre_T Nombre_Válido,
      Categoría CHAR(20) NOT NULL,
      Antigüedad CHAR(20),
      Procedencia CHAR(20),
      Adm CHAR(20)  NOT NULL,
      Dep Tipo_Código NOT NULL,
      Cam Tipo_Código NOT NULL,
PRIMARY KEY (Código_T),
FOREIGN KEY (Dep) REFERENCES Dependencia
      ON UPDATE CASCADE
      ON DELETE CASCADE,
FOREIGN KEY (Cam) REFERENCES Camara
      ON UPDATE CASCADE
      ON DELETE CASCADE
);

CREATE TABLE Planeta (
      Código_p Tipo_Código,
      Nombre_p Nombre_Válido NOT NULL,
      Galaxia CHAR(15) NOT NULL,
      Coordenadas CHAR(15) NOT NULL,
PRIMARY KEY (Código_p),
UNIQUE (Nombre_p),
UNIQUE (Coordenadas)
);

CREATE TABLE Visita (
      Código_t Tipo_Código,
      Planeta Tipo_Código,
```



```
      Fecha_visita DATE,
      Tiempo INTEGER NOT NULL,
PRIMARY KEY (Código_t, Planeta, Fecha_visita),
FOREIGN KEY (Código_t) REFERENCES Tripulación
      ON UPDATE CASCADE
      ON DELETE CASCADE,
FOREIGN KEY (Planeta) REFERENCES Planeta
      ON UPDATE CASCADE
      ON DELETE CASCADE
);


CREATE TABLE Raza (
      Nombre_r Nombre_Válido,
      Altura NUMBER NOT NULL,
      Anchura  NUMBER NOT NULL,
      Peso NUMBER NOT NULL,
      Población_total  NUMBER NOT NULL,
PRIMARY KEY (Nombre_r)
);


CREATE TABLE Habita (
      Planeta Tipo_Código,
      Raza Nombre_Válido,
      Población_Parcial NUMBER NOT NULL,
PRIMARY KEY (Planeta, Raza),
FOREIGN KEY (Planeta) REFERENCES Planeta
      ON UPDATE CASCADE
      ON DELETE CASCADE,
FOREIGN KEY (Raza) REFERENCES Raza
      ON UPDATE CASCADE
      ON DELETE CASCADE
);


CREATE ASSERTION Capacidad_Cámaras
CHECK (NOT EXISTS
      SELECT Código_c
      FROM Cámara
      WHERE Capacidad <
       (SELECT COUNT (Código_t)
        FROM Tripulación
        GROUP BY Cam
        HAVING Cam=Código_c));
```

CONSULTAS

1. Razas que habitan en los planetas visitados por Flash Gordon

Sql

```
SELECT Raza
FROM Habita,Visita, Tripulación
WHERE Tripulación.Nombre_t='Flash Gordon'
        AND Tripulación.Código_t=Visita.Código_t
        AND Visita.Planeta=Habita.Planeta;
```

Álgebra relacional

$$\Pi_{Raza}((\Pi_{Planeta,Raza}(HABITA)) * (\Pi_{Codigo_t,Planeta}(VISITA)) * (\Pi_{Codigo_t} \sigma_{Nombre_t='Flash\ Gordon'}(TRIPULACION)))$$

2. Cámaras que han llegado a su capacidad máxima.

Sql

```
SELECT Código_c
FROM Cámara
WHERE Capacidad =
        (SELECT COUNT (Código_t)
        FROM Tripulación
        GROUP BY Cam
        HAVING Cam=Codigo_c));
```

Álgebra relacional

$$\Pi_{Código_c}(\Pi_{Código_c,Capacidad}(CÁMARA) \cap (TRIPULACIÓN\ AGRUPACIÓN_POR\ Cam, FRECUENCIA\ (Código_t)))$$

3. Nombre de los tripulantes asignados a alguna dependencia de seguridad.

Sql

```
SELECT Nombre_t
FROM Tripulación, Dependencia
WHERE Tripulación.Dep=Dependencia.Código_D
        AND Dependencia.Nombre_S[27]='Seguridad';
```

[27] Suponemos que en el campo Nombre_s se almacena el tipo de Dependencia

Álgebra relacional

$$\Pi_{Nombre_t}((\Pi_{Dep,\ Nombre_t}(TRIPULACIÓN)) * _{Dep=Codigo_d} (\Pi_{Codigo_d} \sigma_{Nombre_s='Seguridad'}(DEPENDENCIA)))$$

PROBLEMA 9: EMPRESA DE TRABAJO TEMPORAL

Enunciado

La empresa de trabajo temporal MITO ha decidido crear un sistema de información para realizar su gestión, considerando las siguientes especificaciones:

La bolsa de trabajo de MITO almacena toda la información relativa a los demandantes de empleo, considerando como relevante el DNI, nombre, dirección, teléfono, clase de carnet de conducir y currículum. Estando este último compuesto por número de titulaciones (de 1 a 4 en media 2), nombre de las mismas, especialidad realizada en cada una de ellas y universidad donde se tituló.

Si el demandante posee experiencia profesional también guardaremos el nombre de la empresa donde realizó el trabajo, el tiempo y el puesto que desempeñó. En general existe un promedio de dos trabajos por persona.

Las empresas cliente se caracterizan por un código de cliente, su CIF, el nombre, la dirección, el teléfono y el número de fax.

Los clientes solicitan al menos un perfil. Un perfil se identifica por un código de perfil y lleva asociada una descripción. Además un perfil puede ser requerido por varios clientes.

Un perfil puede asociarse a distintos demandantes de empleo y a su vez un demandante puede responder a diferentes perfiles.

- Diseñe el esquema relacional. Indique en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas, especifique los modos de borrado y modificación. Marque con un asterisco (*) los campos que acepten valores nulos. Y por último, indique si existe algún atributo que exija definir alguna restricción a fin de mantener la integridad de la base de datos.
- Efectuar además en álgebra relacional y en SQL las siguientes consultas:
  - Nombre de las empresas que piden el perfil de "especialista en circuitos impresos".

- Nombre y apellidos de los demandantes de empleo que se ajustan al perfil de "administrador de Bases de Datos Oracle", así como los datos de la o las empresas que lo solicitan.
- Nombre, apellido y currículum de aquellos demandantes de empleo que hayan trabajado al menos en dos puestos de trabajo relacionados con la programación en bases de datos.

Discusión del enunciado

IDENTIFICACIÓN DE LAS RELACIONES BÁSICAS

"La bolsa de trabajo de MITO almacena toda la información relativa a los demandantes de empleo, considerando como relevante el DNI, nombre, dirección, teléfono, clase de carnet de conducir y currículum. Estando este último compuesto por número de titulaciones (de 1 a 4 en media 2), nombre de las mismas, especialidad realizada en cada una de ellas y universidad donde se tituló"

De este supuesto podemos deducir dos relaciones, la primera serviría para mantener los datos de cada uno de los demandantes de empleo:

DEMANDANTE (DNI, Nombre_S, Dirección_S, Carnet)

Tendría como clave primaria el atributo "DNI", aunque se podría considerar al atributo "Nombre_S" como clave alternativa.

La segunda relación sería:

CURRÍCULUM (Número_Tit, Nombre_Tit, Especialidad)

Esta relación no la podríamos haber considerado como atributo de DEMANDANTE, pues admitiría varios valores para cada tupla de la relación (por ejemplo, por cada demandante habrá varios currículums), lo cual no está admitido en el modelo relacional.

La clave primaria de la relación CURRÍCULUM es "Número_Tit", siendo éste un código que no representa el número de titulaciones que posee un demandante[28] sino que será un valor, en principio arbitrario, que las diferencie y para el cual no se admitan duplicados.

[28] Se podría conseguir el número de titulaciones de un demandante de empleo mediante una operación del Lenguaje de Manipulación de Datos (LMD), simplemente contando el número de tuplas que hay en esta relación correspondientes a ese demandante.

Podemos considerar que el atributo "Especialidad" puede admitir valores nulos, pues una determinada titulación puede no tener ninguna especialidad.

El hecho de que las titulaciones varíen de 1 a 4, en media 2, no es relevante para el diseño lógico de nuestro esquema, aunque sí pudiera serlo para el diseño físico.

"Si el demandante posee experiencia profesional también guardaremos el nombre de la empresa donde realizó el trabajo, el tiempo y el puesto que desempeñó. En general existe un promedio de dos trabajos por persona".

De este supuesto se deduce que debemos crear una nueva relación que almacene los trabajos anteriores de cada demandante de empleo, así tenemos:

EXPERIENCIA_PROF (Código_Exp, Nombre_E, Tiempo, Puesto)

Donde "Código_Exp" es la clave primaria. Se debe suponer que en esta relación aparecerá una tupla por cada puesto ocupado, no por cada empresa en la que se ha trabajado, estando por tanto su granularidad fijada al nivel de puesto de trabajo, no de empresa (en caso contrario, un demandante de empleo no podría haber cambiado de puesto dentro de una empresa, salvo que hubiera alguna manera de evitarlo por ejemplo que se nos dijera que bastaría con registrar el puesto máximo alcanzado dentro de la empresa). Como consecuencia, esta relación no se encontrará normalizada, por ello introduciríamos una nueva relación:

EMPRESA(Código_E, Nombre_E)

Y modificaríamos ligeramente la anterior relación de acuerdo con esta hipótesis

El promedio de trabajos anteriores por persona no es relevante en el diseño.

"Las empresas cliente se caracterizan por un código de cliente, su CIF, el nombre, la dirección, el teléfono y el número de fax".

Del análisis de esta frase se deduce una nueva relación:

CLIENTE (Código_Cli, CIF, Nombre_Cli, Dirección_Cli, Teléfono_Cli, Fax)

Donde "Código_Cli" es la clave primaria y "CIF" es la clave alternativa.

"Los clientes solicitan al menos un perfil. Un perfil se identifica por un código de perfil y lleva asociada una descripción."

Se deben almacenar los diferentes perfiles que puedan ser, tanto pedidos por las empresas cliente, como ofertados por los diferentes demandantes de empleo, por tanto tenemos una nueva relación:

PERFIL (Código_P, Descripción)

Donde "Código_P" es la clave primaria.

ANÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

> *"La bolsa de trabajo de MITO almacena toda la información relativa a los demandantes de empleo, considerando como relevante el DNI, nombre, dirección, teléfono, carnet de conducir y currículum. Estando este último compuesto por número de titulaciones (de 1 a 4 en media 2), nombre de las mismas, especialidad realizada en cada una de ellas y universidad donde se tituló".*

Cada demandante de empleo tiene un currículum compuesto por una o varias titulaciones. Además una misma titulación puede corresponder a varios demandantes de empleo, por tanto debemos incluir una nueva relación CURRÍCULUM_DEMANDANTE, que contenga, para cada demandante de empleo, los códigos de sus correspondientes titulaciones. De esta forma, la clave primaria estará formada por los atributos "DNI" y "Número_Tit", cada uno de los cuales es clave ajena que referencia a DEMANDANTE y a CURRÍCULUM respectivamente.

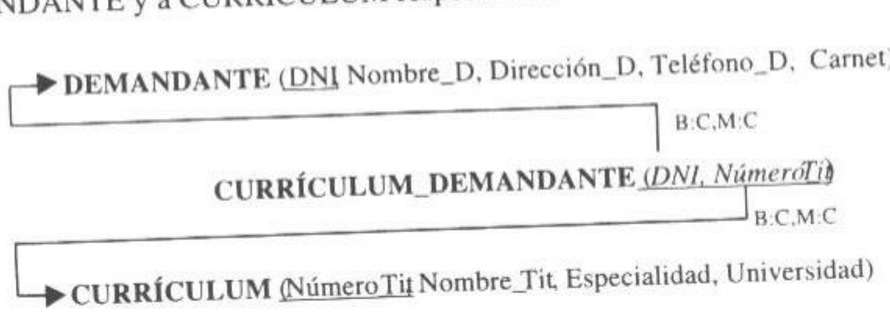


Figura 2.1.47

Opciones de borrado y modificación

El borrado y la modificación de ambas claves ajenas puede hacerse en cascada. pues el borrado o la modificación de las tuplas de la tabla referenciada no implica la pérdida de información en la tabla que referencia (B:C,M:C). Dependiendo de lo restrictivo que se pretenda ser, el borrado podría también ser restringido.

Reglas ECA

Deberíamos controlar mediante disparadores, que todo demandante tiene al menos un currículum, así, por ejemplo, no se podría borrar un currículum de un demandante si éste es el único que tiene registrado..

> *"Si el demandante posee experiencia profesional también guardaremos el nombre de la empresa donde realizó el trabajo, el tiempo y el puesto que desempeñó. En general existe un promedio de dos trabajos por persona".*

Un demandante de empleo puede poseer experiencia profesional (varias), pero una misma experiencia profesional corresponde a un único demandante, por otra parte, tendremos que relacionar la relación EMPRESA con EXPERIENCIA_PROF, por tanto tenemos el subesquema siguiente:

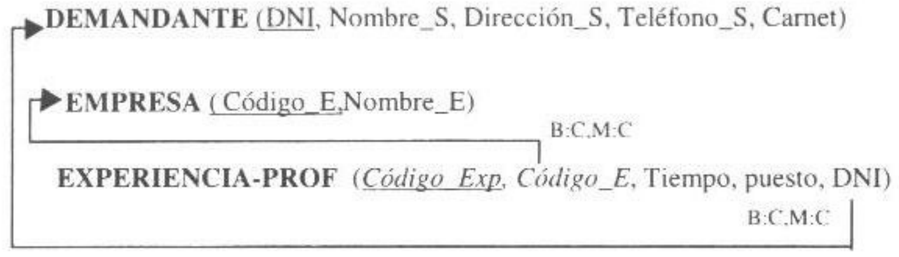


Figura 2.1.48

Donde "DNI" es clave ajena de la relación EXPERIENCIA_PROF, que referencia a DEMANDANTE y "Código_E" es clave ajena de la relación EXPERIENCIA_PROF que referencia a EMPRESA.

Opciones de borrado y modificación

Si eliminamos o modificamos un demandante, su experiencia profesional debe ser eliminada o modificada, por lo tanto en ambos casos se debe hacer en cascada (B:C,M:C). Otra alternativa que proporcionaría mayor seguridad se lograría considerando el borrado restringido.

De manera similar podría considerarse las opciones para la clave ajena "Código_E".

> *"Los clientes solicitan al menos un perfil. Un perfil se identifica por un código de perfil y lleva asociada una descripción. Además un perfil puede ser requerido por varios clientes".*

Las empresas cliente solicitan al menos un perfil, y un mismo perfil puede ser requerido por varias empresas, por tanto tenemos que introducir una nueva relación CLIENTE_PERFIL, cuya clave primaria estará formada por las claves primarias de las relaciones CLIENTE y PERFIL, es decir "Código_Cli" y "Código_P", cada una de las cuales es a su vez clave ajena que referencia a las relaciones de las cuales son clave primaria.

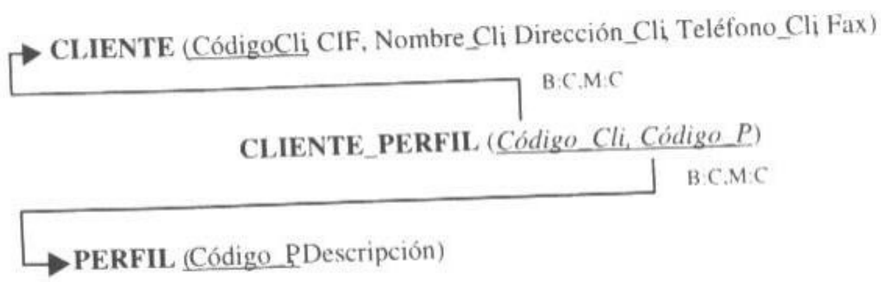


Figura 2.1.49

Opciones de borrado y modificación

Como en los casos anteriores y por las mismas razones el borrado y la modificación de las claves ajenas, lo hemos supuesto en cascada (B:C,M:C), aunque podría ser restringido.

Reglas ECA

Como cada empresa cliente solicita al menos un perfil, debería asegurarse que siempre se registra el perfil al tiempo que se da de alta al cliente y que nunca habrá un cliente sin el (los) correspondiente(s) perfil(es).

> *"Un perfil puede asociarse a distintos demandantes de empleo y a su vez un demandante puede responder a diferentes perfiles".*

Del análisis de esta frase se deduce, como en el caso anterior, la necesidad de introducir una nueva relación, obteniendo el siguiente subesquema:

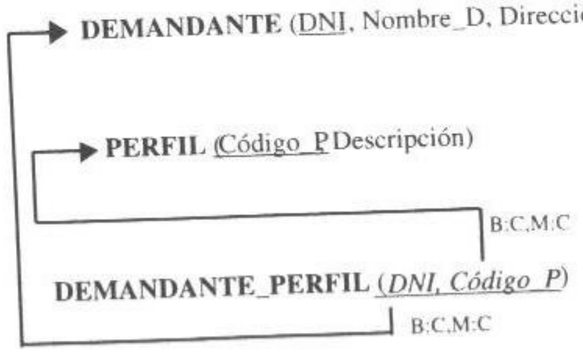


Figura 2.1.50

Opciones de borrado y modificación

Análogamente a los casos anteriores, el borrado y la modificación de las claves ajenas puede hacerse en cascada.

GRAFO RELACIONAL

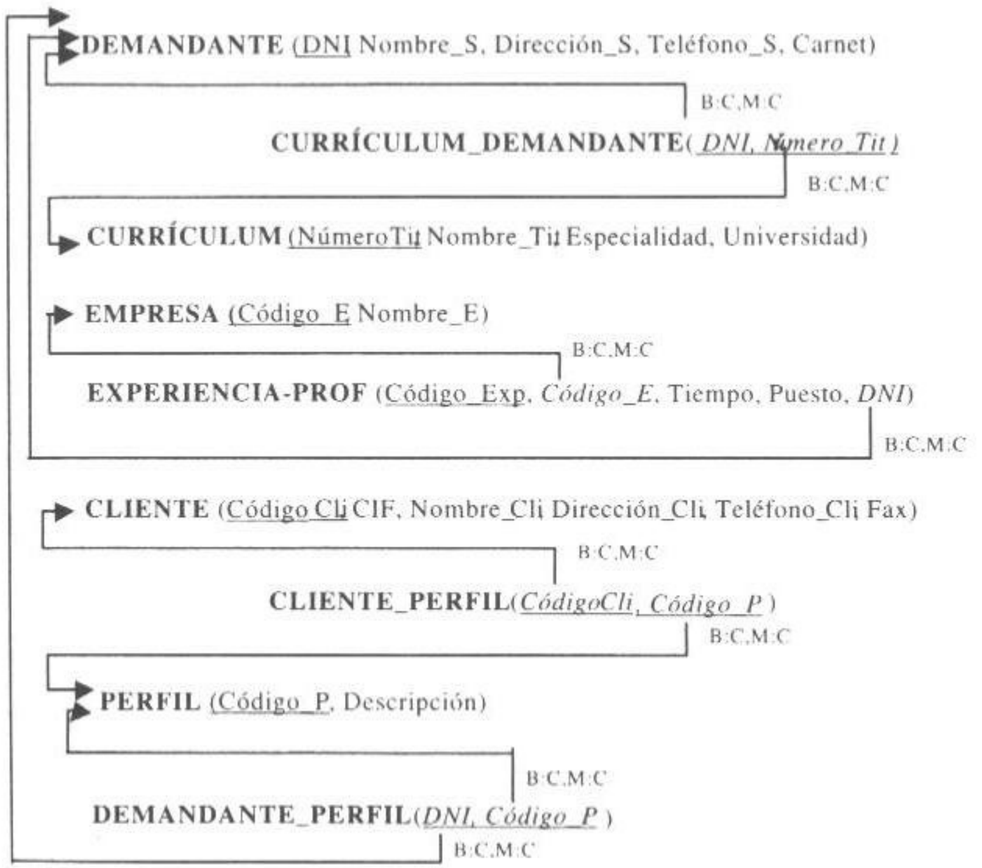


Figura 2.1.51

SCRIPT EN SQL_92

```
CREATE DOMAIN Nombre_Válido CHAR(40);
CREATE DOMAIN Tipo_Código CHAR(5);

CREATE TABLE Demandante (
      DNI CHAR(9),
      Nombre_d Nombre_Válido
      Dirección_d CHAR(20) NOT NULL,
      Teléfono_d CHAR(9) NOT NULL,
      Carnet CHAR(2) NOT NULL,
PRIMARY KEY (DNI)
);

CREATE TABLE Currículum (
      Número_Tit Tipo_Código,
      Nombre_Tit CHAR(15) NOT NULL,
```

```
    Especialidad CHAR(30) NOT NULL,
    Universidad CHAR(50) NOT NULL,
RIMARY KEY (Número_Tit)


REATE TABLE Cliente (
    Código_Cli Tipo_Código,
    Nombre_Cli Nombre_Válido
    Dirección_Cli CHAR(60) NOT NULL,
    Teléfono_Cli CHAR(9) NOT NULL,
    Fax CHAR(9) NOT NULL,
RIMARY KEY (Código_Cli)


REATE TABLE Perfil (
    Código_P Tipo_Código,
    Descripción CHAR(70) NOT NULL,
RIMARY KEY (Código_P)


REATE TABLE Currículum_Demandante (
    DNI CHAR(9),
    Número_Tit Tipo_Código,
RIMARY KEY (DNI, Número_Tit),
OREIGN KEY (DNI) REFERENCES Demandante
    ON UPDATE CASCADE
    ON DELETE CASCADE,
'OREIGN KEY (Número_Tit) REFERENCES Currículum
    ON UPDATE CASCADE
    ON DELETE CASCADE
;

CREATE TABLE Cliente_Perfil (
    Código_Cli Tipo_Código,
    Código_P Tipo_Código,
PRIMARY KEY (Código_Cli, Código_P),
FOREIGN KEY (Código_Cli) REFERENCES Cliente
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Código_P) REFERENCES Perfil
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Demandante_Perfil (
    DNI CHAR(9),
    Código_P Tipo_Código,
```



```
PRIMARY KEY (DNI, Código_P),
FOREIGN KEY (DNI) REFERENCES Demandante
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Código_P) REFERENCES Perfil
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Empresa (
    Código_e Tipo_Código,
    Nombre_E Nombre_Válido,
PRIMARY KEY (Código_Exp)
);

CREATE TABLE Experiencia_Prof (
    Código_Exp Tipo_Código,
    Código_e Nombre_Válido,
    Tiempo CHAR(15) NOT NULL,
    Puesto  CHAR(40) NOT NULL,
    DNI CHAR(9) NOT NULL,
PRIMARY KEY (Código_Exp),
FOREIGN KEY (DNI) REFERENCES Demandante
    ON UPDATE CASCADE
    ON DELETE CASCADE
FOREIGN KEY (Código_e) REFERENCES Empresa
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
```

CONSULTAS

1. Nombre de las empresas que piden el perfil "Especialista en circuitos impresos".

Sql

```
SELECT Nombre_Cli
FROM Cliente, Cliente_perfil, Perfil
WHERE Cliente.Código_Cli=Cliente_perfil.Código_Cli AND
        Cliente_perfil.Código_P=Perfil.Código_P AND Perfil.Descripción =
        'Especialista en circuitos impresos';
```

Álgebra relacional:

$$\Pi_{Nmobre_cli}(\Pi_{Código_Cli,\ Nombre_Cli}(CLIENTE)*(CLIENTE_PERFIL)*\Pi_{Código_P}(\sigma_{Descripción\ =\ 'Especialista\ en\ circuitos\ impresos'}(PERFIL)))$$

2. Nombre y apellidos de los demandantes de empleo que se ajustan al perfil "Administrador de bases de datos ORACLE", así como los datos de las empresas que lo solicitan.

Sql

```
SELECT Nombre_S, CIF, Nombre_Cli, Dirección_Cli, Teléfono_Cli, Fax
FROM Demandante A, Demandante_Perfil B, Perfil C
WHERE A.DNI=B.DNI AND B.Código_P=C.Código_P AND C.Descripción =
        'Administrador de bases de datos ORACLE'
        AND C.Código_P IN (SELECT Código_P
                           FROM Cliente_Perfil, Cliente
                       WHERE Cliente.Código_Cli =
                         Cliente_perfil.Código_Cli);
```

Álgebra relacional:

$$\Pi_{\text{Nombre_S, CIF, Nombre_Cli, Dirección_Cli, Teléfono_Cli, Fax}}(\text{DEMANDANTE}) * ((\text{DEMANDANTE_PERFIL}) * (\Pi_{\text{Código_P, CIF, Nombre_Cli, Dirección_Cli, Teléfono_Cli, Fax}}((\text{CLIENTE})*(\text{CLIENTE_PERFIL}))*\Pi_{\text{Código_P}}(\sigma_{\text{Descripción = 'Administrador de bases de datos ORACLE'}}(\text{PERFIL}))))$$

3. Nombre y apellidos de aquellos demandantes de empleo que hayan trabajado en, al menos, dos puestos de trabajo relacionados con la programación en bases de datos.

Sql

```
SELECT Nombre_S,
FROM Demandante
WHERE Demandante.DNI IN (SELECT DISTINCT DNI
                         FROM Experiencia_Prof
                         WHERE Puesto='%bases de datos%');
```

Álgebra relacional

$$\Pi_{\text{Nombre_S}}(\Pi_{\text{DNI,Nombre_S}}(\text{DEMANDANTE}) * \Pi_{\text{DNI}}(\sigma_{\text{Puesto='\%bases de datos\%'}}(\text{EXPERIENCIA_PROF})))$$

PROBLEMA 10: FONDOS BIBLIOGRÁFICOS

Enunciado

La Biblioteca de la Universidad Carlos III desea una Base de Datos para gestionar sus fondos bibliográficos. Para ello expone las siguientes especificaciones, que han de tenerse en cuenta para su diseño.

De cada fondo se desea guardar el formato o tipo (libro, revista, audiovisual, etc.), el número de ejemplares disponibles, el título, el autor, la editorial, la fecha de edición y la clave de localización, a efectos de poder saber de forma rápida su ubicación en la biblioteca.

Como cada fondo posee una serie de ejemplares que tienen la misma clave de localización que la del fondo del que proceden, se tendrá un número correlativo para poder identificarlos. Además, se quiere tener la información de si se puede prestar o no y, en caso de que esté prestado, de cuál es la fecha de devolución.

Los usuarios que pueden acceder a los fondos de la Biblioteca son: profesores o alumnos de la Universidad Carlos III. De ellos se necesita saber el número de socio, el DNI, el nombre, la dirección de correo electrónico (si poseen), y si se trata de un profesor, el departamento y el área a la que pertenecen.

Existen fondos en la biblioteca que son comprados por distintas áreas de la Universidad y se necesita conocer esta información, ya que los profesores que solicitan el préstamo de un libro que ha comprado su área tendrá un trato especial. Esto significa que en general la filosofía del préstamo es que tanto para alumnos como para profesores, un fondo sólo puede ser prestado durante una semana. Sin embargo, si el profesor pertenece al área que ha proporcionado dicho ejemplar a la biblioteca podrá tenerlo durante un mes.

De los departamentos se guardará su nombre, titulación, y director de éste. Cada departamento está formado por una o varias áreas que se identifican por un nombre, se encuentran en un laboratorio y poseen una línea de investigación principal que también se desea almacenar.

- Diseñe el esquema relacional. Indique en él las claves primarias, claves alternativas y las claves ajenas. De estas últimas, especifique los modos de borrado y modificación. Marque con un asterisco (*) los campos que acepten valores nulos. Y, por último, indique si existe algún atributo que exija definir alguna restricción a fin de mantener la integridad de la base de datos.
- Efectuar además en álgebra relacional y en SQL las siguientes consultas:
  - Número de libros prestados a alumnos de la Universidad Carlos III en este último año.
  - Profesores del área de Bases de Datos que no hayan utilizado el servicio de biblioteca.
  - Título de los ejemplares que no han sido devueltos junto con el nombre del usuario que los posee.

scusión del enunciado

ENTIFICACIÓN DE LAS RELACIONES BÁSICAS

> *"De cada fondo se desea guardar el formato o tipo (libro, revista, audiovisual, etc.), el número de ejemplares disponibles, el título, el autor, la editorial, la fecha de edición y la clave de localización, a efectos de poder saber de forma rápida su ubicación en la biblioteca".*

Para recoger toda la información necesaria acerca de un fondo de la biblioteca esitaremos crear una relación FONDO compuesta por las propiedades que se ican en este párrafo. De entre todas ellas elegiremos como clave primaria el ibuto "C_Localización", ya que supondremos que todo fondo bibliográfico se ubica un lugar único dentro de la biblioteca. Los atributos obligatorios serán "Formato", _Ejemplares", "Título", "Editorial" y "F_Edición". El atributo "Autor" se puede odelar de distintas formas, ya que si hacemos uso del conocimiento del dominio, emos que un libro puede ser anónimo, lo que supondría que el atributo admitiría lores nulos, pero además, un libro puede ser escrito por más de un autor, lo que adría que ser modelado con otra relación. Para poder reflejar estos supuestos, y de a manera aproximar nuestro diseño al mundo real, crearemos otra relación AUTOR, n un atributo "Cód_autor" como clave primaria y "nombre" atributo obligatorio yo valor será el nombre completo del autor.

FONDO (C_Localización, Título, Editorial, F_Edición, Formato, N_Ejemplares)

AUTOR (Cód_Autor, Nombre)

> *"Como cada fondo posee una serie de ejemplares que tienen la misma clave de localización que la del fondo del que proceden, se tendrá un número correlativo para poder identificarlos. Además, se quiere tener la información de si es prestable o no y, en caso de que está prestado, de cual es la fecha de devolución."*

Los fondos bibliográficos poseen distintos ejemplares que son realmente los que prestan a los usuarios de la biblioteca, por lo que necesitaremos crear una relación JEMPLAR para saber si están disponibles o no cada uno de ellos. Como se indica en enunciado la clave primaria de la relación será la concatenación de dos atributos, C_Localización" y "Número", donde este último ha de ser un número correlativo que n el esquema relacional no podremos representar. La relación EJEMPLAR también stará compuesta por los atributos "Prestable", definido dentro del dominio booleano, "F_Devolución", que podrá tomar valores nulos para considerar aquellos fondos que o estén prestados.

EJEMPLAR (C_Localización, Número, Prestable, F_Devolución*)

> *"Los usuarios que pueden acceder a los fondos de la Biblioteca son: profesores o alumnos de la Universidad Carlos III. De ellos se necesita saber el número de socio, el DNI, el nombre, la dirección de correo electrónico (si poseen) y si se trata de un profesor, el departamento y el área a la que pertenecen."*

Entendemos, por tanto, que los usuarios de la biblioteca son de dos tipos, profesor o alumno, y que éstos comparten ciertas propiedades. Por lo que se creará una relación USUARIO cuya clave primaria será el número de socio, "N_Socio", el DNI ("DNI") será clave alternativa, el nombre ("Nombre") y el tipo "Tipo" serán atributos obligatorios y la dirección de correo ("e-mail") un atributo que admite valores nulos, ya que puede haber socios que no posean, incluso se podría considerar clave alternativa, ya que cada usuario, si tiene dirección electrónica, es única.

USUARIO (N_Socio, **DNI**, Nombre, Tipo, **e-mail***)

El problema que nos encontramos es cómo reflejar aquellas propiedades que sólo poseen el tipo de usuario profesor ("Departamento" y "Área"). La solución más sencilla sería incluir los atributos "Departamento" y "Área" en la relación USUARIO, por lo que las tuplas correspondientes a socios del tipo alumno, tendrían valores nulos en dichos atributos. Para evitar tener muchos valores nulos en esta relación se podría considerar otra solución, que es la que adoptaremos creando una nueva relación PROFESOR cuya clave primaria sería "N_Socio" y como atributos obligatorios "Departamento" y "Área".

PROFESOR (N_Socio, Departamento, Área)

> *"Existen fondos en la biblioteca que son comprados por distintas áreas de la Universidad y se necesita conocer esta información, ya que los profesores que solicitan el préstamo de un libro que ha comprado su área tendrá un trato especial. Esto significa que, en general, la filosofía del préstamo es que tanto para alumnos como para profesores, un fondo sólo puede ser prestado como máximo por una semana. Sin embargo, si el profesor pertenece al área que ha proporcionado dicho ejemplar a la biblioteca podrá tenerlo durante un mes."*

> *"De los departamentos se guardará su nombre, titulación, y director de éste. Cada departamento está formado por una o varias áreas que se identifican por un nombre, se encuentran en un laboratorio y poseen una línea de investigación principal que también se desea almacenar."*

Aparecen en este párrafo las relaciones DEPARTAMENTO y ÁREA con los atributos que se quieren almacenar de cada una de ellas. Para la relación primera se ha elegido como clave primaria "Nombre", suponiendo que cada departamento tiene un nombre único que le identifica del resto. La segunda relación tendrá el atributo "Nombre" como clave primaria y suponemos lo mismo que para la relación anterior.

DEPARTAMENTO (Nombre, Titulación, Director)

ÁREA (Nombre, Nº_Laboratorio, Investigación)

Donde se ha supuesto que un departamento está inscrito a una sola Titulación y e el laboratorio puede ser compartido por distintas áreas, por lo tanto el atributo °_Laboratorio" será un atributo obligatorio pero no clave alternativa.

NÁLISIS DE LOS SUPUESTOS SEMÁNTICOS

"De cada fondo se desea guardar el formato o tipo (libro, revista, audiovisual, etc.), el número de ejemplares disponibles, el título, el autor, la editorial, la fecha de edición y la clave de localización, a efectos de poder saber de forma rápida su ubicación en la biblioteca".

Como se ha comentado anteriormente, se supone que un fondo bibliográfico ede haber sido escrito por más de un autor e incluso ser anónimo, por lo que para ntemplar este supuesto la relación FONDO se debe ampliar con un nuevo atributo, utor", que será clave ajena que referencie a la relación AUTOR y que podrá tomar lores nulos.

FONDO (C_Localización , Título, Editorial, F_Edición, Formato, N_Ejemplares, *Autor**)

B:R,M:C

AUTOR (Cód_Autor , Nombre)

Figura 2.1.52

pciones de borrado y modificación

No sería conveniente que al borrar un autor de la relación AUTOR se borraran dos los fondos donde éste aparece, ni que se dejara un valor nulo ni por defecto en la elación FONDO. Por lo que se tendrá un borrado restringido y la modificación en ascada.

eglas ECA

Si se supone que en la relación AUTOR sólo pueden aparecer autores que se orrespondan con uno o varios fondos de la biblioteca, tendríamos que controlar ediante disparadores el que cada vez que se incorporara un nuevo autor, éste osteriormente se relacionara con algún fondo.

"Como cada fondo posee una serie de ejemplares que tienen la misma clave de localización que la del fondo del que proceden, se tendrá un número correlativo para poder identificarlos. Además, se quiere tener la información de si es prestable o no y, en caso de que esté prestado, de cuál es la fecha de devolución."

Existe una clara relación entre FONDO y EJEMPLAR, de donde suponemos que todo fondo tiene al menos un ejemplar y donde todo ejemplar procede de un fondo, por lo que bastaría con poner como clave ajena al atributo "C_Localización" que además forma parte de la clave primaria y hacer que referencie a la relación FONDO.

FONDO (C_Localización , Título, Editorial, F_Edición, Formato, N_Ejemplares, *Autor**)

B:C,M:C

EJEMPLAR (*C_Localización*, Número, Prestable, F_Devolución*)

Figura 2.1.53

Opciones de borrado y modificación

Al formar parte la clave ajena de la clave primaria de la relación EJEMPLAR sólo tenemos dos opciones tanto para el borrado como para la modificación y éstas son restringido o en cascada. Se tomará el borrado en cascada, ya que si eliminamos un fondo no tendría sentido mantener los ejemplares. La modificación también será en cascada.

Restricción de rechazo

Una restricción que se debe tener en cuenta es que cuando el atributo "Prestable" toma el valor TRUE el valor del atributo "F_Devolución" tendrá que ser nulo. Esta restricción se podría realizar con un CHECK o una regla de verificación.

Reglas ECA

Con la clave ajena "C_Localización" controlamos que todo ejemplar proviene de uno y sólo un fondo. Pero si quisiéramos que todo fondo tuviera un ejemplar como mínimo en la biblioteca tendríamos que implementar un disparador para controlar dicha restricción.

El atributo que recoge el número de ejemplares que tiene un determinado fondo se calculará mediante un disparador. Donde cada vez que se añada una tupla en la relación EJEMPLAR tendrá que aumentar en uno "N_Ejemplares" para el fondo correspondiente. Se disminuirá en uno si estamos borrando un ejemplar y si se realiza una modificación en el atributo "C_Localización" de EJEMPLAR tendremos que

mentar en uno "N_Ejemplares" en la tupla correspondiente al nuevo fondo y dis-inuir en uno en la tupla del antiguo fondo.

> *"Los usuarios que pueden acceder a los fondos de la Biblioteca son: profesores o alumnos de la Universidad Carlos III. De ellos se necesita saber el número de socio, el DNI, el nombre, la dirección de correo electrónico (si poseen) y si se trata de un profesor, el departamento y el área a la que pertenecen."*

Al crear una relación USUARIO y otra relación PROFESOR para recoger los tributos propios de éste, existe una asociación entre ambas, ya que la clave primaria e profesor es un número de socio cuyo valor debe aparecer en la relación USUARIO.

Figura 2.1.54

Opciones de borrado y modificación

Como un profesor es un usuario, si borramos o modificamos alguna tupla en la relación USUARIO que se corresponda con el tipo profesor, esta acción se debería propagar en cascada a la relación PROFESOR.

Reglas ECA

En primer lugar se tendría que implementar un disparador para cuando se inserte una tupla del tipo profesor en la relación USUARIO se inserte también en la relación PROFESOR, con esta restricción se cumplirá que todo usuario profesor se encuentra en la relación PROFESOR.

En segundo lugar la clave ajena de PROFESOR, por la propia definición de ésta, tomará valores de un usuario, pero no nos vale con sólo la definición, sino que además este valor debe corresponderse con una tupla cuyo atributo "Tipo" sea profesor.

También se indica que necesitamos saber el área y el departamento al que pertenece un profesor, por lo que se crearon los atributos en la relación PROFESOR con este fin. Pero además, sería necesario que cumplieran la regla de integridad referencial, es decir, tendrían que ser claves ajenas que referencien a las relaciones correspondientes, ÁREA y DEPARTAMENTO. Se podría incluso eliminar el atributo "Departamento", ya que con la información del área se podría realizar una consulta para saber el departamento al que pertenece, y de esta forma considerar el atributo

como virtual no se almacenaría y cada vez que se realice una consulta a la relación PROFESOR se calcularía. Ésta es la solución que se presenta en la figura 4 y la que adoptaremos.

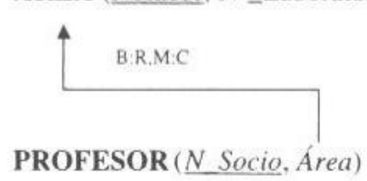


Figura 2.1.55

Opciones de borrado y modificación

La opción de borrado será restringida, ya que no queremos que se eliminen profesores de un área eliminada. La opción de modificación se realizará en cascada.

> *"De los departamentos se guardará su nombre, titulación, y director de éste. Cada departamento está formado por una o varias áreas que se identifican por un nombre, se encuentran en un laboratorio y poseen una línea de investigación principal que también se desea almacenar."*

Para recoger que un departamento puede estar formado por una o varias áreas y suponiendo que un área sólo puede formar parte de un departamento, debemos definir un atributo "Departamento" en la relación ÁREA que tome valores de la relación DEPARTAMENTO, es decir, se definirá una clave ajena.

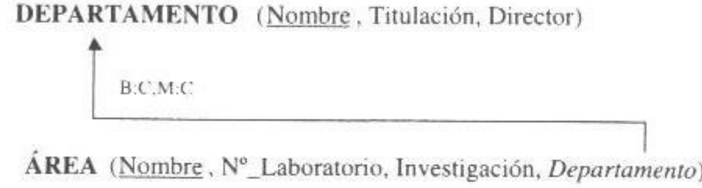


Figura 2.1.56

Opciones de borrado y modificación

Si se elimina o modifica un departamento se tendrán que borrar o modificar las áreas que componen, por lo que las opciones de borrado y de modificación serán en cascada.

Reglas ECA

La restricción de que cada departamento se compone al menos por un área se tendrá que recoger mediante un disparador, ya que cada tupla de DEPARTAMENTO tiene que aparecer por lo menos una vez en la relación ÁREA, es decir, el nombre de los departamentos tendrán que aparecer como valor del atributo "Departamento" en la relación ÁREA por lo menos una vez.

Si se considera que el director del departamento es también un profesor, como en general ocurre en el dominio de las Universidades, el atributo "Director" además de ser obligatorio, será clave ajena que referencia a la relación PROFESOR.

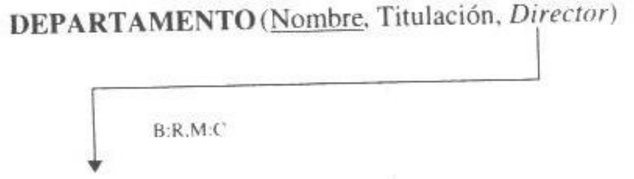


Figura 2.1.57

Opciones de borrado y modificación

No se debería permitir que al borrar un PROFESOR, si éste es director de algún departamento, también se borre éste, por lo que la opción de borrado será restringida y la de modificación en cascada.

Reglas ECA

Podría darse el caso de que en la relación DEPARTAMENTO aparecieran distintas tuplas con el mismo valor en el atributo "Director", lo que significa que un profesor podría ser director de varios departamentos. Haciendo uso del conocimiento del dominio se sabe que este hecho no se da y para controlarlo tendríamos que utilizar un disparador.

> *"Existen fondos en la biblioteca que son comprados por distintas áreas de la Universidad y se necesita conocer esta información, ya que los profesores que solicitan el préstamo de un libro que ha comprado su área tendrá un trato especial. Esto significa que, en general, la filosofía del préstamo es que tanto para alumnos como para profesores, un fondo sólo puede ser prestado como máximo por una semana. Sin embargo, si el profesor pertenece al área que ha proporcionado dicho ejemplar a la biblioteca podrá tenerlo durante un mes."*

Hemos dejado este párrafo para el final, ya que de esta manera, teniendo todas las relaciones básicas, veremos mejor cómo reflejar las restricciones que en él se enuncian.

De los fondos de la biblioteca existen algunos que han sido comprados por un área, información que se necesita recoger para luego saber cómo realizar la política de préstamos. Lo que realmente compra un área es un ejemplar y además se supone que un mismo ejemplar no puede ser adquirido por varias áreas, hecho que se transforma en añadir un atributo, "Área", no obligatorio a la relación EJEMPLAR donde además actuará como clave ajena que referencia a la relación ÁREA.

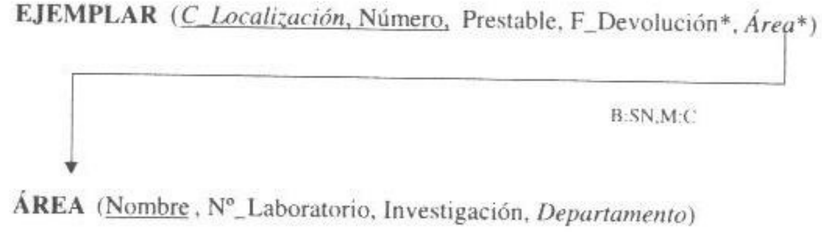


Figura 2.1.58

Opciones de borrado y modificación

Se debe pensar que si borramos un área no sería lógico que se borraran aquellos ejemplares que ésta ha comprado, ya que éstos se mantendrían en la biblioteca aunque el área desapareciese, por lo que se podría dejar con valores nulos. Esto significaría que los ejemplares ya no pertenecerían a ningún área. La modificación sería en cascada para recoger cualquier cambio.

Como en el enunciado no se específica si se quiere un histórico de los préstamos o sólo el estado actual de la biblioteca, supondremos el primer caso y se modelará conforme a él.

Si se hubiera tomado el segundo supuesto significaría que un socio de la biblioteca puede sacar prestados varios ejemplares en un día pero un ejemplar sólo puede ser prestado a un socio. Para modelar este hecho tendríamos que añadir a la relación EJEMPLAR un nuevo atributo no obligatorio, "N_Socio", que además sería clave ajena que referenciaría a USUARIOS.

Como se ha supuesto que se quiere recoger el histórico de los prestamos, se debe reflejar en el grafo relacional que un ejemplar puede ser prestado a uno o varios usuarios a lo largo del tiempo. Y, por supuesto, que un usuario puede sacar prestados de la biblioteca de cero a varios ejemplares. Por lo que tenemos que incluir atributos de fecha que nos indiquen la dimensión temporal de los préstamos en la nueva relación intermedia entre ejemplar y usuario. La clave primaria será la concatenación

cuatro atributos, "N_Socio", "C_Loc", "Num" y "F_Ini". El primero de éstos será ave ajena que referencia a la relación USUARIOS, los dos siguientes también rman una clave ajena que referencia a la relación EJEMPLAR y el último será el que dique la fecha en la que se inició el préstamo.

En esta relación intermedia, que llamaremos PRÉSTAMO, debería aparecer un ributo indicando si el ejemplar se ha devuelto y la fecha de devolución. Por lo que el ributo que aparece en la relación EJEMPLAR, "F_Devolución" desaparecería y Prestable" se calcularía a partir del atributo "Devuelto" de la relación PRÉSTAMO .

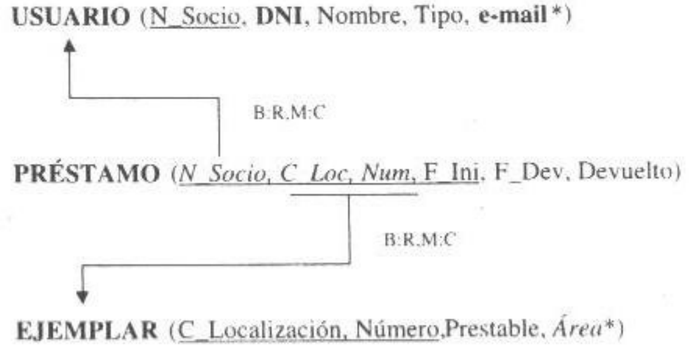


Figura 2.1.59

pciones de borrado y modificación

Las opciones de borrado en ambas claves son restringidas para no permitir que se orren usuarios o ejemplares que a lo mejor no han devuelto un préstamo. Es decir, el orrado en la relación USUARIO debe ser controlado, no se pueden eliminar usuarios ue no hallan devuelto los ejemplares que han tomado prestados de la biblioteca. De gual forma no se pueden eliminar ejemplares que no hallan sido devueltos, ya que de sta forma se perdería el patrimonio de la biblioteca. La modificación se realizará en ascada en las dos restricciones de integridad.

eglas ECA

Para controlar que de la relación PRÉSTAMO no se borre una tupla donde el tributo "Devuelto" tome valor FALSE, se tendrá que asociar a esta relación un isparador que se ejecute en la acción de borrado y compruebe este valor. Si es ALSE la tupla no se borra y, en caso contrario, se eliminará de la relación.

El atributo que nos indica la fecha de devolución de un préstamo se calculará de orma automática, por eso en el grafo relacional aparece como atributo obligatorio. ada vez que se inserte una tupla en PRÉSTAMO tendrá que ejecutarse un disparador ue compruebe que si el usuario es un profesor que pertenece al mismo área que ha roporcionado el ejemplar que quiere sacar prestado, la fecha de devolución del préstamo será un mes más de la fecha de inicio del mismo. En caso contrario, la fecha le devolución será una semana más a la fecha de inicio del préstamo.

GRAFO RELACIONAL

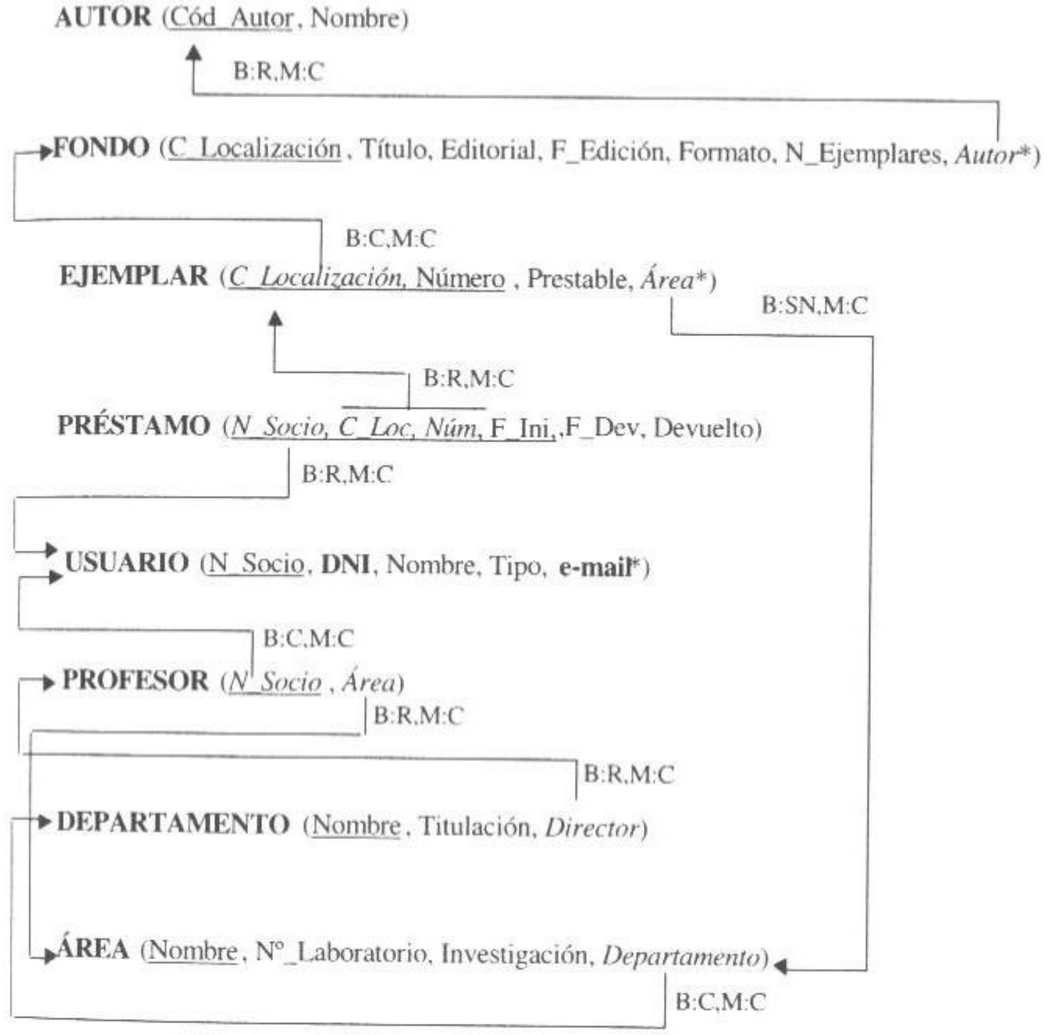


Figura 2.1.60

SCRIPT EN SQL 92

```
CREATE DOMAIN Nombre_Válido CHAR(40);
CREATE DOMAIN Booleano CHAR(5)
    CHECK (VALUE IN ('TRUE', 'FALSE'))
    DEFAULT FALSE;

CREATE DOMAIN Entero_Obligatorio INTEGER NOT NULL;
CREATE DOMAIN Tipo_Socio CHAR(1)
    CHECK (VALUE IN ('P', 'A'));
```

```
REATE TABLE Autor (
    Cód_Autor Entero_Obligatorio,
    Nombre Nombre_Válido NOT NULL,
    PRIMARY KEY (Cód_Autor)


REATE TABLE Fondo (
    C_Localización CHAR(8),
    Titulo Nombre_Válido NOT NULL,
    Editorial Nombre_Válido NOT NULL,
    F_Edición DATE,
    N_Ejemplares Entero_Obligatorio,
    Autor Entero_Obligatorio,
RIMARY KEY (C_Localización),
OREIGN KEY (Autor) REFERENCES Autor
    ON UPDATE CASCADE


REATE TABLE Usuario (
    N_Socio CHAR(8),
    DNI Entero_Obligatorio,
    Nombre Nombre_Válido NOT NULL,
    Tipo Tipo_Socio NOT NULL,
    e-mail Nombre_Válido NOT NULL,
    UNIQUE (DNI),
    UNIQUE (e-mail),
PRIMARY KEY (N_Socio)
;

CREATE TABLE Ejemplar (
    C_Localización CHAR(8),
    Número INTEGER (8),
    Prestable Booleano NOT NULL,
    Área Nombre_Válido,
PRIMARY KEY (C_Localización, Número)
);

CREATE TABLE Departamento (
    Nombre Nombre_Válido,
    Titulación Nombre_Válido NOT NULL,
    Director Nombre_Válido NOT NULL,
PRIMARY KEY (Nombre)
);
CREATE TABLE Área (
    Nombre Nombre_Válido,
    Nº_Laboratorio Entero_Obligatorio,
```



```
    Investigación Nombre_Válido NOT NULL,
    Departamento Nombre_Válido NOT NULL,
PRIMARY KEY (Nombre),
FOREIGN KEY (Departamento) REFERENCES Departamento
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Profesor (
    N_Socio CHAR(8),
    Área Nombre_Válido,
PRIMARY KEY (N_Socio),
FOREIGN KEY (N_Socio) REFERENCES Usuario
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Área) REFERENCES Área
    ON UPDATE CASCADE
);

CREATE TABLE Préstamo (
    N_Socio CHAR(8),
    C_Localización CHAR(8),
    Número INTEGER(8),
    F_Ini DATE,
    F_Dev DATE NOT NULL,
    Devuelto Booleano NOT NULL,
PRIMARY KEY (N_Socio, C_Localización, Número, F_Ini),
FOREIGN KEY (N_Socio) REFERENCES Usuario
    ON UPDATE CASCADE,
FOREIGN KEY (C_Localización, Número) REFERENCES Ejemplar
    ON UPDATE CASCADE
);

ALTER TABLE Ejemplar
    ADD FOREIGN KEY (Área) REFERENCES Área
     ON UPDATE CASCADE
     ON DELETE SET NULL;

ALTER TABLE Departamento
    ADD FOREIGN KEY (Director) REFERENCES Profesor
    ON UPDATE CASCADE;
```

Observación: Hemos tenido que añadir las restricciones de integridad referencial en las relaciones Profesor y Área porque se produce un ciclo referencial. Cuando se inserten tuplas en estas tablas se deberán desactivar las restricciones y luego activarlas una vez insertadas para que el SGBD donde se implemente compruebe la integridad.

NSULTAS

úmero de libros prestados a alumnos de la universidad Carlos III en este último ño

ql

```
SELECT COUNT (*)
FROM Usuario A, Préstamo B
WHERE A.Tipo = 'A' AND B.F_Ini < '31-12-2000'
        AND B.F_Ini > '31-12-1999' AND A.N_Socio = B.N_Socio;
```

lgebra relacional

$$\text{GRUPACIÓN POR (TIPO), CUENTA(*)}\ (\Pi_{N_Socio,Tipo}\ ((\sigma_{Tipo='A'}(USUARIO))*\ (\sigma_{F_Ini<'31\text{-}12\text{-}2000'\ AND\ F_Ini>'31\text{-}12\text{-}1999'}\ (PRÉSTAMO)))$$

Profesores del área de bases de datos que no hayan utilizado el servicio de iblioteca.

Sql

```
SELECT Nombre
FROM Profesor A, Usuario B
WHERE A.N_Socio = B.N_Socio AND A.Área = 'Bases de Datos'
        AND A:N_Socio NOT IN (SELECT N_Socio FROM Préstamo);
```

Álgebra relacional

$$\Pi_{Nombre}\ (PROFESOR * (\Pi_{N_Socio}\ (\sigma_{Área='Bases\ de\ Datos'}(PROFESOR))-\ \Pi_{N_Socio}(PRÉSTAMO *\sigma_{Área='Bases\ de\ Datos'}(PROFESOR)))$$

Título de los ejemplares que no han sido devueltos junto con el nombre del usuario que los posee.

Sql

```
SELECT A.Título, C.Nombre
FROM Fondo A, Ejemplar B, Usuario C, Préstamo D
WHERE D.Devuelto = 'FALSE'
        AND A.C_Localización = B.C_Localización
        AND B.C_Localización = D.C_Loc AND B.Número = D.Num
        AND C.N_Socio = D.N_Socio
```

Álgebra relacional

$$_{Nombre,\ Título}\ (\Pi_{N_Socio,\ Nombre}(USUARIO) * (\Pi_{C_Localización,\ Título}(FONDO) * (\Pi_{C_Localización,\ Número}(EJEMPLAR) * \Pi_{N_Socio,\ C_Loc,\ Num}(\sigma_{Devuelto='FALSE'}(PRÉSTAMO)))))$$

2.2. TRANSFORMACIÓN DE ESQUEMAS E/R A ESQUEMAS RELACIONALES

PROBLEMA 1: CURSOS DE FORMACIÓN

Enunciado

Transformar el esquema E/R obtenido en el problema 2 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.1 (ver figura en la página sigueinte) el esquema E/R del que partiremos.

En primer lugar, transformaremos la entidad *CURSO* y la interrelación reflexiva asociada a ella. La entidad se transformará en una relación, con clave primaria el código del curso ("Codigo_Curso") y con clave alternativa su nombre ("Nombre"),

mientras que la interrelación **Prerrequisito**, por ser N:M, se transformará en otra relación a donde migrará por duplicado, como clave ajena, la clave primaria de *CURSO* (una referenciará a un curso con prerrequisitos y la otra a los cursos que son prerrequisito del anterior). La concatenación de ambas formará la clave primaria de la relación PRERREQUISITO. Además, se añadirá el atributo "Obligatorio".

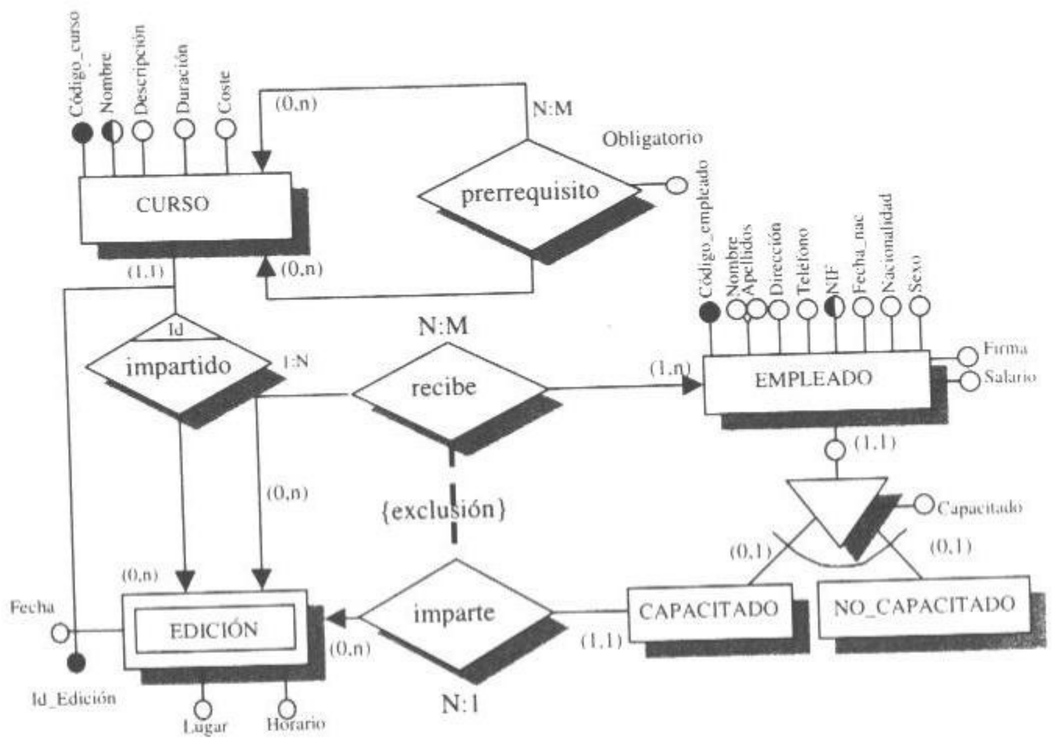


Figura 2.2.1. Esquema E/R a transformar

Los borrados y modificaciones serán en cascada (también podrían ser restringidos dependiendo de la semántica que se quiera recoger), y no será necesario implementar ningún disparador debido a las cardinalidades mínimas cero en ambos extremos de la interrelación.

CURSO (Código_Curso, **Nombre**, Descripción, Duración, Coste)

B:C, M:C B:C, M:C

PRERREQUISITO (*Tiene_Pre*, *Es_Pre*, Obligatorio)

Figura 2.2.2

Podemos ver en el esquema que existe una dependencia en identificación entre los cursos y sus ediciones. Como sabemos, esto se transforma en el modelo relacional añadiendo la clave de los cursos a la de las ediciones. Así, crearemos una nueva relación, EDICIÓN, cuya clave primaria la formarán la fecha de celebración de las ediciones ("Fecha") junto a la clave de los cursos ("Código_Curso"), que a su vez será clave ajena con opciones de borrado y modificación en cascada (o bien restringido, si no permitimos borrar un curso mientras existan ediciones del mismo).

CURSO (Código_Curso, **Nombre**, Descripción, Duración, Coste)

B:C, MC

EDICIÓN (*Código_Curso*, Fecha, Lugar, Horario)

Figura 2.2.3

Como sabemos, para transformar la jerarquía de empleados tenemos varias alternativas. En principio seguiremos la que transforma cada entidad de la jerarquía en una relación. El supertipo *EMPLEADO* propaga su atributo "Código_Empleado" a cada uno de los subtipos (*CAPACITADO* y *NO_CAPACITADO*) que pasan a identificarse por el mismo identificador (como clave ajena). Además, la relación EMPLEADO contendrá el atributo discriminante "Capacitado".

Por ser una jerarquía total, el atributo discriminante "Capacitado" no podrá tomar valores nulos. Además, por ser exclusiva, será necesaria una aserción que asegure que un empleado no pueda estar a la vez en los dos subtipos.

Habría otras soluciones posibles. Por ejemplo, como dijimos cuando diseñamos el esquema E/R, podíamos haber utilizado únicamente un supertipo *CAPACITADO*. En este caso, la jerarquía no sería total ni, evidentemente, exclusiva, y los controles a realizar serían distintos.

EMPLEADO (Código_Empleado, Nombre, Apellidos, Dirección, Teléfono, **Nif**, Fecha_Nac, Nacionalidad, Sexo, Firma, Salario, Capacitado)

B:C, M:C

CAPACITADO (*Código_Empleado*)

B:C, M:C

NO_CAPACITADO (*Código_Empleado*)

Figura 2.2.4

Por último, sólo nos falta transformar las interrelaciones **Recibe** e **Imparte**. **Imparte** no es más que una propagación de la clave de la relación CAPACITADO a la relación EDICIÓN, donde será clave ajena. En cuanto a **Recibe**, será necesaria una nueva relación cuya clave primaria será la concatenación de las claves de EMPLEADO y EDICIÓN, que serán claves ajenas. Será necesario controlar, además, que toda edición de un curso tenga al menos un asistente, es decir, que al menos lo reciba un empleado.

En cuanto a la restricción de exclusividad establecida entre las interrelaciones Imparte y Recibe, deberemos controlarla mediante una aserción que impida que una ocurrencia de una esté presente en la otra.

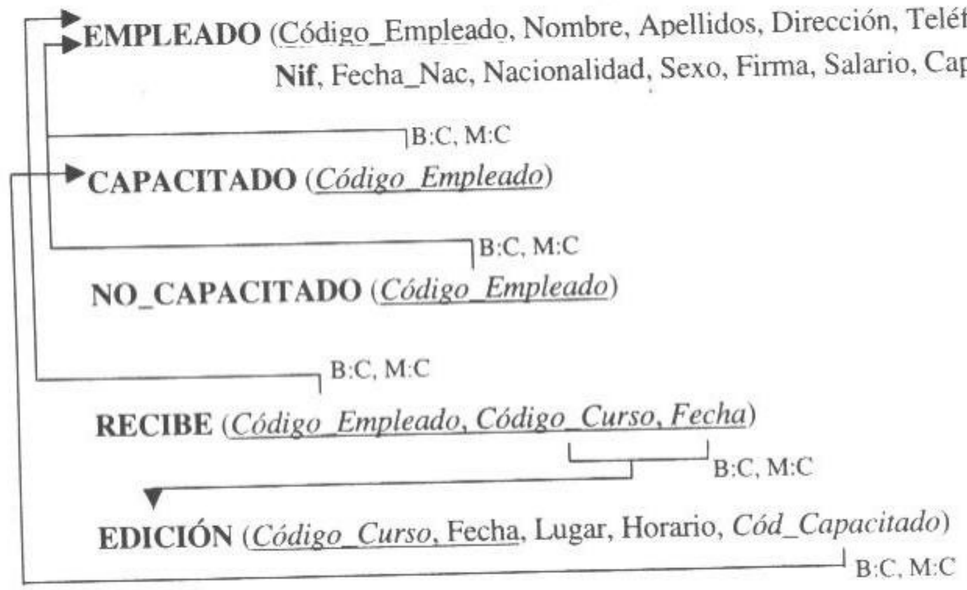


Figura 2.2.5

Propuesta de solución

En la figura 2.2.6 se muestra el grafo relacional con el resultado de la transformación.

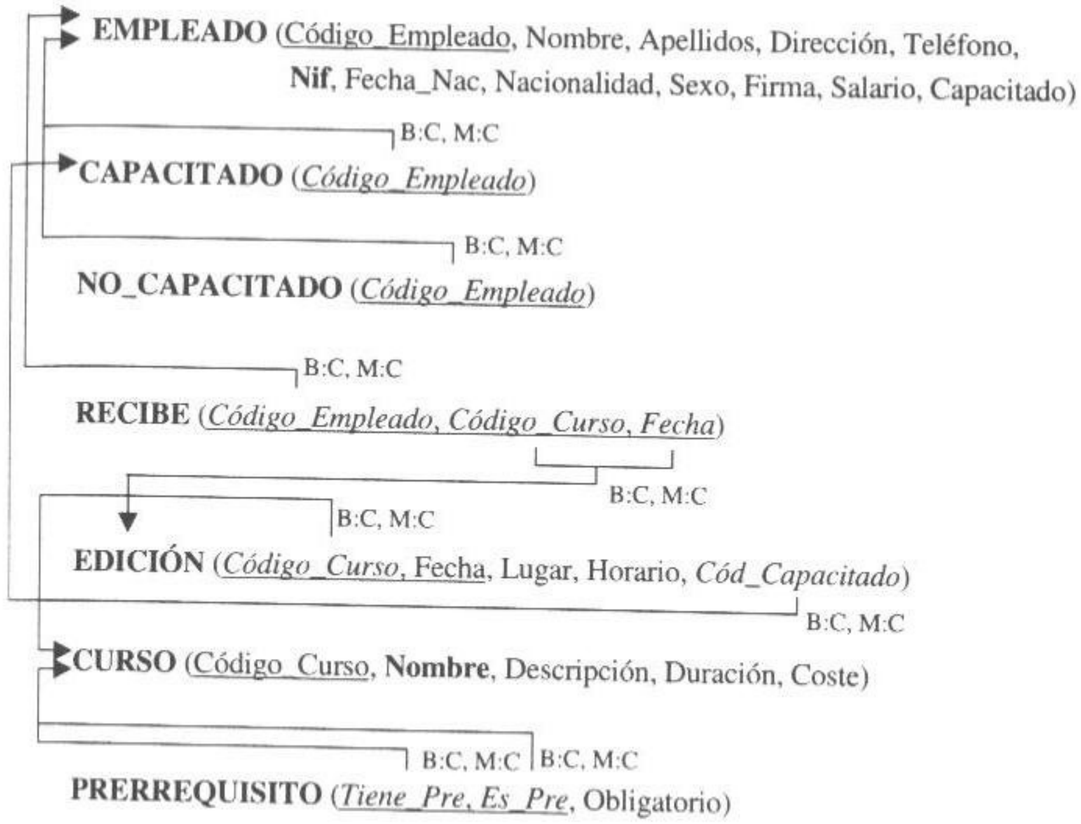


Figura 2.2.6. Esquema relacional solución

Semántica no recogida

Llegados a este punto, se debe considerar con que medios (aserciones, disparadores, procedimientos almacenados, modificaciones a los programas que acceden a la BD,...) se controlarán las siguientes restricciones no abordadas en la transformación al relacional:

- Exclusividad de la jerarquía de empleados.
- Exclusividad entre las relaciones IMPARTE y RECIBE.
- Todo curso lo recibe al menos un empleado.

Por otra parte, en el enunciado original, no hay restricciones que no se hayan podido reflejar en el esquema E/R, ni se considera necesario introducir otras restricciones que no figuraran en el mismo.

PROBLEMA 2: CAMPEONATO DE AJEDREZ

Enunciado

Transformar el esquema E/R obtenido en el problema 3 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.7 el esquema E/R del que partiremos.

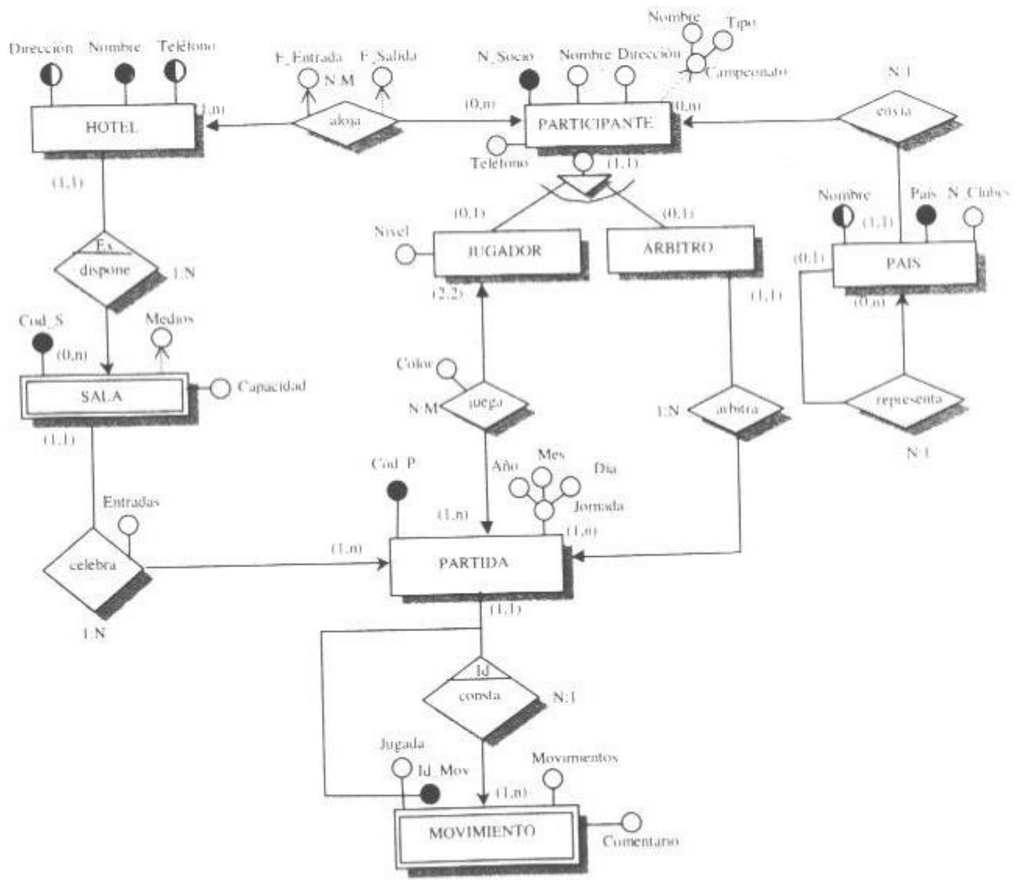


Figura 2.2.7. Esquema E/R a transformar

Se comenzará transformando la generalización creada para los participantes de campeonato. Existen varias formas de realizar la transformación de una jerarquía, en este caso como los subtipos tienen muy pocos atributos propios y buscando

eficiencia crearemos una relación correspondiente al supertipo y veremos mas adelante cómo transformar las interrelaciones asociadas a los subtipos. Por tanto, la entidad *PARTICIPANTE* dará lugar a la relación PARTICIPANTE y sus atributos serán "N_Socio" como clave primaria, "Nombre", "Dirección", "Teléfono" y "Tipo" como atributos que no admiten valores nulos y donde, además, este último será el atributo discriminante que tomará los valores, jugador o árbitro. Se incluirá el atributo "Nivel" para el cual deberá asegurarse que tomará valores nulos para aquellas tuplas correspondientes al tipo de participante árbitro y que tomará un valor en el correspondiente dominio para aquellas tuplas correspondientes al tipo de participante jugador. Como puede observarse, las propiedades de la generalización, totalidad y exclusividad se conservan al introducir el atributo discriminante "Tipo".

El atributo multivaluado "Campeonato" se transforma en una relación denominada CAMPEONATO cuya clave primaria será la concatenación de los atributos "N_Socio" y "Nombre" y donde "N_Socio" será clave ajena que referencia a la relación PARTICIPANTE. Las opciones de borrado y de modificación de dicha clave serán en cascada.

PARTICIPANTE(N_Socio, Nombre, Dirección, Teléfono, Tipo, Nivel*)

B:C, M:C

CAMPEONATO (N_Socio, Nombre, Tipo)

Figura 2.2.8

La entidad *PAÍS* se convierte en una relación PAÍS cuya clave primaria es "País", el "Nombre" es clave alternativa y "N_Clubes" un atributo que no admite valores nulos. Como se indica en el enunciado la clave primaria es un número correlativo según el orden alfabético que ocupe el país, restricción que podríamos implementar mediante un disparador que cada vez que se insertara un nuevo país calculara la posición que ocupa y desplazara la de los países que ocupan una posición posterior a este.

La entidad *PAÍS* participa en dos interrelaciones binarias que pasamos a estudiar:

La interrelación **Envía**, cuyo tipo de correspondencia es 1:N, propaga la clave de PAÍS a la relación PARTICIPANTE, relación esta que pasa a tener otro atributo más, "País", que no admite valores nulos y cuya opción de borrado será restringida para evitar que se eliminen participantes al intentar borrar un país y la de modificación en cascada.

La interrelación **Representa** es reflexiva y para evitar valores nulos, en vez de realizar una propagación de clave, crearemos una nueva relación, REPRESENTA, compuesta de dos atributos "País" y "País_Representado". Ambos atributos juntos formarían en principio la clave primaria, sin embargo, dado que cada país

epresentado sólo puede aparecer una vez, se tendría que la relación REPRESENTA endría a lo sumo tantas tuplas como la relación PAÍS, por tanto con el atributo País_Representado" sería suficiente para identificar cada tupla de la relación EPRESENTA, por lo que será su clave primaria. Cada uno de estos dos atributos es lave ajena que referencia a la relación PAÍS. En ambos casos las opciones de borrado modificación serán en cascada. Se tendría que comprobar que, para una misma tupla, l valor de "País" es distinto al de "País_Representado" y que un país representado no uede representar a otros países.

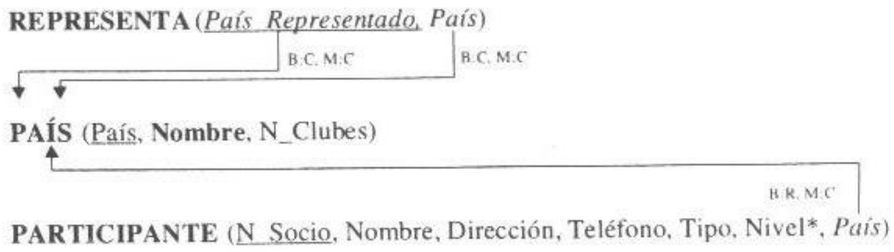


Figura 2.2.9

La entidad *PARTIDA* se transforma en la relación PARTIDA. Esta relación endrá como clave primaria el atributo "Cód_P" y como atributos que no admiten valores nulos a "Año", "Mes" y "Día". La interrelación que asocia *PARTIDA* con *JUGADOR* es N:M, pero no se transformará en una relación sino que excepcionalmente, dada la cardinalidad (2,2), para optimizar las comprobaciones que se tendrían que realizar si creáramos una nueva relación, como por ejemplo, que una partida sólo la jueguen dos jugadores y que cada uno de ellos participe con un color de fichas distinto, se ha optado por introducir en la relación PARTIDA dos atributos. "Jugador1" y "Jugador2". Ambos serán claves ajenas que referencian a PARTICIPANTE y que tendrán como opción de borrado restringida y de modificación en cascada. Además, se añadiría un atributo "Color1" que tomaría el valor negro o blanco dependiendo del color de las fichas con las que juegue el jugador 1. No sería necesario que se introduzca otro atributo para el color de las fichas del jugador 2, ya que evidentemente será el color no elegido para el jugador 1. Ni siquiera es necesario este atributo si se adopta (como nosotros haremos) el convenio de que el jugador 1 siempre es el que juega con blancas y el jugador 2 el que juega con negras. Las únicas reglas de verificación que se deben crear son que el jugador 1 sea distinto del jugador 2 y que los valores que toman ambas claves ajenas se correspondan en la relación PARTICIPANTE a una tupla cuyo atributo "Tipo" toma el valor jugador.

También existe una interrelación entre *PARTIDA* y *ÁRBITRO* que al ser 1:N propagará la clave de PARTICIPANTE a la relación PARTIDA, por lo que se tendrá un atributo "Árbitro" que no admite valores nulos, clave ajena que referencia a PARTICIPANTE con borrado restringido y modificación en cascada. Se necesitaría la restricción de que el valor que toma el atributo "Árbitro" se corresponde en la relación PARTICIPANTE a una tupla cuyo atributo "Tipo" toma el valor árbitro.

Al meter en la relación PARTIDA las claves ajenas "Jugador1", "Jugador2" y "Árbitro" no admitiendo para ellas valores nulos recogemos la cardinalidad mínima y máxima de dos jugadores y un árbitro por partida. Sin embargo no podemos recoger en el esquema relacional que todo participante participe como mínimo una vez en el campeonato.

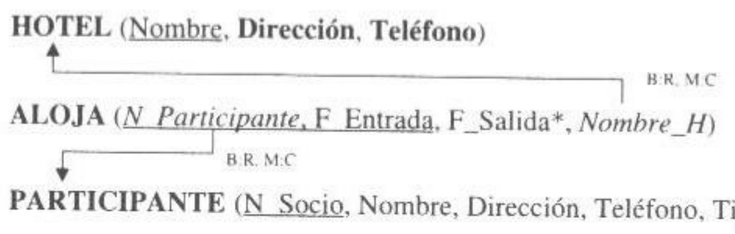


Figura 2.2.10

Los participantes del campeonato se alojan en hoteles, entidad que se transforma en una relación HOTEL cuya clave primaria es "nombre" y como claves alternativas, la "dirección" y el "teléfono". Existe una interrelación N:M entre *HOTEL* y *PARTICIPANTE* que se transformará en una relación denominada ALOJA cuya clave primaria estará compuesta por dos atributos, "Nombre_H" y "N_Participante", que respectivamente referencian a HOTEL y a PARTICIPANTE, ya que también son claves ajenas. La opción de borrado para "Nombre_H" es restringida, ya que no se puede permitir que borren un hotel y se pierda información acerca de los participantes que estaban alojados en el mismo. Las demás opciones que faltan serán en cascada. No se puede recoger en el esquema que un participante al menos se ha alojado en un hotel, restricción que se deberá considerar posteriormente.

Como la interrelación posee dos atributos multivaluados, uno obligatorio y otro opcional, que no habíamos considerado en la relación ALOJA, podría ocurrir que para un mismo participante y un mismo hotel tengamos distintos valores, ya que se ha podido alojar en el mismo hotel durante períodos de tiempo distintos. Para solucionar este problema se ha adoptado la solución de aumentar la clave primaria con la fecha de entrada, pero puede observarse que bastará para identificar a todas las tuplas los atributos "N_Participante" y "F_Entrada", ya que nunca ocurrirá que un participante se aloje en la misma fecha en dos hoteles distintos.

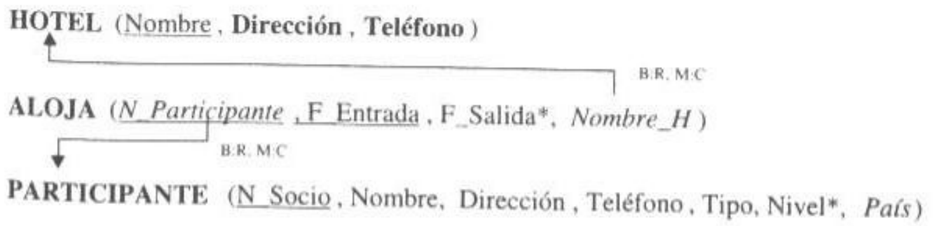


Figura 2.2.11

Transformaremos la entidad débil *SALA* en una relación, SALA, con clave maria el atributo, "Cód_S". Esta relación tendrá, además, los atributos "Capacidad" N_Hotel", donde este último será clave ajena que referencia a la relación HOTEL y e por provenir de la transformación de una interrelación en existencia las opciones borrado y de modificación serán en cascada. El atributo mutivaluado "Medios" se nsformará en una relación MEDIO cuya clave primaria estará compuesta por los ibutos "Sala" y "Medio". El primero referencia a la relación SALA y su opción de rrado y modificación será en cascada.

Las partidas se juegan en una determinada sala, por lo se ha de añadir a la ación PARTIDA un atributo no nulo denominado "N_Sala" que además será clave ena que referencia a SALA y que tendrá como opción de borrado restringido y de odificación en cascada. También se añadirá el atributo "entradas" que proviene de la terrelación **Celebra**.

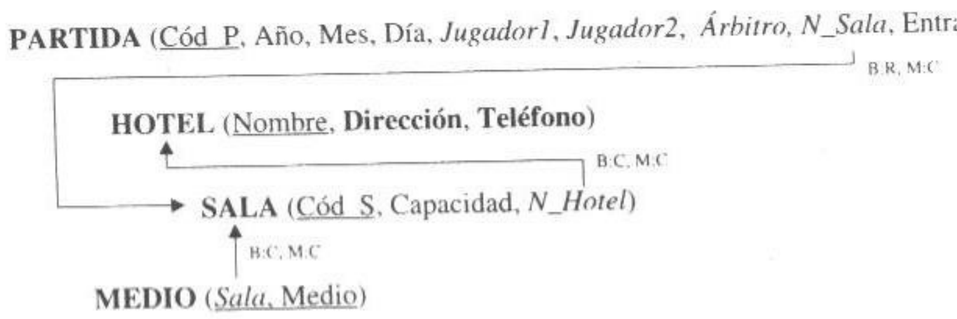


Figura 2.2.12

Por último, sólo queda transformar la entidad *MOVIMIENTO*, que se convertirá n una relación llamada MOVIMIENTO, cuya clave primaria será la concatenación de os atributos "Jugada" y "Partida", ya que tenemos que esta entidad se relaciona con la ntidad *PARTIDA* mediante una interrelación en identificación. Por tanto, el atributo "Partida" también será clave ajena con opciones de borrado y modificación en cascada.

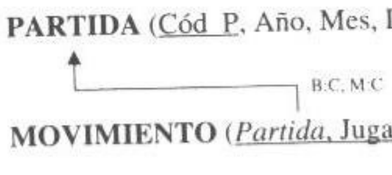


Figura 2.2.13

Propuesta de solución

En la figura 2.2.14 se muestra el grafo relacional con el resultado de la transformación.

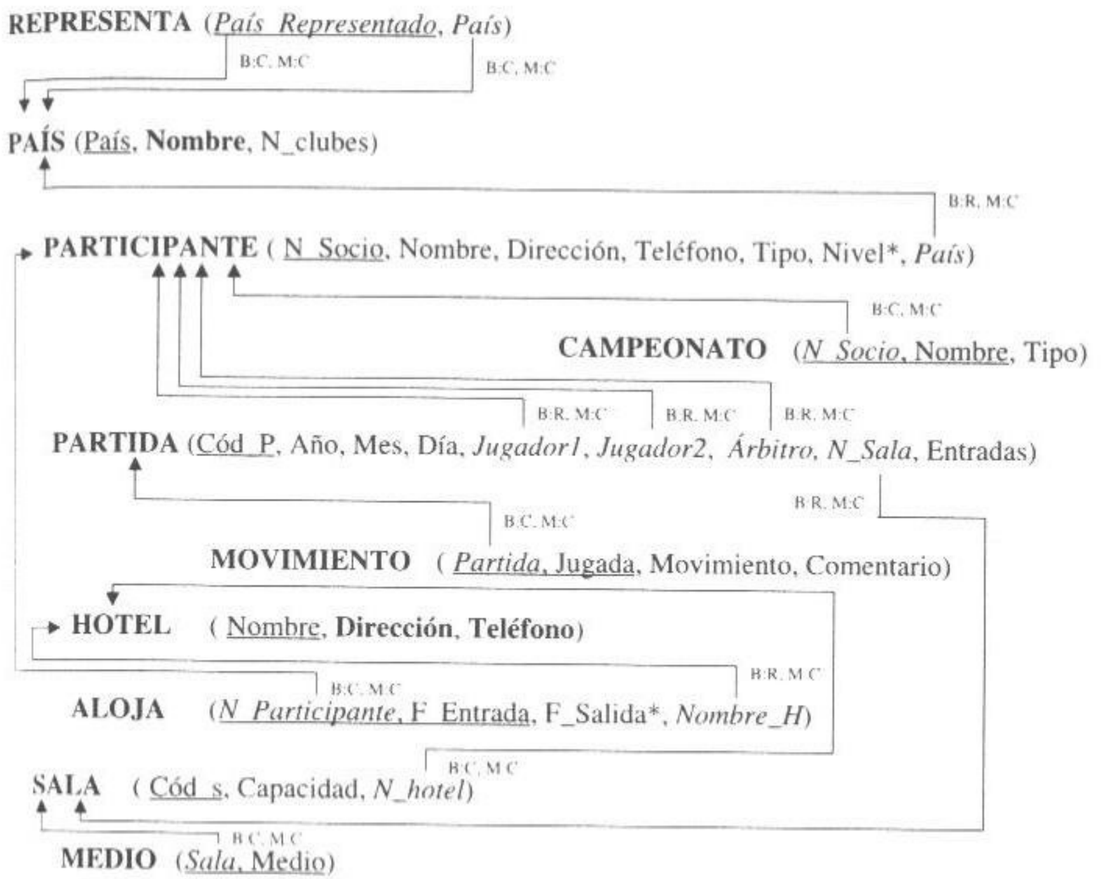


Figura 2.2.14. Esquema relacional solución

Semántica no recogida

Además de la semántica considerada en el correspondiente ejercicio del primer capítulo, se deberían considerar las siguientes restricciones:

- En la relación PARTICIPANTE se debería construir una regla de verificación para controlar que las tuplas correspondientes a los árbitros del campeonato tengan valor nulo en el atributo "Nivel", mientras que las tuplas correspondientes a los atributos han de tener un valor no nulo en "Nivel".
- En la relación PARTIDA se tendría que crear una regla de verificación o CHECK para controlar que un jugador no puede jugar consigo mismo.
- Se debería controlar con disparadores que las claves ajenas "Jugador1" y "Jugador2" en PARTIDA son valores correspondientes a jugadores, y que el atributo "Árbitro" tome el valor de un participante que es un árbitro.
- Todo participante se aloja en un hotel.

- En cualquier sala se celebra al menos una partida.
- Todo jugador juega al menos una partida.
- Cualquier árbitro arbitra al menos una partida.
- En toda partida siempre hay como mínimo un movimiento.

Se deben proporcionar elementos para calcular:

- Secuencialidad del código de país.
- Secuencialidad del código de partida.
- Secuencialidad del número de movimiento dentro de cada partida.

Debe considerarse además, aquella semántica presente en el enunciado y que no pudo reflejarse ni en el esquema E/R ni en el esquema relacional:

- Un árbitro no puede arbitrar partidas en las que participen jugadores de su mismo país.

Por último, se han de incluir aquellas restricciones que se consideran convenientes o adecuadas:

- El número de entradas vendidas para una partida, no puede superar la capacidad de la sala en la que se celebra la partida.
- Un país que representa a otro no puede a su vez ser representado por ningún otro.
- Separación entre intervalos temporales de estancia de un participante en el mismo hotel.
- Corrección de los anteriores intervalos.
- Un movimiento se describe con una cadena de cinco caracteres correctamente compuesta[29] y que representa un movimiento legal para la partida que se esta jugando.

PROBLEMA 3: ENERGÍA ELÉCTRICA

Enunciado

Transformar el esquema E/R obtenido en el problema 4 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

[29] Se entiende, para los expertos en ajedrez, que se trata de la notación algebraica.

Discusión del enunciado

Para realizar la transformación partiremos del esquema mostrado en la figura 2.2.15.

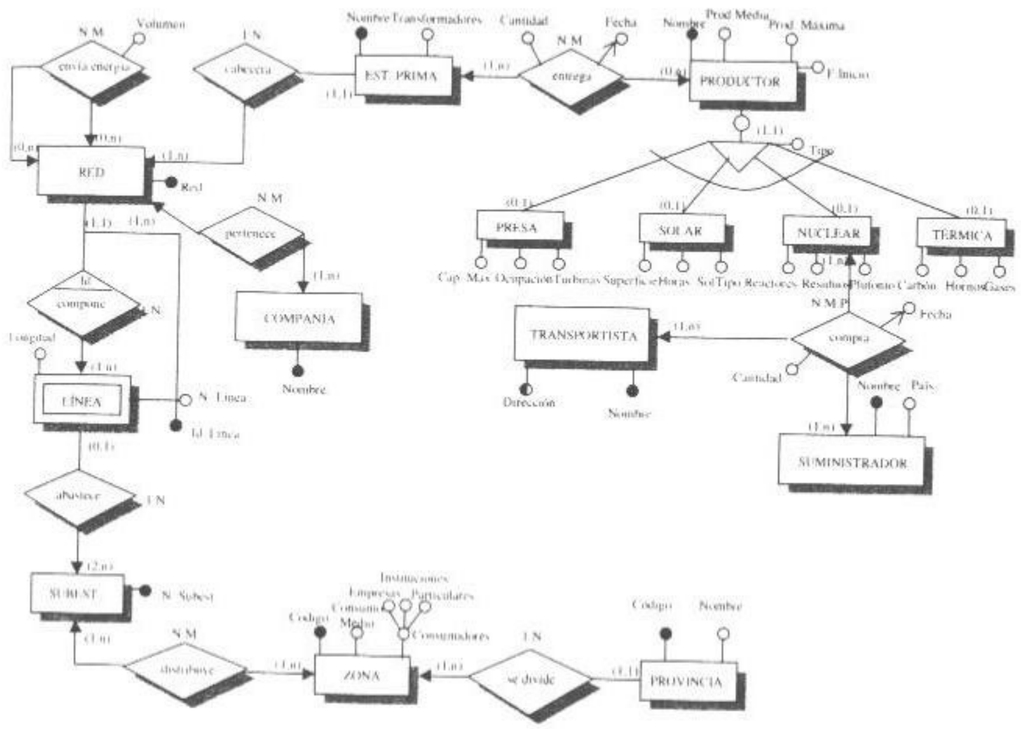


Figura 2.2.15. Esquema E/R a transformar

Como ya sabemos, toda entidad del modelo E/R se transformará en una relación en el modelo relacional. Así las entidades *ZONA*, *RED*, *PROVINCIA*, *COMPAÑÍA*, *TRANSPORTISTA*, *SUBEST*, *SUMINISTRADOR* y *EST.PRIMA.*, se transformarán, respectivamente, en las relaciones que se muestran a continuación.

ZONA (Código, Consumo_Medio, Núm_Part, Núm_Emp, Núm_Inst)

RED (Red)

PROVINCIA (Código, Nombre)

COMPAÑÍA (Nombre)

TRANSPORTISTA (Nombre, **Dirección**)

SUBEST (N_subest)

SUMINISTRADOR (Nombre, Pais)

EST_PRIMA (Nombre, Transformadores)

Figura 2.2.16

La entidad *LÍNEA* es una entidad débil con respecto a la entidad *RED*, que depende de esta última en identificación, con lo que su clave primaria estará formada por el atributo "N_Línea" y por la clave primaria de la relación RED. Este último atributo será además, clave ajena de la relación LÍNEA y sus opciones de borrado y modificación serán en cascada.

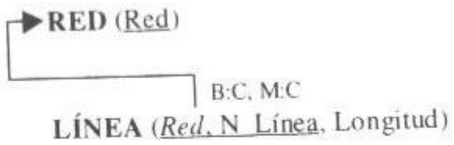


Figura 2.2.17

En cuanto a la transformación de la jerarquía que aparece en el esquema E/R, los productores se dividen en cuatro categorías o subtipos: *PRESA*, *SOLAR*, *NUCLEAR* y *TÉRMICA*. Como todos estos subtipos tienen atributos propios, y el supertipo también, la transformación al relacional se hará considerando cada una de las entidades que conforman la jerarquía como una relación, de esta forma, tendremos las relaciones PRODUCTOR, PRESA, SOLAR, NUCLEAR y TÉRMICA. Las cuatro últimas (las que provienen de los subtipos de la jerarquía) tendrán como clave primaria la clave primaria de la relación PRODUCTOR que, además, será clave ajena que referencia a esa relación. El borrado y la modificación en cualquiera de los casos se hará en cascada. Como la jerarquía es total el atributo discriminante Tipo no podrá admitir valores nulos. La exclusividad de la jerarquía se debería controlar con una aserción.

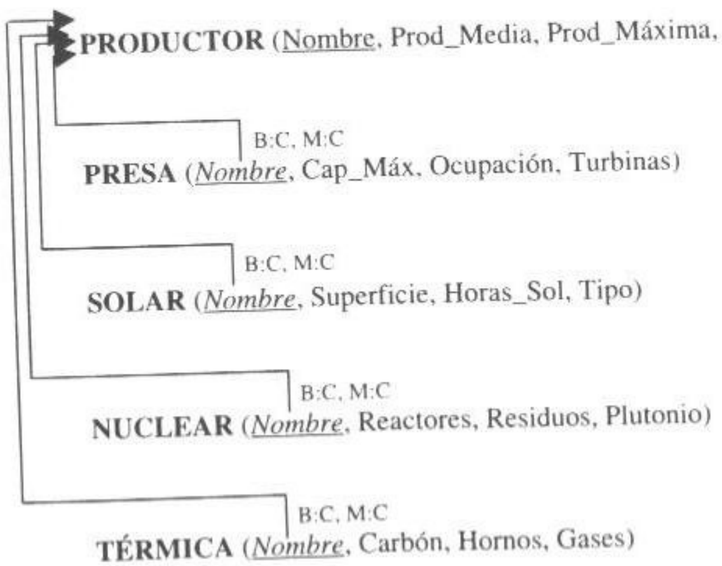


Figura 2.2.18

Comenzamos el estudio de la transformación de las interrelaciones 1:N.

En primer lugar tenemos una interrelación **Cabecera** que enlaza las entidades *EST. PRIMA* y *RED*. La cardinalidad máxima "n" está situada en el lado de la entidad *RED*, esto significa que una estación primaria es cabecera de varias redes, pero una red solo puede tener como cabecera una estación primaria, por lo tanto la transformación la haremos propagando la clave primaria de la relación EST_PRIMA hacia la relación RED y este atributo será clave ajena de la relación RED que referencia a EST_PRIMA. Como toda red tiene que tener una estación primaria como cabecera, la clave ajena no puede admitir valores nulos y por lo tanto el borrado no podrá hacerse con puesta a nulos, tendrá que hacerse en cascada o restringido. La modificación se hará en cascada.

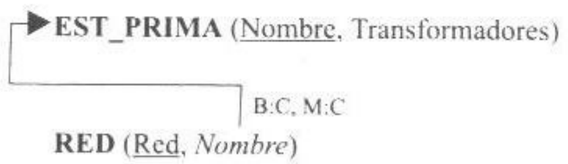


Figura 2.2.19

De la misma forma, la interrelación **Abastece** se transforma propagando la clave de la relación LINEA a la relación SUBEST. Además, como una subestación puede ser o no abastecida por una línea, la clave ajena puede admitir valores nulos y el borrado puede hacerse con puesta a nulos, en cascada o restringido.

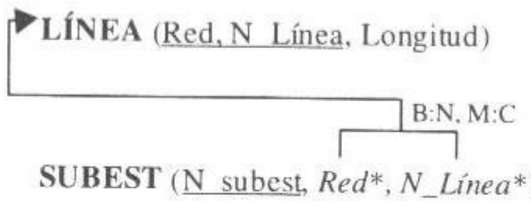


Figura 2.2.20

Por último, la interrelación existente entre las entidades *ZONA* y *PROVINCIA*, se transforma, como en los casos anteriores, propagando la clave primaria de la relación PROVINCIA hacia la relación ZONA. En este caso la clave ajena no puede admitir valores nulos, pues cada zona está adscrita a una determinada provincia. El borrado de la clave ajena debe ser restringido para evitar que desaparezcan las zonas cuando se elimine una provincia de la cual dependan. La modificación será en cascada.

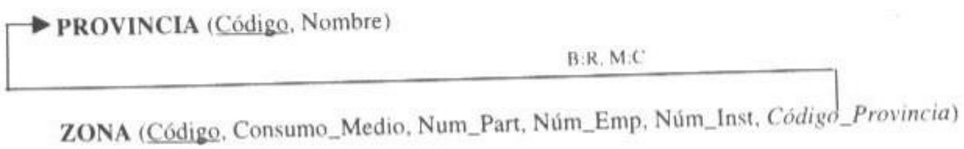


Figura 2.2.21

Continuando el estudio de las interrelaciones, vamos a comentar la transformación de las interrelaciones N:M.

Entre las entidades *SUBEST* y *ZONA* existe una interrelación **Distribuye** que es N:M, pues cada subestación distribuye a varias zonas y a una misma zona la pueden distribuir varias subestaciones. La transformación de esta interrelación se hará construyendo una nueva relación SUBEST_ZONA cuya clave primaria estará compuesta por la concatenación de las claves primarias de las relaciones SUBEST y ZONA, además cada una de ellas será clave ajena que referencia a la relación de la cual provienen. El borrado de ambas se hará en cascada o restringido mientras que la modificación será en cascada.

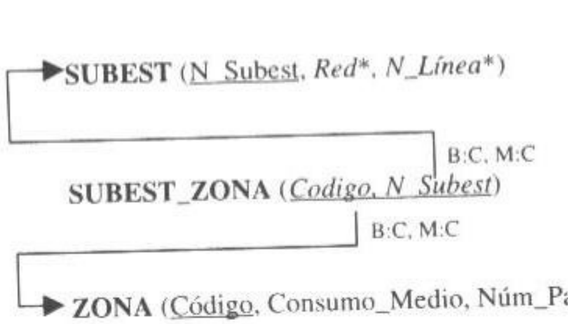


Figura 2.2.22

La interrelación **Pertenece** que enlaza las entidades *RED* y *COMPAÑIA* es también una interrelación N:M y por tanto su transformación es análoga al caso anterior. Los borrados de las claves ajenas son, como antes, en cascada o restringido, las modificaciones también son en cascada.

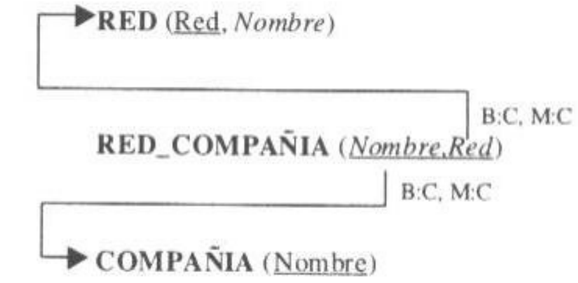


Figura 2.2.23

La entidad *RED* posee una interrelación reflexiva N:M que, además, tiene un atributo propio "Volumen". La transformación se hará considerando una nueva relación ENVÍA_ENERGÍA cuya clave primaria estará formada por la red que envía energía y la red a la cual es enviada la energía. Ambas serán claves ajenas que referencian a la relación RED. Además la relación ENVÍA_ENERGÍA tendrá un atributo propio "Volumen", pues en el enunciado se nos decía que se deseaba recoger la cantidad de energía intercambiada entre dos redes. El borrado se hará restringido en ambos casos para conservar esta cantidad, y la modificación será en cascada.

Figura 2.2.24

La última interrelación N:M que nos queda por analizar es **Entrega**, que enlaza las entidades *EST. PRIMARIA* y *PRODUCTOR*. En principio y como ocurre con todas las interrelaciones N:M, su transformación se hará considerando una nueva relación como en los casos anteriores, pero en este caso, la interrelación posee además un atributo multivaluado "Fecha". Debemos analizar cuál será la clave primaria de la nueva relación; en principio las claves primarias de las relaciones EST_PRIMARIA y PRODUCTOR se propagan hacia la relación intermedia EST_PRODUCTOR, pero como un productor puede hacer entregas a la misma estación primaria en distintas fechas y, además, una estación primaria puede recibir diariamente energía de cada uno de los productores, podemos incluir el atributo "Fecha" en la relación EST_PRODUCTOR, haciendo que pase a formar parte de la clave primaria de la relación, junto con las claves primarias de las relaciones EST_PRIMARIA y PRODUCTOR, que serán claves ajenas que referencian a dichas relaciones. El borrado de ambas claves ajenas será restringido y la modificación en cascada.

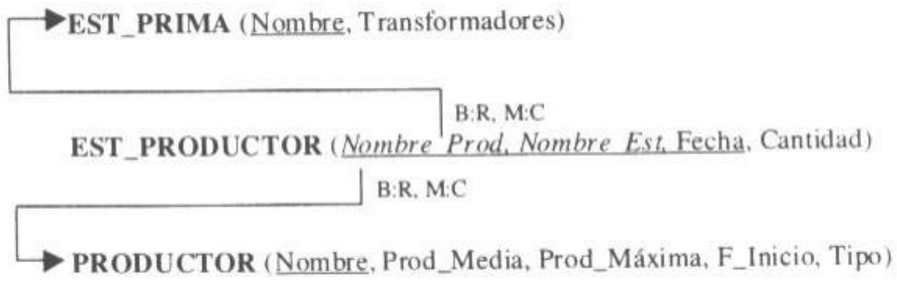


Figura 2.2.25

Por último nos queda por comentar la transformación de la relación ternaria **Compra**, existente entre las entidades *NUCLEAR*, *TRANSPORTISTA* y *SUMINISTRADOR*. Se trata de una interrelación ternaria de cardinalidades máximas "n" en cada uno de sus extremos y con un atributo multivaluado "Fecha" que indica que cada central nuclear puede comprar plutonio a un mismo suministrador y que este plutonio puede ser transportado por un mismo transportista en distintas fechas, lo cual indica que la clave primaria debe estar formada por la concatenación de las claves primarias que provienen de las relaciones NUCLEAR, TRANSPORTISTA y SUMINISTRADOR (y que serán claves ajenas que referenciarán a cada una de las relaciones de las cuales son claves primarias) más la fecha, como se muestra en la siguiente figura. El borrado de las tres claves ajenas será restringido para mantener el histórico de las compras de plutonio. La modificación será en cascada.

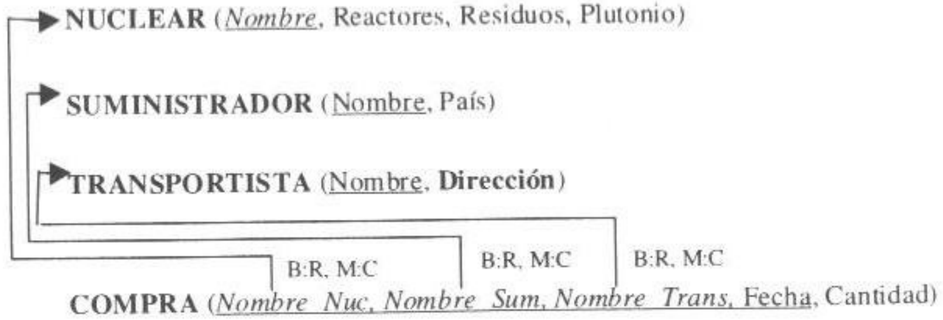


Figura 2.2.26

Propuesta de solución

En la figura 2.2.27 se muestra el grafo relacional con el resultado de la transformación.

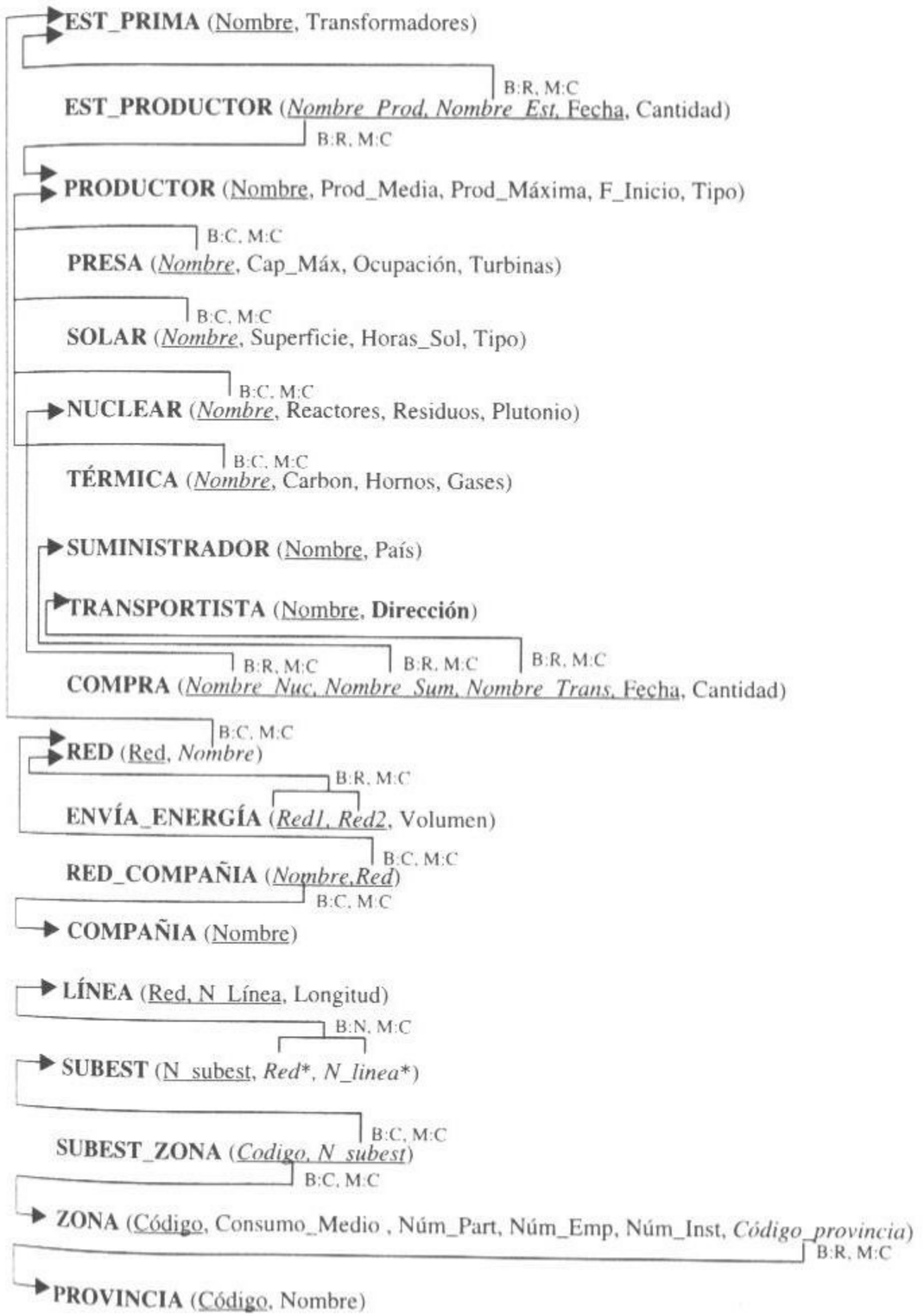


Figura 2.2.27. Esquema relacional solución

emántica no recogida

Se deben controlar las siguientes restricciones perdidas en la transformación al acional:

Exclusividad de la jerarquía de productores.

Todo productor entrega la energía que produce al menos a una estación primaria.

Cualquier estación primaria es cabecera de al menos una red de distribución.

Toda compañía eléctrica es propietaria de al menos una red de distribución.

Una red de distribución es propietaria de al menos una compañía eléctrica.

Cualquier red de distribución esta compuesta por al menos una línea.

Una línea abastece al menos a dos subestaciones.

Toda subestación distribuye al menos a una zona.

A una zona le distribuye como mínimo una subestación.

Una provincia se divide en al menos una zona.

Habría que considerar, además, la siguiente restricción que no pudo reflejarse en esquema E/R.

La energía sobrante de una red no puede enviarse a sí misma.

Por último, no se considera necesario introducir otras restricciones que no guraran en el problema original.

PROBLEMA 4: CONFLICTOS BÉLICOS

Enunciado

Transformar el esquema E/R obtenido en el problema 5 del capítulo 1 en un squema relacional en el que se indiquen las claves primarias y candidatas, así como as claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.28 el esquema E/R del que partiremos.

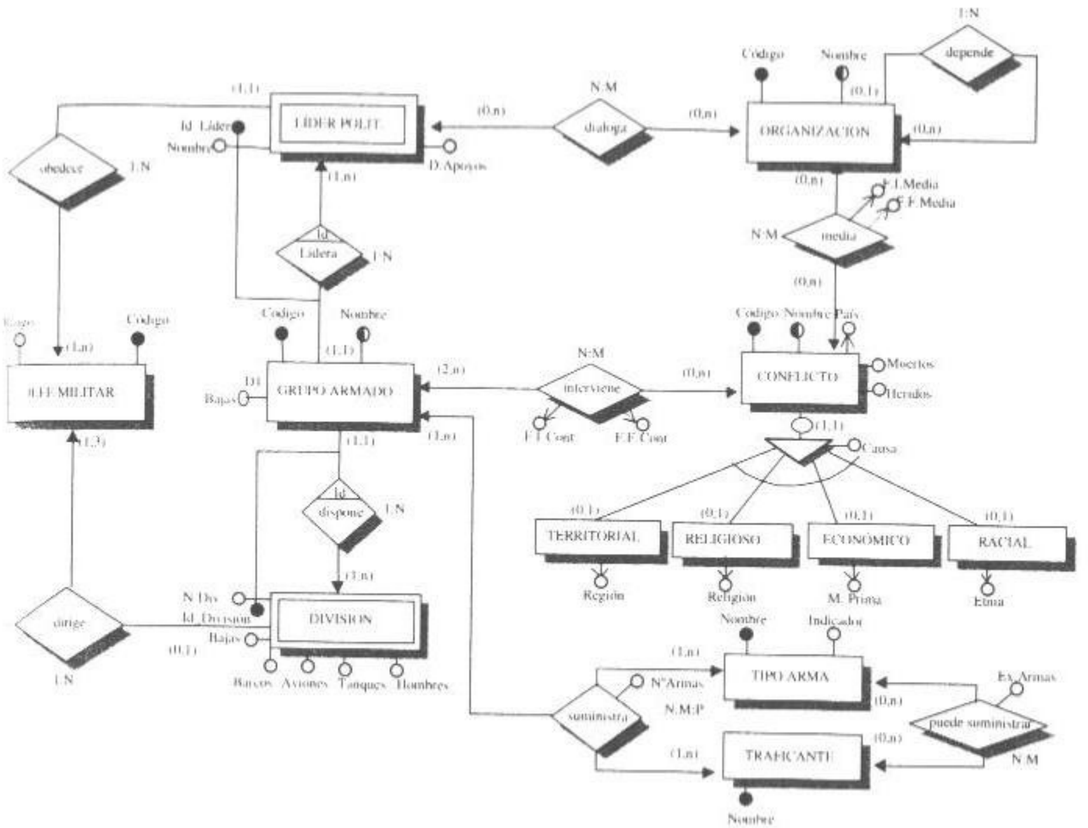


Figura 2.2.28. Esquema E/R a transformar

Comenzamos con la entidad *CONFLICTO* junto con la generalización cuyo supertipo es dicha entidad. Su transformación se realiza construyendo una relación denominada CONFLICTO compuesta por los siguientes atributos: "Código_C" como clave primaria, "Nombre" como clave alternativa, "Muertos" y "Heridos" donde estos dos últimos admiten valores nulos. El atributo multivaluado "País" se transforma en una relación PAÍS cuya clave primaria se formará con la concatenación de dos atributos, "País" y "Código_C". El atributo "Código_C" además de formar parte de la clave primaria es clave ajena que referencia a la relación CONFLICTO y cuyas opciones de borrado y modificación son en cascada.

Para facilitar el control de la totalidad de la jerarquía, se incluye el atributo discriminante "Causa" que, naturalmente, no admitirá valores nulos. La exclusividad de la jerarquía se controlará con una aserción.

Dado que los subtipos de la generalización tienen atributos propios cada uno de los mismos se transforma en una relación cuya clave primaria sería "Código_C", aunque para evitar que cada uno de los atributos multivaluados que estos subtipos tienen se convierta en otra relación y dado que no tienen ni más atributos ni interrelaciones propias, se ha pensado en tomar como clave primaria el código del

conflicto, "Código_C" y el nombre de la región, la religión, la materia prima o la etnia dependiendo del subtipo transformado. El "Código_C" es clave ajena que referencia a CONFLICTO con borrado y modificación en cascada en todos los casos.

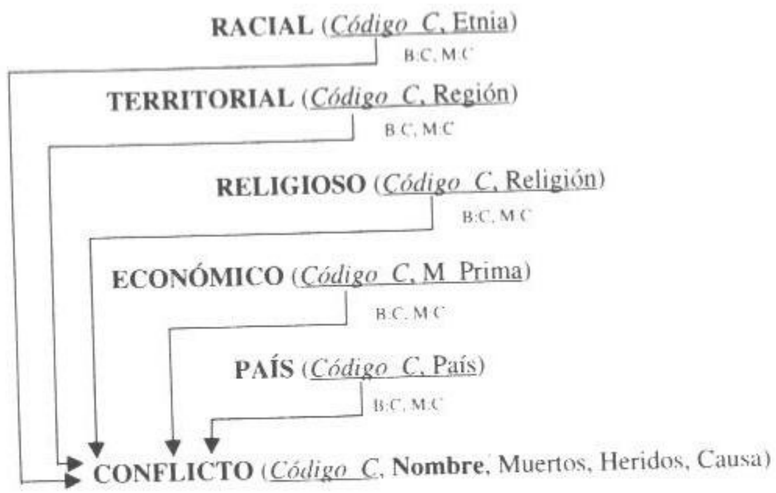


Figura 2.2.29

La entidad *GRUPO ARMADO* se convierte en una relación, GRUPO, cuya clave primaria será "Código" y como clave alternativa tendrá el atributo "Nombre". Tenemos un atributo derivado "N_Bajas" que se convierte en un atributo de esta entidad. El valor de este atributo se calculará con un disparador asociado a la relación DIVISIÓN (proviene de la entidad *DIVISIÓN*) que en cada actualización se encargará de recalcular el número total de bajas.

La relación DIVISIÓN se compone de los atributos: "Grupo" y "División", juntos actúan como clave primaria, ya que, existe una interrelación en identificación entre esta tabla y GRUPO, por lo que el atributo "Grupo" será clave ajena que referencia a GRUPO y sus opciones de borrado y modificación serán en cascada. Otros atributos cuyo valor es obligado, serán "Hombres", "Tanques", "Aviones", "Barcos" y "Bajas" y su valor por defecto será cero.

Como las entidades *CONFLICTO* y *GRUPO ARMADO* se encuentran relacionadas mediante una interrelación N:M se creará una relación denominada INTERVIENE cuya clave primaria no sólo estará formada por las claves primarias de las dos relaciones que asocia sino que además se incluirá el atributo "F_Ini" para que de esta forma se pueda asegurar la unicidad de la clave primaria. Esta relación tiene otro atributo que admite valores nulos denominado "F_Fin". En la transformación no se reflejaría que en un conflicto por lo menos intervienen dos grupos armados, restricción ésta que se tendrá que considerar posteriormente. Los atributos "Grupo" y "Conflicto" son a su vez claves ajenas que referencian a GRUPO y CONFLICTO respectivamente, y cuyas opciones de borrado y modificación pueden ser restringidas o en cascada. Elegiremos la opción de borrado restringido y la modificación en cascada.

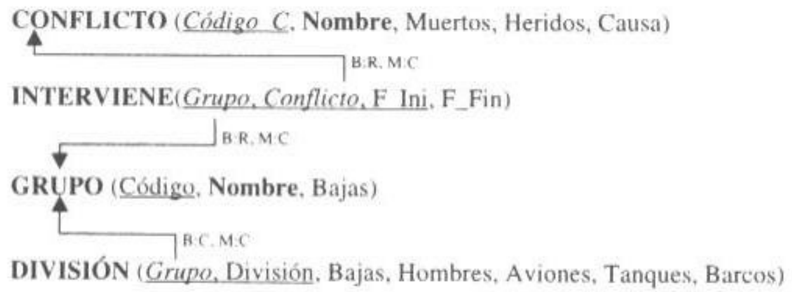


Figura 2.2.30

Construimos las relaciones ARMA y TRAFICANTE provenientes de las correspondientes entidades. ARMA se compone del atributo "Nombre", clave primaria de la relación, e "Indicador" cuyos valores no admiten nulos. La relación TRAFICANTE sólo tiene un atributo, "Nombre" que obviamente es clave primaria. Para recoger las armas que pueden suministrar y la cantidad, se crea una relación STOCK con clave primaria compuesta por dos atributos "N_Arma" y "N_Traficante" cada uno de estos atributos es clave ajena que referencia a ARMA y TRAFICANTE, respectivamente. La opción de borrado para "N_Arma" será restringida, sin embargo para "N_Traficante" será en cascada. En ambas la opción de modificación será en cascada. Esta relación tendrá un atributo "Cantidad" que controla el número de armas de cada tipo que puede proporcionar un determinado traficante.

La interrelación ternaria existente entre los tipos de entidad *TIPO ARMA*, *TRAFICANTE* y *GRUPO ARMADO* se transforma en una relación, SUMINISTRA, cuya clave primaria se compone de tres atributos, "Grupo", "Arma" y "Traficante" donde cada uno de ellos es clave ajena que referencia, respectivamente, a las relaciones GRUPO, ARMA y TRAFICANTE. Para la clave ajena "Grupo" se tendrá la opción de borrado y de modificación en cascada. Para el resto serán borrado restringido y modificación en cascada. Aunque como hemos dicho las opciones podrían ser otras dependiendo de la semántica que se quiera recoger, pero siempre dentro del espectro restringido o en cascada. Esta relación además contiene un atributo "N_Armas" que proviene del atributo que posee la interrelación **Suministra**.

La entidad *LÍDER POLIT.* se transforma en una relación, LÍDER, cuya clave primaria es "Cód_Líder" y "Cód_Grupo", este último atributo es además clave ajena que referencia a GRUPO y que por venir de una interrelación en identificación sus opciones de borrado y modificación serán en cascada. También está compuesta por un atributo que no admite valores nulos denominado "Apoyo".

Se crea la relación JEFE que proviene de la transformación de la entidad *JEFE MILITAR*, y por supuesto se transforman también los atributos asociados a la misma. Por lo que "Código" es la clave primaria de la relación, "Rango" un atributo que no admite valores nulos y dos claves ajenas que surgen de la transformación de dos interrelaciones, una con *LÍDER POLIT.* y la otra con *DIVISIÓN*. La clave ajena

sociada a la primera interrelación está compuesta por dos atributos "Cód_Líder" y Grupo" que no admitirán valores nulos y la asociada a la segunda, también está ompuesta por dos atributos, "División" y "C_Grupo" que si que admitirán valores ulos. Las opciones de borrado serán restringidas y las de modificación en cascada.

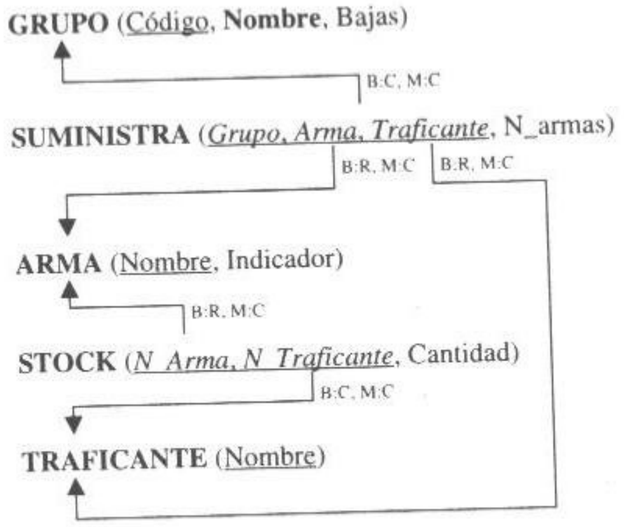


Figura 2.2.31

Puede observarse que en esta relación, JEFE, aparecería dos veces el código de grupo armado, la primera como clave ajena proveniente de DIVISIÓN ("C_Grupo") y la segunda como clave ajena proveniente de LÍDER ("Cód_Grupo"), los valores que tomarán estos atributos serán para la misma tupla (siempre que ambos estén definidos) idénticos, por lo que habría información redundante, sin embargo, es difícil eliminarla si se pretende mantener las dos claves ajenas. Hay que tener en cuenta que no todo jefe militar dirige una división, por lo que "C_Grupo" estaría definido cuando "Cód_Grupo" podría no estarlo. La solución más simple para conseguirlo consiste en dejar en esta relación un único código de grupo armado (clave ajena a LÍDER) y eliminar la clave ajena a DIVISIÓN, claro que habría que añadir algún mecanismo (tipo disparador) que compruebe que cuando el atributo "División" sea distinto de nulo, su valor junto con el de "C_Grupo" aparezca en la relación "División", al tiempo debería implementarse algún mecanismo que supliera las opciones de borrado y modificación de la clave ajena eliminada. Otra solución, mucho más sencilla, consiste en dejarlo como está y admitir esa duplicidad, a lo sumo, podría añadirse una regla de verificación que compruebe que en cualquier tupla de JEFE o bien "C_Grupo" y "Cód_Grupo" son iguales o bien "C_Grupo" es nulo.

Los conflictos también se relacionan con las organizaciones mediadoras. Para representarlo se transforma la entidad *ORGANIZACIÓN* en la relación ORGANIZACIÓN y los atributos que la componen son: "Código", "Nombre" y "Tipo" cuyas restricciones son de clave primaria, de clave alternativa, y de obligatoriedad, respectivamente. Para controlar que el atributo "Tipo" solo tome valores en el dominio (gubernamental, no gubernamental o internacional) se debe

crear una regla de verificación (check) que lo controle. La interrelación N:M que existe entre estas dos entidades se transforma en una relación denominada MEDIA cuya clave primaria se compone de tres atributos, "Conflicto, "Organización" y "F_Ini", y además tiene los atributos "T_Ayuda", "Desplegados" y "F_Fin", donde este último admite valores nulos. Para que el atributo "T_Ayuda" tome solamente valores de su dominio (médica, diplomática o presencial) se creará la correspondiente regla de verificación. Tanto "Conflicto" como "Organización" serán claves ajenas que referencian a las relaciones CONFLICTO y ORGANIZACIÓN, respectivamente, y tendrán como opción de borrado y de modificación en cascada

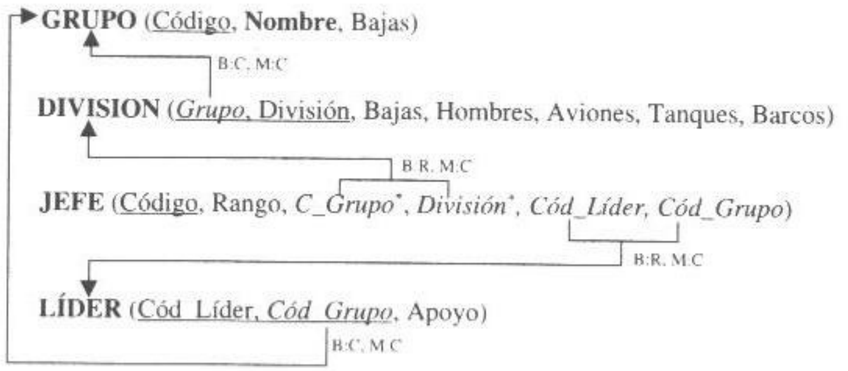


Figura 2.2.32

Para reflejar la interrelación reflexiva **Depende** se creará dentro de ORGANIZACIÓN una clave ajena a sí misma que podrá tomar valores nulos y cuya opción de borrado podría ser puesta a nulos y la de modificación en cascada.

Transformaremos la interrelación **Dialoga** en una relación DIALOGA compuesta por tres atributos, "Grupo", "Líder" y "Organización", que forman la clave primaria, y a su vez los dos primeros son clave ajena que referencia a LÍDER y el otro también es clave ajena que referencia a ORGANIZACIÓN. Ambas claves tendrán como opción de borrado la restringida y de modificación en cascada.

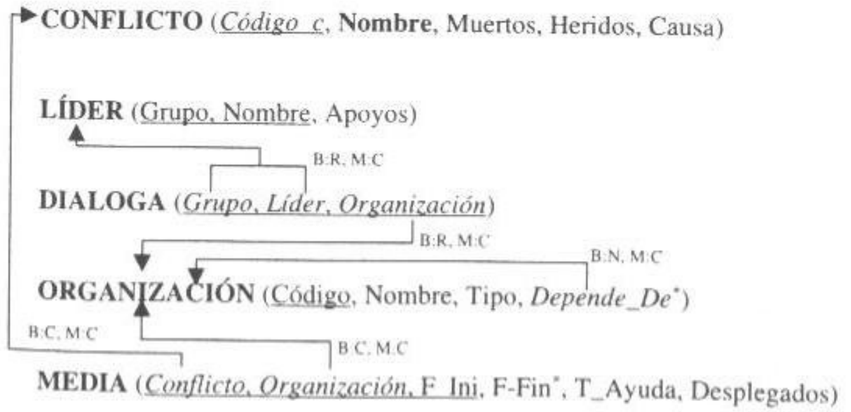


Figura 2.2.33

ropuesta de solución

En la figura 2.2.33 se muestra el grafo relacional con el resultado de la ansformación.

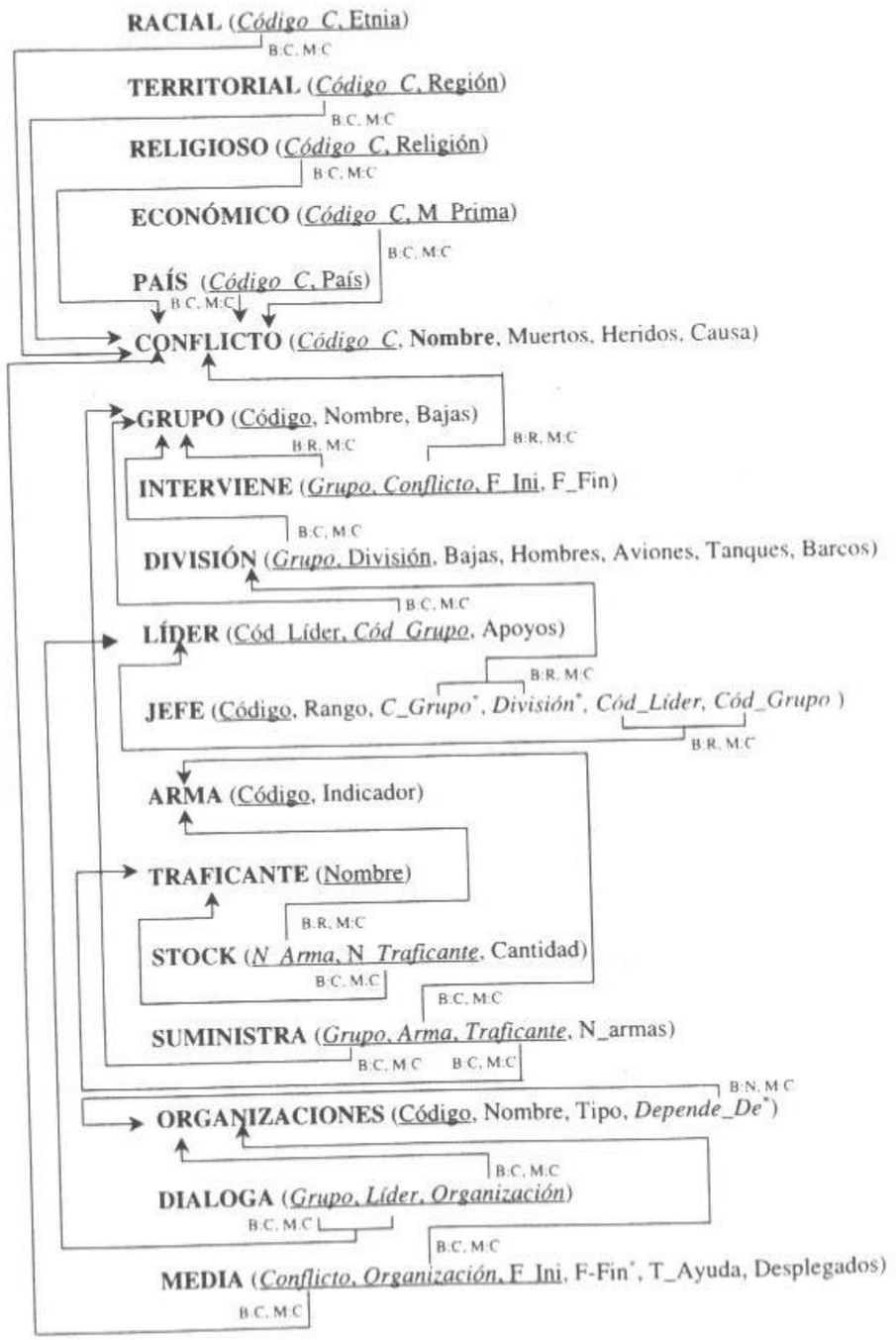


Figura 2.2.34. Esquema relacional solución

Semántica no recogida

Se deben controlar las siguientes restricciones perdidas en la transformación al relacional:

- Exclusividad de la jerarquía de conflictos.
- Cualquier jefe militar obedece al menos a un líder político.
- Una división es dirigida por al menos un jefe militar.
- Una división es dirigida por tres jefes militares como máximo.
- Todo grupo armado dispone de al menos una división.
- Todo grupo armado es liderado por al menos un líder político.
- En un conflicto intervienen como mínimo dos grupos armados.
- En un conflicto intervienen al menos dos grupos armados.
- Cualquier conflicto afecta siempre como poco a un país.

Con un disparador, procedimiento almacenado o dentro del código de los programas, se debería:

- Calcular las bajas totales en cada grupo armado, a partir de las bajas producidas en sus divisiones.
- Generar y asegurar la secuencialidad del número de división dentro del grupo armado.

Por otra parte, en el enunciado original, no hay restricciones que no se hayan podido reflejar en el esquema E/R.

Por último, podría considerarse la introducción de otras restricciones que no figuraban en el problema original, pero que proporcionarían calidad a los datos presentes en la BD:

- Una organización no depende de sí misma ni de ninguna de las organizaciones que de ella dependen.
- Separación entre intervalos temporales de mediación de una organización en el mismo conflicto.
- Separación entre intervalos temporales de intervención de un grupo armado en el mismo conflicto.
- Corrección de los anteriores intervalos.

PROBLEMA 5: GESTIÓN DE NÓMINAS

Enunciado

Transformar el esquema E/R obtenido en el problema 6 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.35 el esquema E/R del que partiremos.

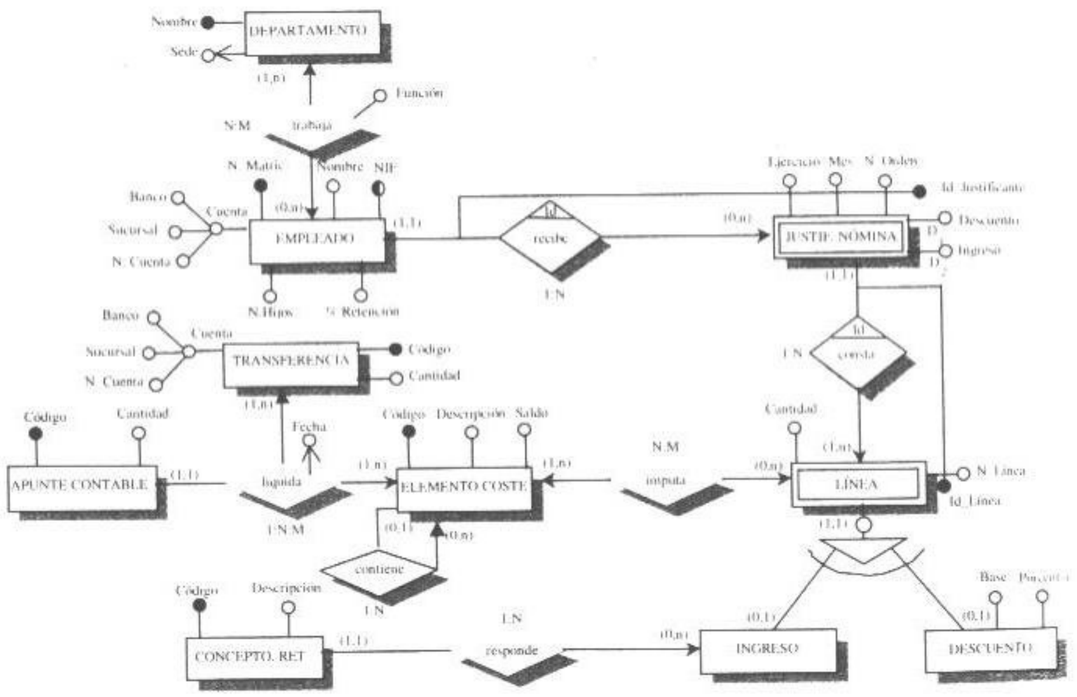


Figura 2.2.35. Esquema E/R a transformar

Comenzamos transformando las entidades. Cada entidad dará lugar a una relación; así, al transformar la entidad *DEPARTAMENTO* se tendrá la relación DEPARTAMENTO con un único atributo "Nombre", el cual será su clave primaria (por tratarse del atributo identificador principal de la entidad); el atributo "Sede" se trata de un atributo multivaluado que, como sabemos, dará lugar a una relación SEDE. esta relación tendrá como atributos "Departamento" (nombre del departamento) y "Sede" y su clave primaria estará compuesta por estos dos atributos (si los departamentos no compartieran sede, "Sede" sería la clave primaria dado que no

podría repetirse; sin embargo, no parece lógico suponer que diferentes departamentos no puedan compartir sede). El atributo "Departamento" es una clave ajena que referencia a la relación DEPARTAMENTO, sus opciones de borrado y modificación serán ambas en cascada, pues la relación SEDE proviene de un atributo multivaluado.

Figura 2.2.36

De transformar la entidad *EMPLEADO* obtendremos la relación EMPLEADO, que tendrá los atributos "N.Mat" (clave primaria de la relación), "NIF" (clave alternativa), "Nombre", "N.Hijos", "Retencion", "Banco", "Sucursal" y "N.Cuenta". Todos estos atributos no admitirán nulos, pues en el esquema E/R se habían considerado como obligatorios.

La interrelación N:M **Trabaja** entre las entidades *DEPARTAMENTO* y *EMPLEADO* se transformará en la relación TRABAJA, la cual tendrá como atributos "Departamento", "N.Mat" y "Función"; su clave primaria estará compuesta por los dos primeros atributos, cada uno de los cuales será a su vez clave ajena a las relaciones DEPARTAMENTO y EMPLEADO, respectivamente. En ambos casos, las opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica que pretendamos recoger. En cualquier caso, habría de asegurarse que la cardinalidad mínima uno (todo empleado trabaja al menos en un departamento) siempre se cumple. Por otra parte, ninguno de los atributos de esta relación admite valores nulos.

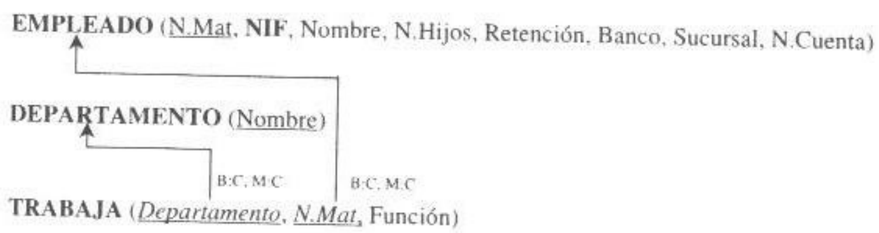


Figura 2.2.37

La entidad *JUSTIFICANTE DE NÓMINA* se transformará en la relación JUSTIFICANTE que tendrá los atributos "N.Mat" (proveniente de la entidad *EMPLEADO*), "Ejercicio", "Mes" y "N.Orden"; los atributos derivados "Descuento" e "Ingreso" podrán considerarse como virtuales y calcularse en cada recuperación, o bien materializarse y recalcularse según se van realizando operaciones sobre el esquema. Optamos por materializarlos, con lo que la relación JUSTIFICANTE tendrá también los dos atributos "Descuento" e "Ingreso" y, como consecuencia habrían de considerarse las reglas de cálculo de los mismos.

La clave primaria de la relación JUSTIFICANTE estará formada por los atributos ercicio", "Mes" y "N.Orden" junto con el atributo "N.Mat" que proviene de la idad *EMPLEADO*, al tratarse la interrelación **Recibe** de una dependencia en ntificación. Este atributo "N.Mat" es una clave ajena que referencia a la relación MPLEADO; las opciones de borrado y modificación serán en cascada o restringida pendiendo de la semántica y el tratamiento que se pretenda realizar.

El atributo "N.Orden" no puede ser nulo, dado que forma parte de la clave maria de la relación JUSTIFICANTE. Se debe asegurar que se inicializará con un lor por defecto (por ej. 1), para evitar que sea nulo, y si un justificante de nómina es único de su mes siempre tendrá número de orden 1. Además, cada vez que un npleado recibiera más de un justificante de nómina en cada mes deberíamos egurarnos, con un disparador, de que el valor que recibe la correspondiente tupla del STIFICANTE se incrementa secuencialmente como se afirmaba en el enunciado iginal.

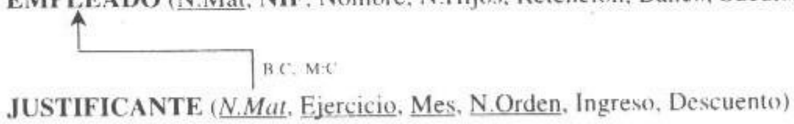


Figura 2.2.38

La transformación de la interrelación 1:N **Recibe** ya se ha realizado mediante ropagación de clave. De la misma manera, se transformará la interrelación 1:N **onsta** aunque, en este caso, la cardinalidad mínima uno (todo justificante de nómina onsta como mínimo de una línea, que además es de ingresos) no se podrá recoger.

La entidad *LÍNEA* se transformará en la relación LÍNEA con atributos "N.Mat", Ejercicio", "Mes", "N.Orden" (todos ellos pro venientes de la propagación de clave de interrelación **Consta**), "N.Línea" (que junto a los anteriores es clave primaria de la elación, dado que se trata de una dependencia en identificación) y "Cantidad". Los uatro primeros atributos son clave ajena que referencia a la relación USTIFICANTE; las opciones de borrado y modificación de la misma serán en ascada o restringida dependiendo de la semántica pretendida.

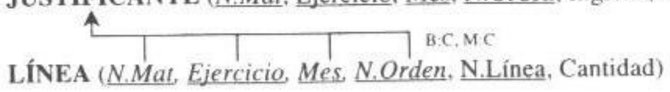


Figura 2.2.39

Con el atributo "N.Línea" en la relación LÍNEA sucede lo mismo que con el tributo "N.Orden" en la relación JUSTIFICANTE, debemos asegurar su iniciación y u secuencialidad.

La entidad *CONCEPTO* se transformará en la relación CONCEPTO, que tendrá or atributos "Código" (clave primaria) y "Descripción". Ninguno de estos atributos dmitirá valores nulos.

Transformaremos la generalización de tipos de línea optando por crear una relación diferente para el supertipo y para cada uno de los subtipos. Obtenemos así la relación DESCUENTO con clave primaria igual que la de la relación LÍNEA y dos atributos "Base" y "Porcentaje" adicionales; ninguno de ellos admite nulos pues así se ha considerado en el esquema E/R. También se obtiene la relación INGRESO con la misma clave primaria y que recibe además por propagación de clave de la interrelación 1:N **Responde** el atributo "Concepto" que será una clave ajena a la relación CONCEPTO cuyas opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica pretendida. El conjunto de atributos que forman la clave primaria, tanto en la relación INGRESO como en la relación DESCUENTO, serán clave ajena a la relación LÍNEA y sus opciones de modificación y borrado serán en cascada o restringidas nuevamente dependiendo de la semántica y tratamiento que se pretendan. En la relación LÍNEA se incluye el atributo "Tipo" que no admite nulos para controlar la totalidad de la jerarquía. La exclusividad de la generalización se ha de forzar con la inclusión de la correspondiente aserción.

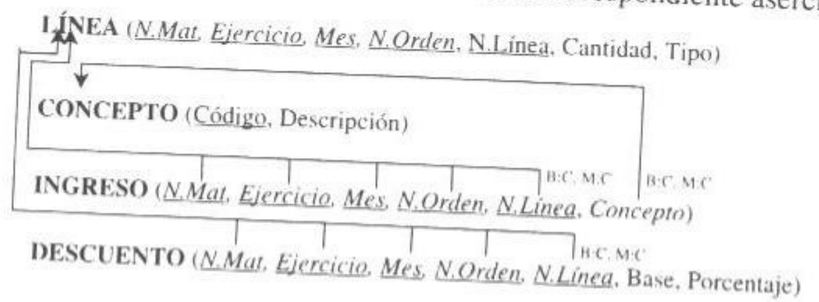


Figura 2.2.40

La entidad *ELEMENTO COSTE* se transforma en la relación COSTE, que tendrá tributos "Código" (clave primaria), "Descripción" y "Saldo"; ninguno de ellos admite alores nulos. La interrelación reflexiva **Contiene** podría transformarse mediante ropagación de clave, reflejando así para cada elemento de coste el posible elemento le coste que le contiene. Optamos por transformarla creando una nueva relación ONTIENE, que contendrá los atributos "Coste" (elemento de coste del que depende) "Coste_D" (elemento de coste hijo); ambos serán claves ajenas a la relación COSTE on opciones de borrado y modificación en cascada o restringidas. Evitaremos el mplear los dos atributos de la relación como clave primaria dado que bastará para dentificar a cada tupla de esta relación el atributo "Coste_D", dado que un hijo sólo ndrá un padre y por tanto aparecerá una única vez como hijo en las tuplas esta lación.

La interrelación N:M **Imputa** se transformará en la relación IMPUTA con los tributos "N.Mat", "Ejercicio", "Mes", "N.Orden", "N.Línea" y "Coste"; todos j untos mponen la clave primaria de la relación y todos salvo "Coste" son clave ajena que ferencia a la relación LÍNEA; el atributo "Coste" es clave ajena que referencia a la ación COSTE. En ambos casos, las opciones de integridad referencial serán en scada o restringidas según la semántica deseada.

La entidad *TRANSFERENCIA* se transforma en la relación TRANSFERENCIA los atributos "Código" (clave primaria), "Banco", "Sucursal", "Cuenta" y

antidad"; ninguno de ellos podrá admitir valores nulos al haber sido considerados mo obligatorios en el esquema E/R.

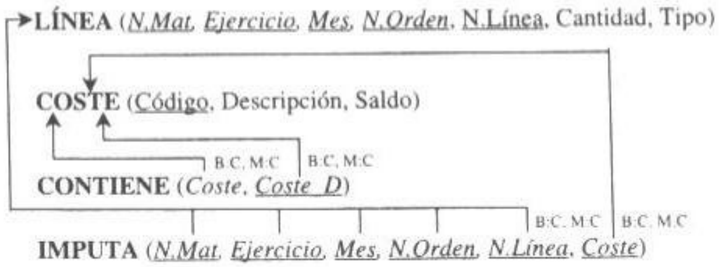


Figura 2.2.41

La entidad *APUNTE CONTABLE* se transforma en la relación APUNTE y sus ributos serán "Código" (clave primaria) y "Cantidad"; ninguno de ellos podrá admitir alores nulos al haber sido considerados como obligatorios en el esquema E/R.

La interrelación ternaria **Líquida** se transformará de acuerdo a lo establecido por caso general de transformación. Sin embargo, esta interrelación tiene un atributo ultivaluado que habrá que considerar y hay una cardinalidad máxima 1 que permitirá mplificar la solución. Se crea así una relación LÍQUIDA que tendrá como atributos s claves ajenas que provienen de cada una de las entidades que asocia esta terrelación, por ejemplo, "Coste", "Transferencia" y "Apunte", tendrá además, un ributo "Fecha". Estas claves ajenas tendrán las opciones de integridad referencial ue corresponda (en cascada o restringidas) dependiendo de la semántica pretendida. a clave primaria de esta interrelación estará formada por "Coste", "Transferencia" y Fecha"; obsérvese cómo no incluye "Apunte" dado que por cada elemento de coste quidado con cargo a una transferencia el apunte contable está determinado cardinalidad 1,1) por lo que no se necesita el apunte para distinguir las tuplas de esta elación; sin embargo, sí se necesita el atributo "Fecha" para distinguir tuplas orrespondientes a liquidaciones iguales aunque realizadas en diferente fecha.

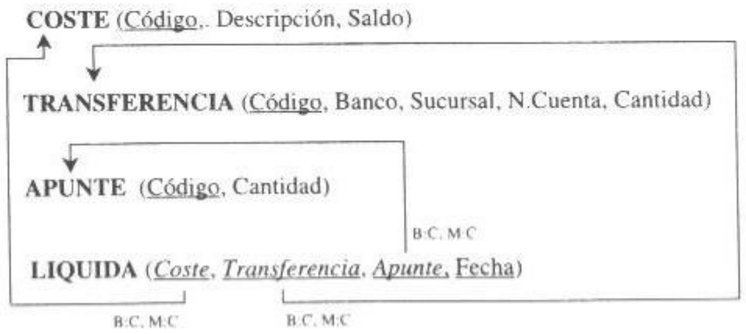


Figura 2.2.42

Propuesta de solución

El resultado de la transformación puede verse en la figura 2.2.43.

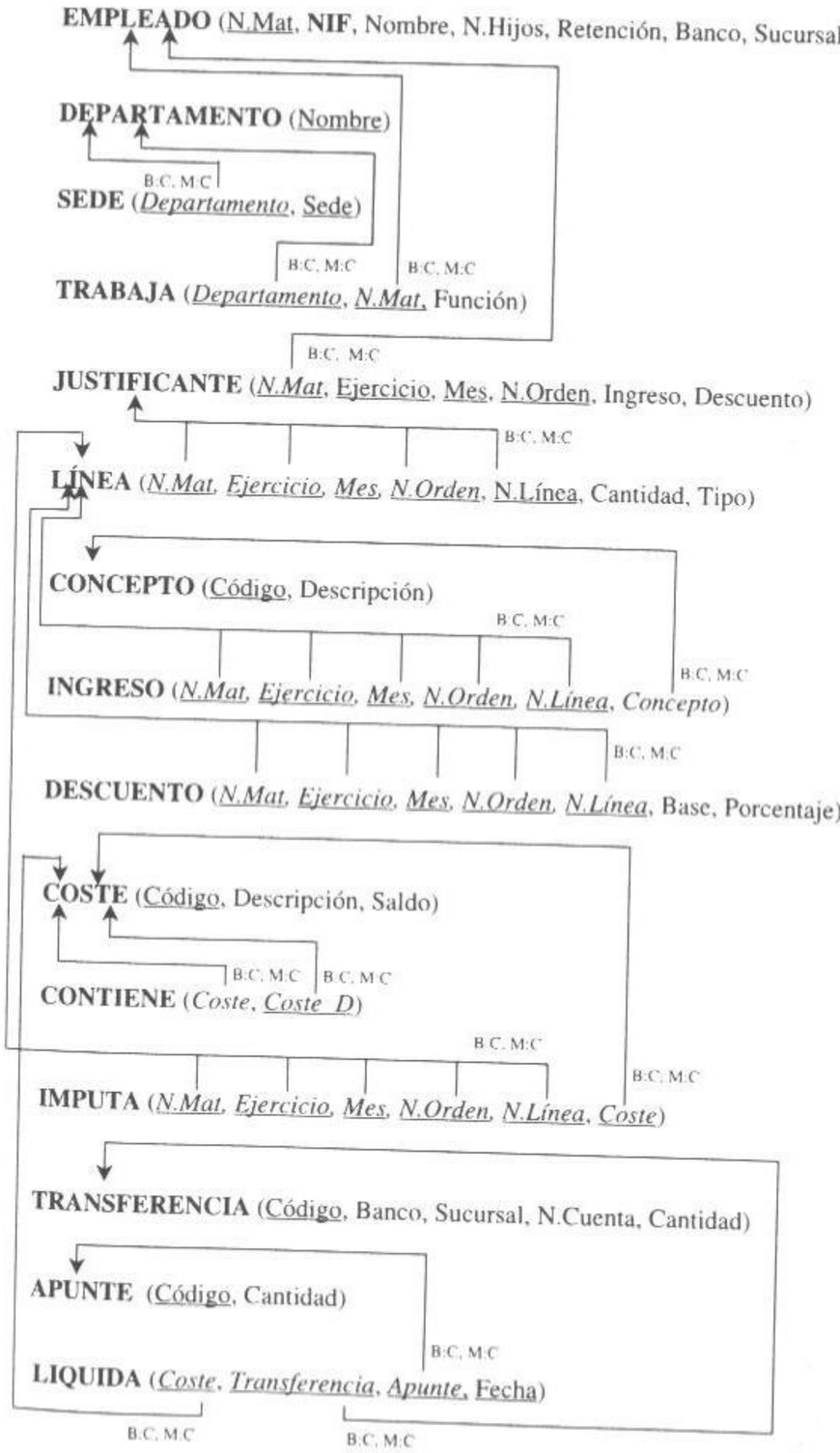


Figura 2.2.43. Esquema relacional solución

Semántica no recogida

No ha sido posible recoger dentro del modelo relacional las siguientes restricciones que sí se recogían en el esquema E/R:

- Exclusividad de la jerarquía de líneas del justificante.
- Cada empleado trabaja en al menos un departamento.
- En cada justificante hay al menos una línea.
- Una línea se imputa al menos a un elemento de coste.
- Todo departamento tiene una sede.

Se deben proporcionar además mecanismos que soporten:

- Reglas de cálculo del atributo derivado "Descuento" en JUSTIFICANTE.
- Reglas de cálculo del atributo derivado "Ingreso" en JUSTIFICANTE.
- Generar la secuencialidad del número de orden del justificante.
- Generar la secuencialidad del número de línea del justificante.

Habría que recoger, además, bien mediante disparadores, aserciones o codificándola en las aplicaciones que accedan a la BD, la semántica que, estando presente en el enunciado, no pudo recogerse ni en el esquema E/R ni en el modelo relacional:

- Cada empleado recibe al menos un justificante de nómina cada mes.
- Existe al menos una línea de ingresos en cada justificante.
- Un empleado no puede recibir nóminas tras la extinción de su vida laboral.
- La cantidad en una línea de ingreso es positiva.
- La cantidad en una línea de descuento es negativa.

Por último, se consideraría cualquier otra restricción que se considere apropiada:

- Un elemento de coste no se contiene a sí mismo ni está contenido en ninguno de sus descendientes.

PROBLEMA 6: ENTORNO DE EJECUCIÓN

Enunciado

Transformar el esquema E/R obtenido en el problema 7 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que se recogía en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.44 el esquema E/R del que partiremos.

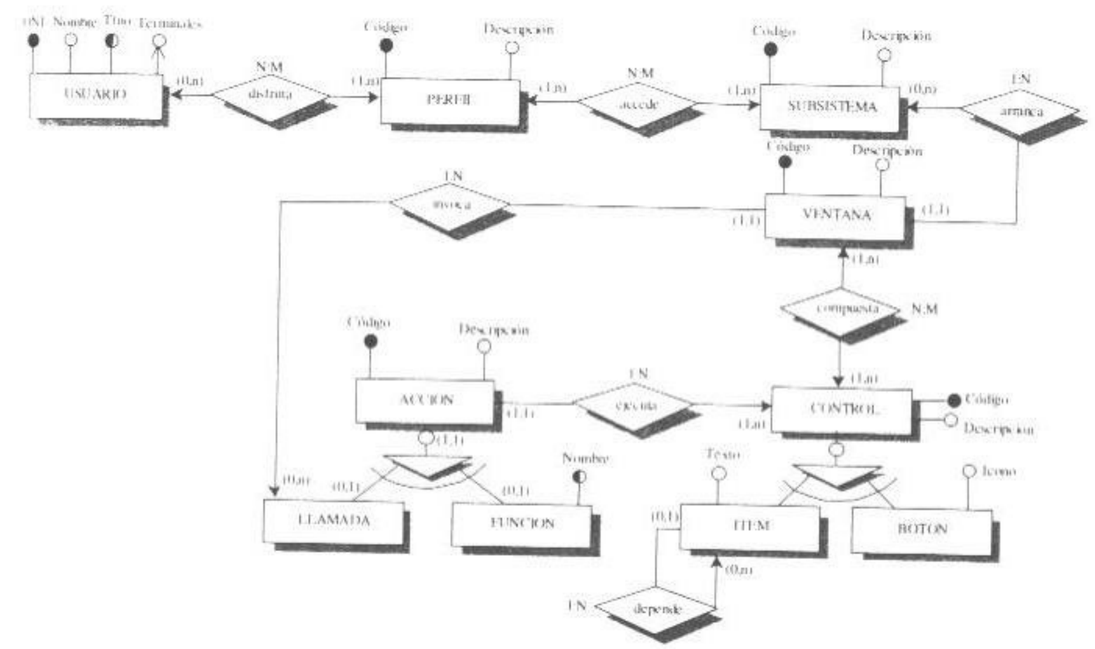


Figura 2.2.44. Esquema E/R a transformar

Comenzamos transformando la entidad *USUARIO* en la relación USUARIO con los atributos "DNI" (clave primaria, por tratarse del atributo identificador principal de la entidad), "Nombre" y "Teléfono" (clave alternativa) tal que ninguno de ellos admite valores nulos. El atributo multivaluado "Terminales" dará lugar a una relación TERMINAL que tendrá los atributos "DNI" y "Terminal" como clave primaria. El atributo "DNI" se trata de una clave ajena que referencia a la relación USUARIO y sus opciones de borrado y modificación serán ambas en cascada, pues la relación TERMINAL proviene de un atributo multivaluado.

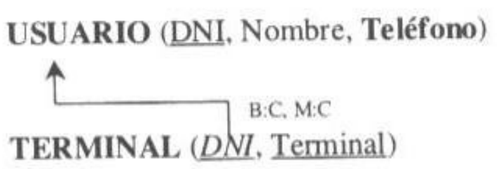


Figura 2.2.45

De la transformación de la entidad *PERFIL* se obtiene la relación PERFIL, que tendrá los atributos "Perfil" (clave primaria de la relación) y "Descripción". La interrelación N:M **Disfruta** entre las entidades *USUARIO* y *PERFIL* se transformará en la relación DISFRUTA, la cual tendrá como atributos "DNI" y "Perfil" que compondrán la clave primaria y cada uno de ellos será a su vez clave ajena a las relaciones USUARIO y PERFIL, respectivamente. En ambos casos, las opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica que pretendamos recoger. En cualquier caso, habría de asegurarse que la cardinalidad mínima uno (todo usuario disfruta al menos de un perfil) que no puede recogerse siempre se cumplirá.

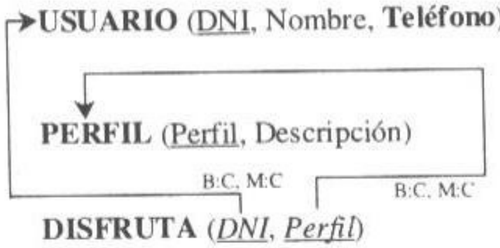


Figura 2.2.46

La entidad *SUBSISTEMA* se transformará en la relación SUBSISTEMA conteniendo los atributos "Subsistema" (clave primaria), "Descripción" y "Ventana" (proveniente de la interrelación **Arranca** con la entidad *VENTANA*). La cardinalidad mínima 1 que refleja que todo subsistema arranca en una ventana impedirá que este atributo pueda tomar valores nulos. Este atributo "Ventana" es una clave ajena que referencia a la relación VENTANA (aún sin crear). Las opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica y el tratamiento que se pretenda realizar.

La interrelación N:M **Accede** entre las entidades *SUBSISTEMA* y *PERFIL* se transformará en la relación ACCEDE, la cual tendrá como atributos "Perfil" y "Subsistema" que juntos compondrán la clave primaria; cada uno de ellos será a su vez clave ajena a las relaciones PERFIL y SUBSISTEMA, respectivamente. En ambos

casos, las opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica que pretendamos recoger. Además, deberíamos asegurarnos de alguna manera que las cardinalidades mínimas uno (todo perfil accede al menos a un subsistema y todo subsistema accede al menos a un perfil) siempre se cumplen.

La entidad *VENTANA* se transformará en la relación VENTANA, que tendrá por atributos "Ventana" (clave primaria) y "Descripción". Ninguno de estos atributos admitirá valores nulos. La transformación de la interrelación 1:N **Arranca** ya se ha realizado mediante propagación de clave.

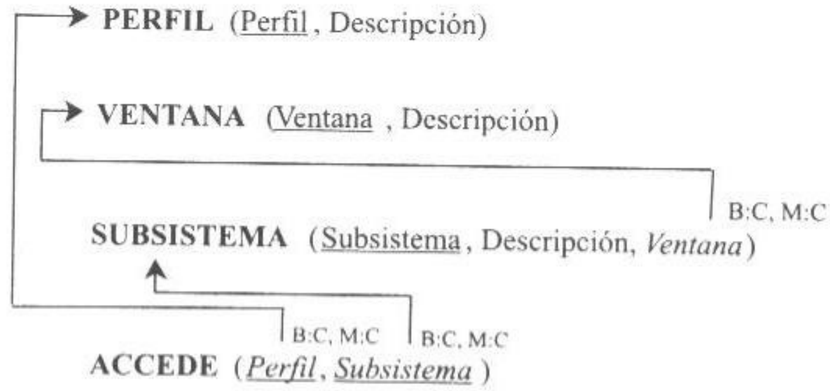


Figura 2.2.47

La entidad *ACCIÓN* se transformará en la relación ACCIÓN que tendrá como atributos "Acción" (clave primaria) y "Descripción" (ninguno de estos atributos admitirá valores nulos al no haberse considerado como opcionales en el esquema E/R).

Transformaremos las dos generalizaciones del esquema E/R optando por crear una relación diferente para el supertipo y para cada uno de los subtipos. Así, obtenemos para la jerarquía de acciones las relaciones LLAMADA y FUNCIÓN, ambas con clave primaria igual que la de la relación ACCIÓN. La relación FUNCIÓN tendrá, además, el atributo adicional "Nombre" (clave alternativa), mientras que la relación LLAMADA tendrá además el atributo "Ventana" (clave ajena que referencia a la relación VENTANA, resultado de la propagación de clave de la interrelación 1:N **Invoca**). Todas las claves ajenas en estas relaciones tendrán sus opciones de modificación y borrado en cascada o restringidas dependiendo de la semántica y tratamiento que se pretendan.

La totalidad de la generalización se controla con el atributo discriminante "Tipo" en la relación ACCIÓN mientras que la exclusividad se controlaría con una aserción.

La entidad *CONTROL* se transformará en la relación CONTROL con sus atributos "Control" (clave primaria), "Acción" (clave ajena que referencia a la relación ACCIÓN, resultado de la propagación de clave de la interrelación 1:N **Ejecuta**) y "Descripción". Las opciones de borrado y modificación de la clave ajena "Acción" serán en cascada o restringidas dependiendo de la semántica pretendida. Ninguno de los atributos admitirá nulos y la cardinalidad mínima 1 (toda acción la ejecuta al menos un control) no se podría reflejar.

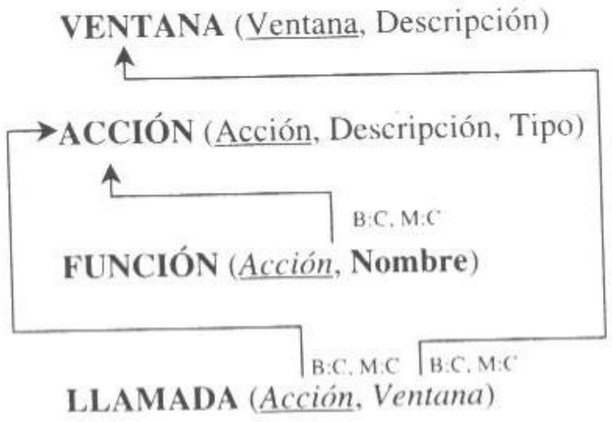


Figura 2.2.48

La generalización de controles se realizará de manera similar a la jerarquía de acciones. De esta forma se tienen las relaciones ÍTEM y BOTÓN, ambas con clave primaria igual que la de la relación CONTROL. La relación BOTÓN tendrá, además, el atributo "Icono" que admitirá valores nulos, mientras que la relación ÍTEM tendrá el atributo "Texto" y el atributo "Depende" (clave ajena, resultado de la propagación de clave de la interrelación reflexiva 1:N **Depende**, que referencia a la propia relación ÍTEM y que permite recoger para cada ítem el posible ítem padre del que depende) que admitirá valores nulos. Nuevamente, todas las claves ajenas en estas relaciones tendrán sus opciones de modificación y borrado en cascada o restringidas, dependiendo de la semántica y tratamiento que se pretendan.

La totalidad de la generalización se controla con el atributo discriminante "Tipo" en la relación CONTROL mientras que la exclusividad se controlaría con la correspondiente aserción.

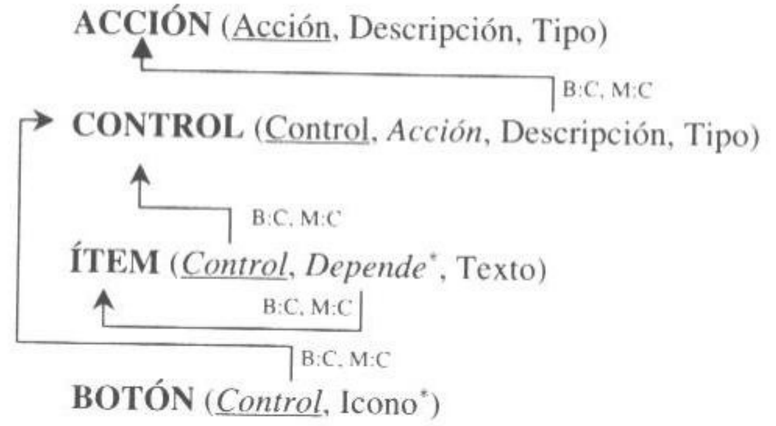


Figura 2.2.49

Por último, falta transformar la interrelación N:M **Compuesta** entre las entidades *VENTANA* y *CONTROL* que dará lugar a la relación COMPONE, con los atributos "Ventana" y "Control", que forman la clave primaria; cada uno de ellos será respectivamente clave ajena a las relaciones VENTANA y CONTROL. En ambos casos, las opciones de borrado y modificación serán en cascada o restringidas dependiendo de la semántica que pretendamos recoger. Por otro lado, habría de asegurarse que las cardinalidades mínimas uno (toda ventana está compuesta al menos por un control y todo control pertenece al menos a una ventana) siempre se cumplen, dado que en el esquema relacional no pueden ser reflejadas.

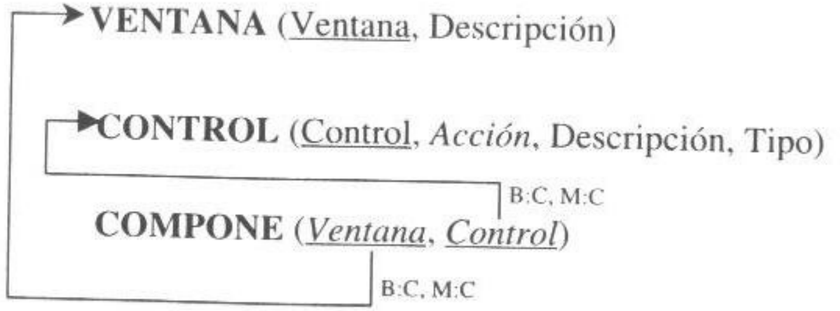


Figura 2.2.50

Propuesta de solución

El resultado de la transformación puede verse en la figura 2.2.51.

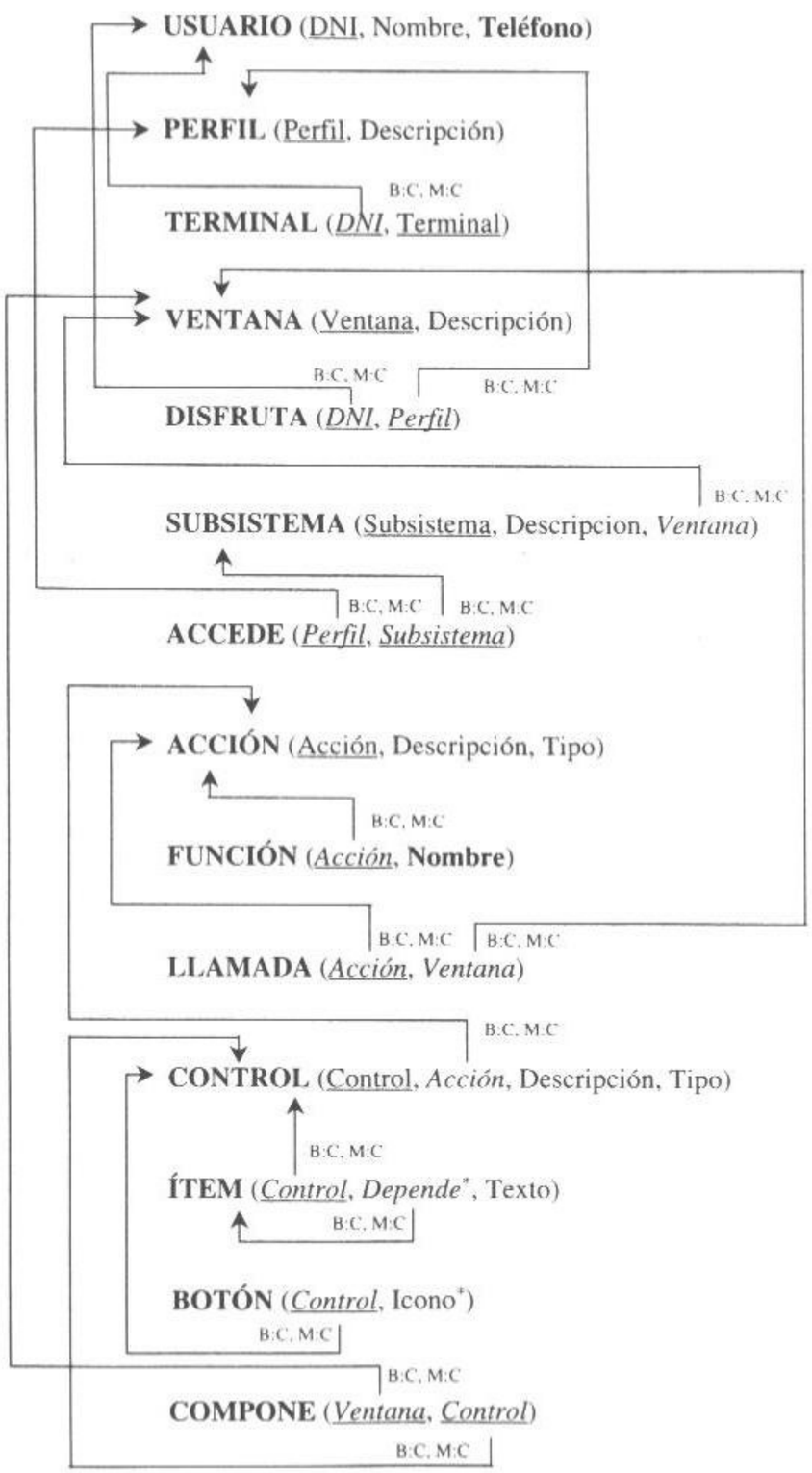


Figura 2.2.51. Esquema relacional solución

Semántica no recogida

No ha sido posible recoger dentro del modelo relacional la siguiente semántica que sí se recogía en el esquema E/R:

- Exclusividad de la jerarquía de acciones.
- Exclusividad de la jerarquía de controles.
- Un usuario disfruta al menos de un perfil de acceso.
- Un perfil permite el acceso al menos a un subsistema.
- En todo subsistema hay al menos un perfil para poder accederle.
- Una ventana esta compuesta por al menos un control.
- Un control pertenece al menos a una ventana.
- Toda acción es ejecutada por al menos un control.
- Todo usuario trabaja al menos en un terminal.

Habría que recoger, además, la semántica que, estando presente en el enunciado, no ha podido recogerse ni en el esquema E/R ni en el modelo relacional:

- Toda ventana tiene un control cuya acción asociada es el cierre de la misma.

Por último, presentamos aquellas restricciones que consideramos lógico o adecuado introducir:

- Un ítem de menú no depende de sí mismo ni de ninguno de sus descendientes.

PROBLEMA 7: ORGANIZACIONES NO GUBERNAMENTALES

Enunciado

Transformar el esquema E/R obtenido en el problema 8 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.52 el esquema E/R del que partiremos.

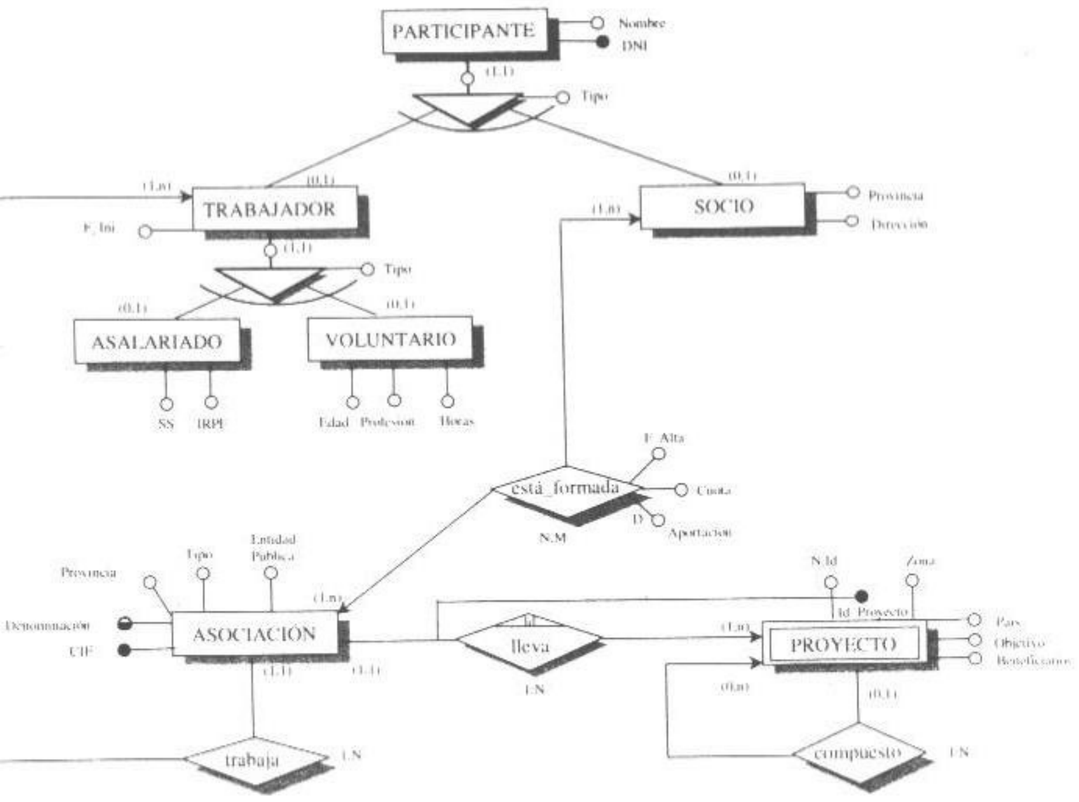


Figura 2.2.52. Esquema E/R a transformar

Comenzamos realizando la transformación de las jerarquías asociadas a la entidad *PARTICIPANTE*. De las diversas alternativas para la transformación de jerarquías, optamos por crear una relación para cada una de las entidades participantes. Se obtienen, así, las siguientes relaciones, las cuales comparten la clave primaria "DNI" y tienen los mismos atributos que poseían las correspondientes entidades, ninguno de ellos admite nulos.

Figura 2.2.53

Se incluye el atributo discriminante "Tipo", en las relaciones PARTICIPANTE y TRABAJADOR que, aunque no son estrictamente necesarios, permiten un control más sencillo de la totalidad y exclusividad de las jerarquías.

Salvo en PARTICIPANTE, en las demás relaciones, el atributo "DNI" es clave ajena que referencia a PARTICIPANTE con las opciones de borrado y modificación en cascada.

Una alternativa de transformar estas jerarquías que reduciría el número de relaciones, sería fundir estas cinco relaciones en una sola, empleando un único atributo discriminante "Tipo" que indique a cuÁl de las categorías pertenece cada una de las tuplas de esa relación. Sin embargo, la presencia de atributos e interrelaciones propias en las distintas entidades participantes en la jerarquía (incluso en el nivel intermedio) y el hecho de que en esta solución habría una mayor cantidad de nulos nos ha hecho decantarnos por la solución expuesta que al tiempo refleja de una forma más fiel la semántica del problema.

Transformamos a continuación la entidad *ASOCIACIÓN* y la interrelación **Trabaja**, que, al tratarse de una interrelación 1:N, introducirá una clave ajena, "Asociación", en la relación TRABAJADOR. Esta clave ajena tendrá opciones de borrado y modificación en cascada, pues se cree que los trabajadores de una asociación dejarán de interesarnos cuando ésta desaparezca. La clave primaria de la relación ASOCIACIÓN es el atributo "CIF", y el atributo "Denominación" será clave alternativa. El resto de los atributos de la entidad se pondrían también en la relación. Ninguno de los atributos de esta relación admite valores nulos.

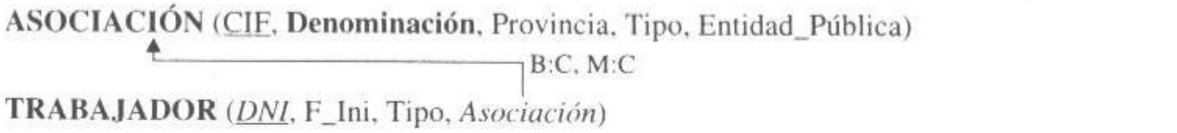


Figura 2.2.54

La interrelación **Está_Formada** entre las entidades *ASOCIACIÓN* y *SOCIO* es N:M y posee atributos propios, su transformación proporcionará una relación FORMADA, cuya clave primaria son las dos claves ajenas "CIF" y "DNI" que respectivamente referencian a las relaciones ASOCIACIÓN y SOCIO. En ambos casos, las opciones de borrado y modificación serán restringidas o en cascada según la semántica que se pretenda. Los atributos propios de esta interrelación serán atributos de la nueva relación que no admitirán valores nulos.

Transformamos por último la entidad *PROYECTO* que está afectada por dos interrelaciones 1:N, la interrelación **Lleva** que es una dependencia en identificación, y la interrelación **Compuesto**, que es reflexiva. Se obtendrá así la relación PROYECTO,

ue tendrá en principio los atributos "N_Id", "Zona", "País", "Objetivo" y Beneficiario".

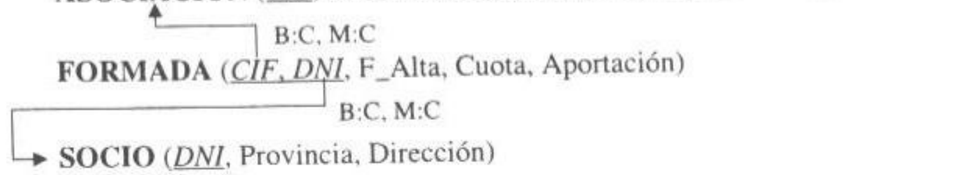


Figura 2.2.55

Al ser **Lleva** una dependencia en identificación, propagará la clave de ASOCIACIÓN a PROYECTO como parte de su clave primaria, con lo que la clave primaria de PROYECTO estará formada por los atributos "Asociación" y "N_Id". El atributo "Asociación" es una clave ajena que referencia a la relación ASOCIACIÓN y para evitar que al borrarse una asociación se eliminen los proyectos que ésta lleva su opción de borrado será restringida o por defecto (no podría ser nula pues el atributo no admite valores nulos), aunque lo habitual en las dependencias en identificación es poner la opción de borrado en cascada, pensando en la seguridad, se ha optado por ponerla restringido.

La interrelación reflexiva **Compuesto** se transformará mediante propagación de clave, con dos nuevos atributos, "A_Padre" y "P_Padre" en la relación PROYECTO. que compondrán la referencia al proyecto padre y que serán por tanto clave ajena a la propia relación. Si se hubiera supuesto que los subproyectos deben llevarlos la misma asociación que lleva el proyecto padre, habría bastado con un único atributo. Dada la cardinalidad mínima cero (un proyecto puede ser componente de otros), este atributo admitirá valores nulos. Se considera que no se debería permitir borrar un proyecto mientras éste tenga subproyectos, por lo que la opción de borrado será restringida y la de modificación en cascada.

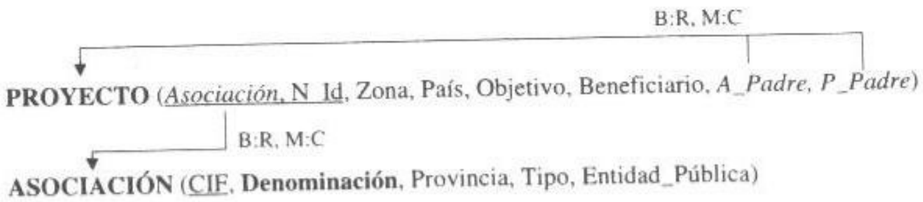


Figura 2.2.56

Propuesta de solución

En la figura 2.2.57 se muestra el grafo relacional con el resultado de la transformación.

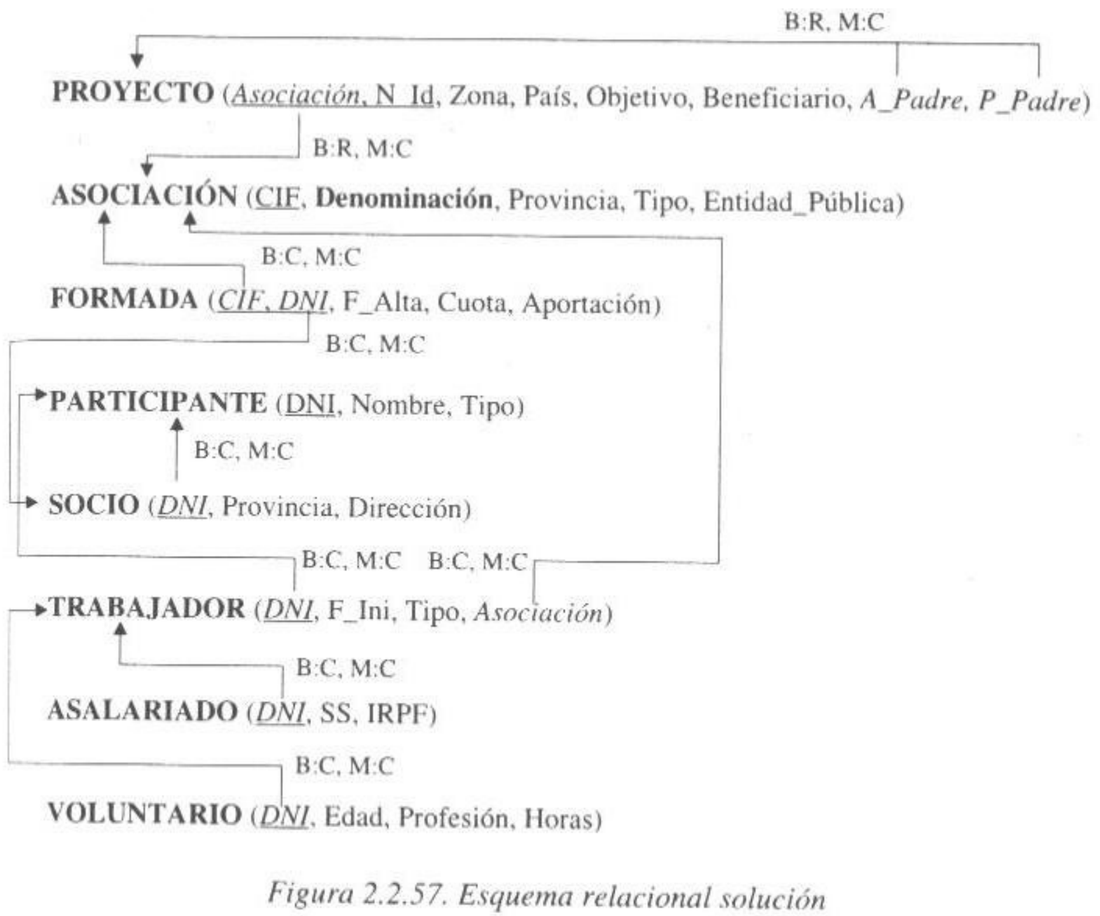


Figura 2.2.57. Esquema relacional solución

Semántica no recogida

No ha sido posible recoger dentro del modelo relacional las siguientes restricciones que sí se recogían en el esquema E/R:

- Exclusividad de la jerarquía de participantes.
- Exclusividad de la jerarquía de trabajadores.
- En una asociación trabaja como mínimo un trabajador.
- Una asociación está formada por al menos un socio.
- Todo socio forma parte de al menos una asociación.
- Toda asociación lleva al menos un proyecto.

Se deben proporcionar además mecanismos que soporten:

- Reglas de cálculo del atributo derivado "Aportación" en la relación FORMADA.

Y por supuesto, medios para recoger, además, la semántica que, estando presente el enunciado, no ha podido recogerse ni en el esquema E/R ni en el modelo acional:

Los subproyectos de un mismo proyecto se realizan en el mismo país en el que se realiza el proyecto.

Todos los proyectos asignados a un trabajador sean del mismo país.

Un socio de una asociación no pueda trabajar en otra.

Por último, se consideraría cualquier otra restricción que se considere apropiada:

Un proyecto no depende de sí mismo ni de ninguno de sus descendientes.

PROBLEMA 8: ADMINISTRACIÓN DE FINCAS

Enunciado

Transformar el esquema E/R obtenido en el problema 9 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

En la figura 2.2.58 se muestra el esquema E/R que transformaremos al modelo relacional.

Se irá realizando la transformación del esquema de forma gradual para presentar al final el grafo relacional completo. En primer lugar, cada una de las entidades del esquema E/R dará lugar a una relación en el modelo relacional. Así, la entidad *ADMINISTRADOR* se transforma en la relación ADMINISTRADOR cuya clave primaria es el atributo "Num_Colegiado", el atributo "DNI" es clave alternativa y "Nombre" es un atributo que no permite nulos. La entidad *COMUNIDAD DE VECINOS* da lugar a la relación COMUNIDAD DE VECINOS con "Cod_Comunidad" como clave primaria y "Nombre", "Calle", "Cód_Postal" y "Población" como atributos descriptivos.

La interrelación 1:N **Gestiona** se transforma en la relación GESTIONA que contiene los atributos "Num_Colegiado" procedente de la entidad *ADMINISTRADOR*, "Cod_Comunidad" procedente de *COMUNIDAD DE VECINOS* y, por último, el atributo de la interrelación "Honorarios". La clave la forma el atributo "Cod_Comunidad", ya que una determinada comunidad de vecinos determina de

forma unívoca el administrador que la gestiona (interrelación 1:N). Las opciones de borrado y modificación pueden ser en cascada o restringidas según convenga (aquí se ha optado por borrado y modificación en cascada). A continuación se muestra el grafo correspondiente a esta transformación (la relación COMUNIDAD DE VECINOS aparece con puntos suspensivos porque todavía le faltan algunos atributos que se añadirán en las siguiente transformaciones).

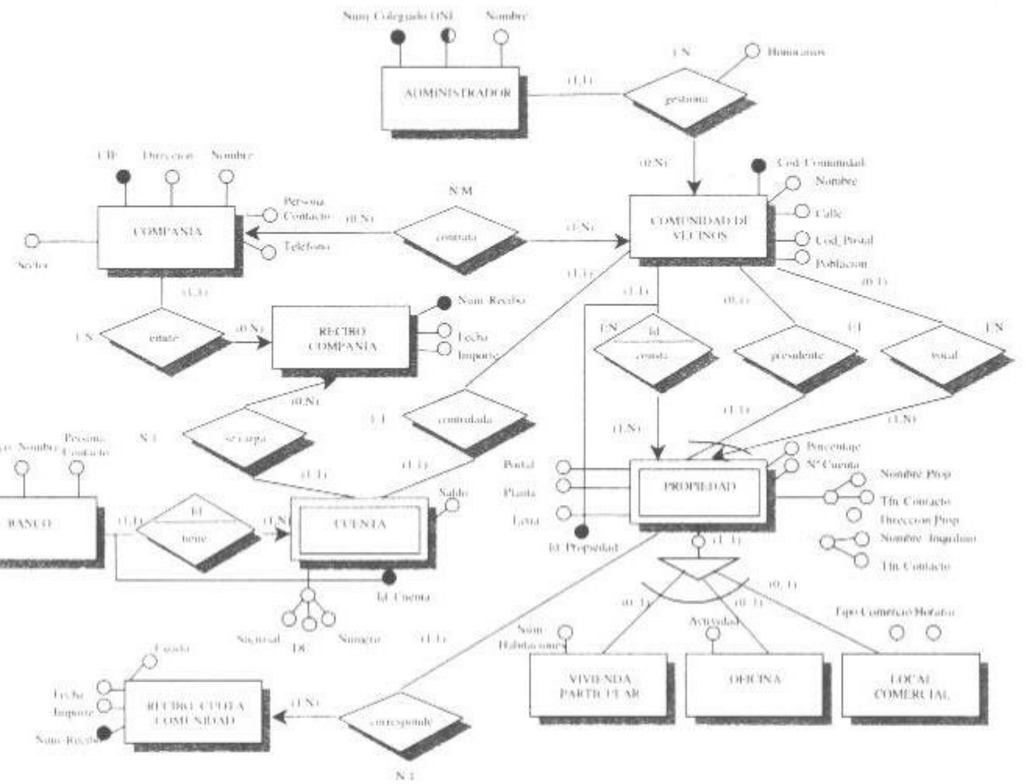


Figura 2.2.58. Esquema E/R a transformar

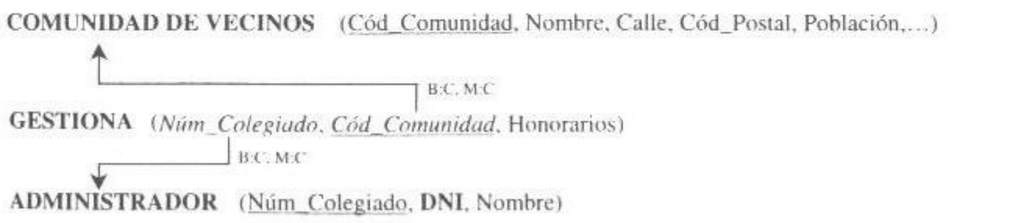


Figura 2.2.59

La entidad *COMPAÑÍA* da lugar a la relación COMPAÑÍA con los atributos "CIF" (clave primaria de la relación), "Nombre", "Dirección", "Sector", "Teléfono" y "Persona_Contacto". La interrelación N:M **Contrata** entre las entidades *COMPAÑÍA* y *COMUNIDAD DE VECINOS* se transforma en la relación CONTRATA con los

atributos "Cod_Comunidad" y "CIF"; su clave primaria es la concatenación de estos dos atributos y las opciones de borrado y modificación pueden ser en cascada o restringidas según interese. El grafo quedaría como se muestra a continuación.

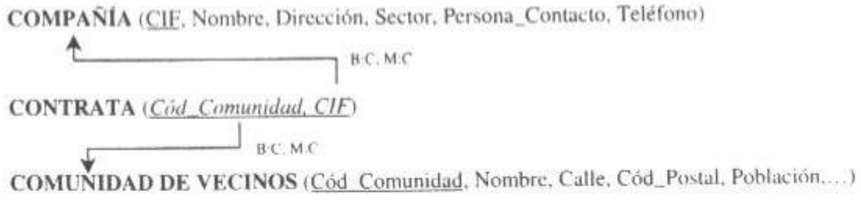


Figura 2.2.60

Veamos a continuación la transformación de la entidad *PROPIEDAD*; puesto que existe una dependencia en identificación (interrelación **Consta**) con la entidad *COMUNIDAD DE VECINOS* se transformará en la relación PROPIEDAD con la concatenación de los atributos "Cod_Comunidad", "Portal", "Planta" y "Letra" como clave primaria, siendo "Cod_Comunidad" clave ajena que referencia la relación COMUNIDAD DE VECINOS (opciones de borrado y modificación obligatoriamente en cascada). Además, la relación PROPIEDAD contiene los atributos "Nombre Propietario", "Tlf_Contacto" (ambos como no nulos) y "Dirección" (que admite valores nulos); en cuanto a los datos del inquilino, se tienen los atributos "Nombre_Inq" y "Tlf_Contacto_Inq" que admiten nulos (pues una determinada propiedad puede que no tenga inquilino). Por último, la relación PROPIEDAD incluye los atributos "N_Cuenta" y "Porcentaje" que no admiten valores nulos.

Respecto a la jerarquía de propiedades, cada subtipo se transformará en una relación mediante la propagación de clave del supertipo. Así, se tendrán las relaciones VIVIENDA PARTICULAR, OFICINA y LOCAL COMERCIAL. Todas ellas tienen como clave la concatenación de "Cod_Comunidad", "Portal", "Planta" y "Letra" (todos ellos forman la clave ajena que referencia la relación PROPIEDAD). Además, la relación VIVIENDA PARTICULAR tendrá el atributo "Num_Habitaciones", OFICINA tendrá el atributo "Actividad" y LOCAL COMERCIAL tendrá los atributos "Tipo_Comercio" y "Horario" que pueden admitir valores nulos. Las opciones de borrado y modificación serán en cascada (cuando se elimina/modifica una determinada propiedad hay que eliminarla/modificarla en el subtipo correspondiente). En cuanto a la totalidad y exclusividad de la jerarquía se controlarán por medio de CHECKS o aserciones así como por disparadores o procedimientos almacenados. También podría incluirse (en este caso no se ha hecho) en la relación PROPIEDAD un atributo discriminante ("Tipo_Propiedad") que indique si una propiedad es vivienda particular, oficina o local comercial que nos ayude a controlar la totalidad de la jerarquía. A continuación se muestra el grafo relacional correspondiente a este conjunto de transformaciones.

Para completar los atributos de la relación COMUNIDAD DE VECINOS vamos a transformar las interrelaciones **Presidente**, **Vocal** y **Controlada**. La interrelación

1:1 **Presidente** se transforma mediante propagación de clave, es decir, la clave de PROPIEDAD pasa a la relación COMUNIDAD DE VECINOS con el nombre de "Presidente" (no admitiendo nulos, pues una determinada comunidad de vecinos siempre tiene que tener un presidente). Obsérvese que por simplificar el grafo resultante sólo se ha propagado como atributo compuesto, realmente, habría que desagregar en todos sus atributos componentes; las opciones de borrado y modificación pueden ser restringidas o puesta a valor por defecto según convenga; la interrelación 1:N **Vocal** la transformaremos en una nueva relación denominada VOCAL que constará de los atributos "Cod_Comunidad", "Portal", "Planta" y "Letra", todos ellos formando la clave primaria de la relación. Por último, la interrelación 1:1 **Controlada** entre las entidades *CUENTA* y *COMUNIDAD DE VECINOS* se transformará también mediante propagación de clave. Para ello, tenemos que transformar primero la entidad *CUENTA*.

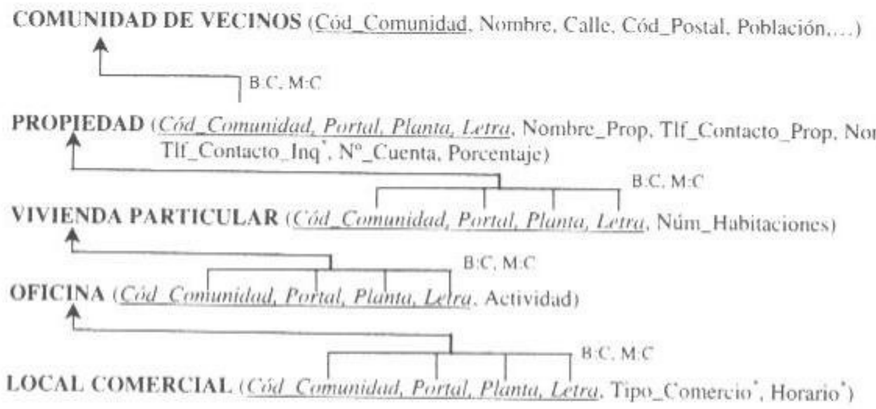


Figura 2.2.61

La entidad débil *CUENTA* depende en identificación de la entidad *BANCO*. Transformamos en primer lugar la entidad *BANCO* en la relación BANCO con los atributos "Cod_Banco" (que es clave primaria), "Nombre" y "Persona_Contacto". La relación CUENTA consta de los atributos "Sucursal", "DC" (dígito de control), "Número", "Cod_Banco" (clave ajena que referencia la relación BANCO con opciones de borrado y modificación en cascada) y "Saldo"; la clave primaria de está relación es la concatenación de "Cod_Banco", "Sucursal", "DC" y "Número".

En este momento ya es posible transformar la interrelación **Controlada** propagando la clave de la relación CUENTA (que no puede tomar valores nulos) a la relación COMUNIDAD DE VECINOS (con opciones de borrado y modificación restringidas o puestas a valor por defecto, según interese) con lo que esta relación ya estaría completa, por simplificar, al igual que hicimos con "Presidente", "Cuenta" se ha propagado como atributo compuesto sin desagregarle en sus componentes.

El grafo relacional correspondiente a este último grupo de transformaciones se muestra a continuación.

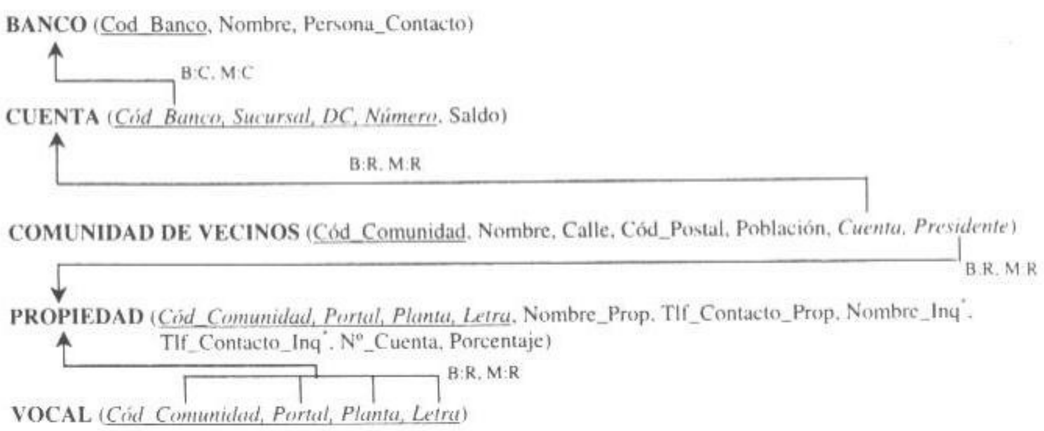


Figura 2.2.62

Las últimas entidades que hay que transformar son *RECIBO COMPAÑÍA* y *RECIBO CUOTA COMUNIDAD*. Vamos, en primer lugar, con la entidad *RECIBO COMPAÑÍA* que se transforma en la relación RECIBO COMPAÑÍA con los atributos "Num_Recibo" (clave primaria), "Fecha" e "Importe (no admiten valores nulos). La interrelación 1:N **Emite** entre *COMPAÑÍA* y *RECIBO COMPAÑÍA* da lugar a la propagación de clave desde la relación COMPAÑÍA a RECIBO COMPAÑÍA (atributo que no puede tomar valores nulos por la cardinalidad mínima de 1). Similar transformación sufre la interrelación 1:N **Se Carga** entre *RECIBO COMPAÑÍA* y *CUENTA* en la que la clave de la relación CUENTA pasa a la relación RECIBO COMPAÑÍA como clave ajena. Las opciones de borrado y modificación para ambos casos pueden ser en cascada, restringidos o incluso con puesta a valor por defecto, según la semántica que se quiera representar. El grafo correspondiente a estas relaciones se muestra a continuación.

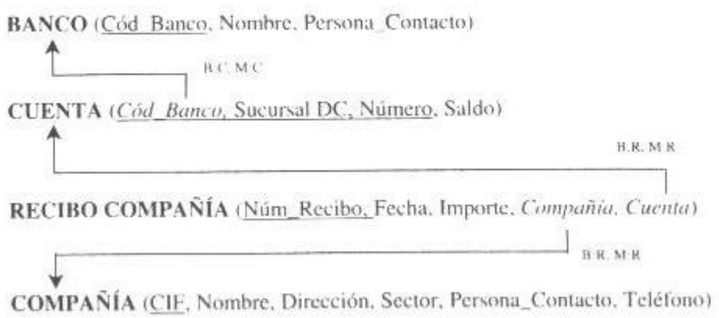


Figura 2.2.63

Por último, la entidad *RECIBO CUOTA COMUNIDAD* se transforma en la relación RECIBO CUOTA COMUNIDAD con los atributos "Num_Recibo", "Fecha", "Importe" y "Saldo". La interrelación 1:N **Corresponde** se transforma mediante propagación de clave desde la relación PROPIEDAD a la relación RECIBO CUOTA

COMUNIDAD con el nombre de "Propiedad" (clave ajena que no admite valores nulos), como se muestra a continuación.

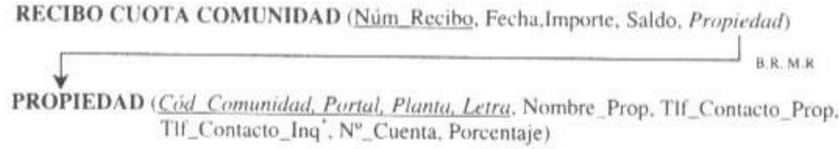


Figura 2.2.64

Propuesta de solución

El grafo relacional completo se muestra en la figura 2.2.65.

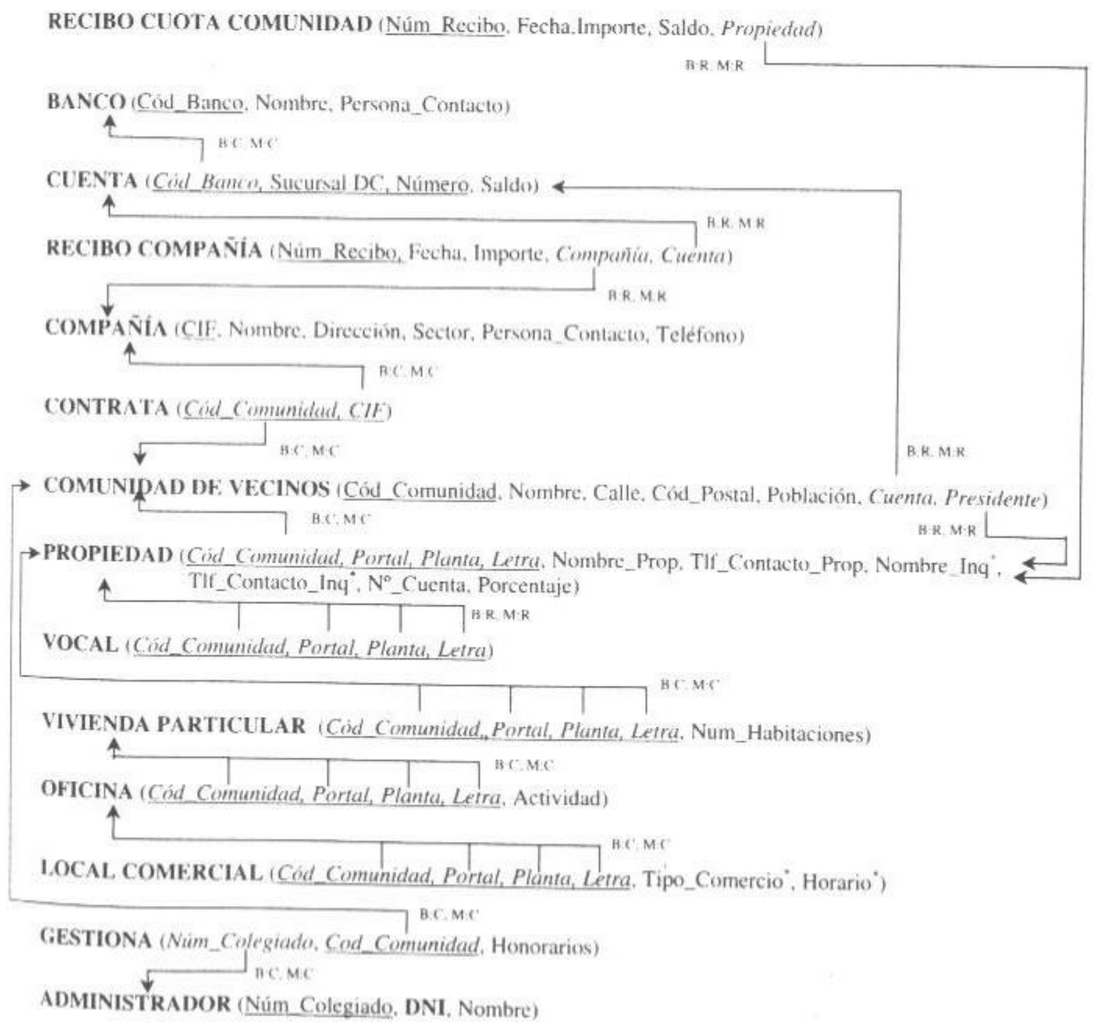


Figura 2.2.65. Esquema relacional solución

mántica no recogida

No ha sido posible recoger dentro del modelo relacional la siguiente semántica sí se recogía en el esquema E/R:

Toda compañía debe ser contratada por al menos una comunidad de vecinos.
A toda propiedad le debe corresponder al menos un recibo de cuota de comunidad.
Toda comunidad de vecinos tiene al menos un vocal.
Totalidad y exclusividad de la jerarquía de propiedades.

ROBLEMA 9: GESTIÓN DE HOSPITALES

unciado

Transformar el esquema E/R obtenido en el problema 14 del capítulo 1 en un quema relacional en el que se indiquen las claves primarias y candidatas, así como claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema R pero que se pierde en la transformación al modelo relacional.

scusión del enunciado

Presentamos en la figura 2.2.66 el esquema E/R del que partiremos.

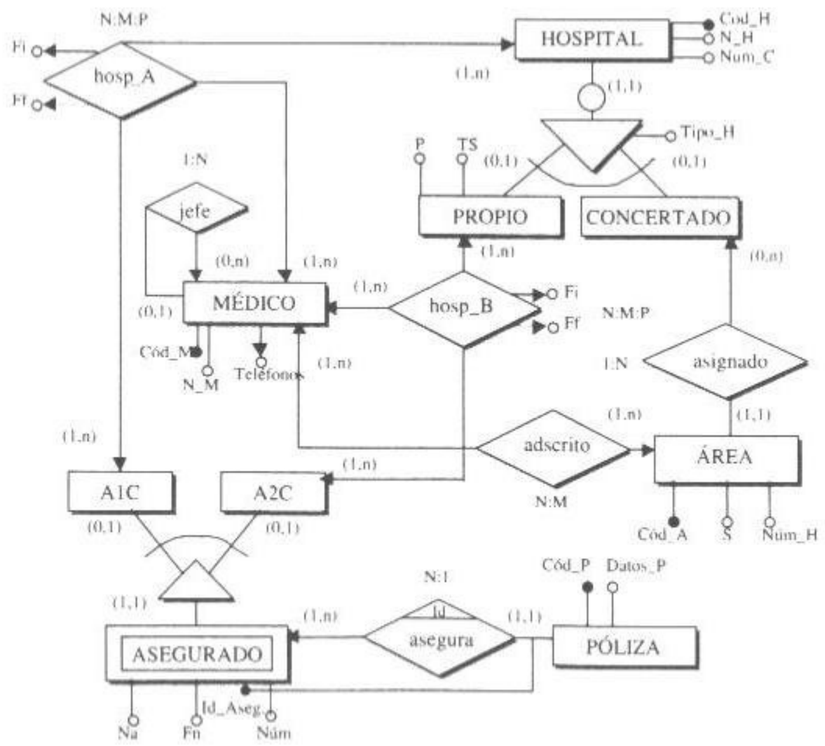


Figura 2.2.66. Esquema E/R a transformar

En la transformación al modelo relacional, cada entidad dará lugar a una relación, siendo su clave primaria el/los atributos identificador/es principales de cada entidad. Así, las entidades *MÉDICO*, *ÁREA* y *PÓLIZA* se transformarán, respectivamente, en las relaciones MÉDICO, ÁREA y PÓLIZA. El resto de las entidades las trataremos con un poco más de detalle, ya que implican o bien dependencias de identificación entre las entidades, o bien jerarquías. De momento, nos olvidaremos del atributo multivaluado "Teléfonos" de la entidad *MÉDICO*.

ÁREA (Cód_A , S, Núm_H)

MÉDICO (Cód_M , N_M)

PÓLIZA (Cód_P, Datos_P)

Figura 2.2.67

La entidad *ASEGURADO* presenta una dependencia en identificación con respecto a la entidad *PÓLIZA*, es decir, para su identificación se necesita concatenar un número correlativo ("Núm") al código de la póliza ("Cód_P"). Por lo tanto, dicha entidad se transformará en una relación cuya clave primaria estará formada por "Núm" y "Cód_P" (este último atributo será, además, clave ajena de la relación PÓLIZA, y sus opciones de borrado y modificación serán en cascada).

PÓLIZA (Cód_P , Datos_P)

B:C, M:C

ASEGURADO (*Cód_P*, Núm, Na, Fn)

Figura 2.2.68

Trataremos a continuación las dos jerarquías que aparecen en el esquema. En primer lugar, tenemos una jerarquía que nos divide los hospitales entre propios y concertados. La transformación la realizaremos siguiendo el caso general que se explicó en la introducción al modelo, es decir, transformar cada entidad en una relación. Esta solución, si bien puede no ser la más eficiente, sí es la que más semántica conserva. Así pues, tendremos una relación HOSPITAL, una relación PROPIO, y una relación CONCERTADO. Tanto la relación PROPIO como la relación CONCERTADO tendrán como clave primaria la misma que HOSPITAL (esto es, "Cód_H", que es el código del hospital) que, además, será clave ajena que referencia a la relación HOSPITAL. El borrado y la modificación será, en ambos

casos, en cascada. Por ser una jerarquía total, el atributo discriminante de la misma, "Tipo_H", no podrá tomar nunca el valor nulo, sino que tendrá que tomar el valor correspondiente a los hospitales propios o bien el de los hospitales concertados. Existen aspectos de la jerarquía que, como luego veremos, deberán ser controlados mediante disparadores y aserciones.

Cada una de las entidades hijas de la jerarquía de Asegurados la transformaremos también en una relación, porque a pesar de que ninguna de ellas posee atributos propios, sí participan en interrelaciones. Tendremos, por tanto, dos relaciones: A1C, y A2C. Ambas tendrán como clave primaria la misma que ASEGURADO, es decir, "Núm" junto con "Cód_P", que, a su vez, será clave ajena de ASEGURADO. Las opciones de borrado y modificación serán, al igual que en el caso de los hospitales, en cascada. Hemos añadido a la relación ASEGURADO un atributo discriminante "Tipo_A", que identifique al tipo de asegurado, que podrá tomar valores nulos al ser una jerarquía parcial (otra posibilidad sería definir un dominio para este atributo que incluyera los valores "A1C", "A2C", y "Otros"). Sería necesario controlar la exclusividad mediante disparadores, así como controlar que efectivamente los asegurados de primera o segunda categoría aparecen en la relación ASEGURADO con el valor del atributo "Tipo_A" adecuado, y en la relación A1C, o A2C, según corresponda.

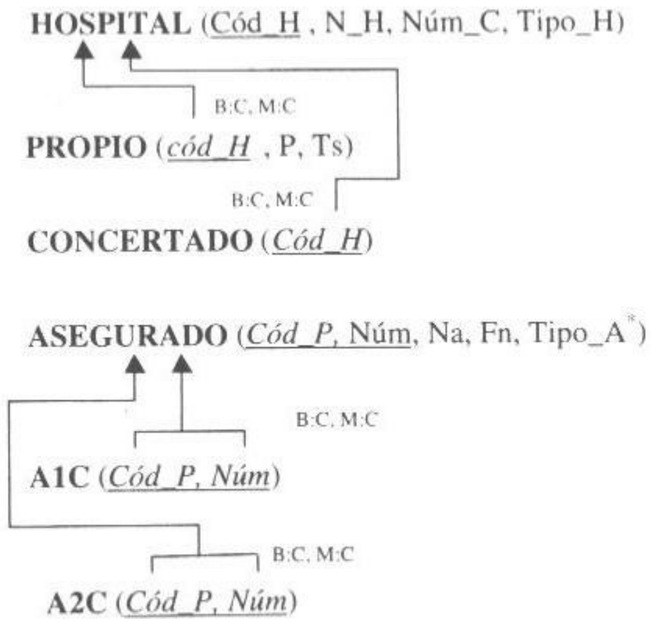


Figura 2.2.69

El atributo multivaluado "Teléfonos" de la entidad *MÉDICO* lo transformaremos en otra relación, cuya clave primaria será el propio número de teléfono concatenado con el atributo "Cód_M", que es clave primaria de la relación MÉDICO, y cuyas opciones de borrado y modificación serán en cascada. Si quisiéramos que un mismo teléfono no pudiera estar compartido por más de un médico, la clave de la relación sería únicamente el atributo "Tel".

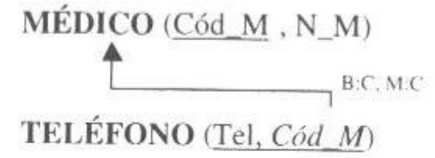


Figura 2.2.70

Veamos a continuación la transformación de las interrelaciones. Comenzaremos con las binarias que tienen cardinalidades máximas 1 en alguno de sus extremos, es decir, **Jefe** y **Asignado** (la interrelación **Asegura**, que presenta una dependencia en identificación, ya la hemos tratado al transformar la entidad débil *ASEGURADO*).

La interrelación **Asignado** tiene cardinalidades (1,1) con respecto a la entidad *ÁREA*. Por lo tanto, la transformación se realizará propagando la clave primaria de la relación ÁREA ("Cód_A") a la relación CONCERTADO, como clave ajena. El hecho de que la cardinalidad mínima sea 1 significa que todo hospital concertado debe estar asignado obligatoriamente a una única área, por lo tanto, la clave ajena no podrá tomar valores nulos. En el otro extremo de la interrelación las cardinalidades son (0,n). Esto significa que un área puede tener asignados cualquier número de hospitales (incluso ninguno), por lo que no será necesario ningún otro mecanismo que controle esta semántica.

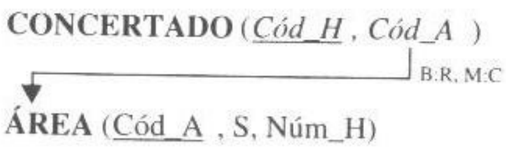


Figura 2.2.71

La otra interrelación binaria con cardinalidad máxima 1 en uno de sus extremos es **Jefe**, que nos permite recoger aquellos médicos que son jefes de otros. La interrelación es prácticamente idéntica a la anterior, excepto por el hecho de que es una interrelación reflexiva. La transformación se realiza de forma análoga. Propagaremos la clave primaria de la relación MÉDICO ("Cód_M") a sí misma

amaremos a este nuevo atributo "Jefe"), actuando como clave ajena que podrá tomar lores nulos (ya que la cardinalidad mínima era 0).

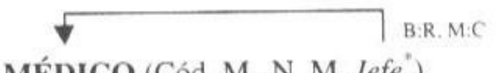


Figura 2.2.72

La última interrelación binaria que nos queda por tratar es **Adscrito**, que elaciona los médicos con las áreas a las que están adscritos. Las cardinalidades son ,n) en ambos extremos, lo cual significa que "todo médico ha de estar adscrito al enos a un área y todo área debe tener adscrito al menos a un médico". Por ser una nterrelación N:M, la transformaremos creando una nueva relación, ADSCRITO, a la ue viajarán las claves primarias de las dos relaciones, y cuya concatenación formará clave primaria de dicha relación. Cada una de ellas será, además, clave ajena, y sus pciones de borrado y modificación serán en cascada. Las cardinalidades mínimas l e cada extremo deberemos controlarlas mediante disparadores.

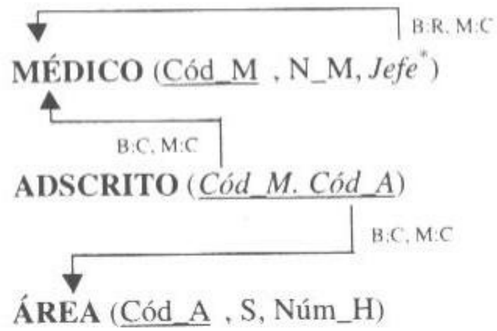


Figura 2.2.73

Hemos dejado para el final las dos interrelaciones más complicadas. Son dos nterrelaciones ternarias con atributos multivaluados, que reflejan el hecho de que un asegurado ingrese en un hospital prescrito por un médico, con una fecha de inicio y una fecha de fin de la hospitalización. Los atributos que recogen estas fechas son multivaluados, porque puede ocurrir que un mismo asegurado puede ingresar en un mismo hospital prescrito por el mismo médico en fechas distintas. Las dos interrelaciones son muy similares, por lo que las trataremos de forma genérica, para después particularizar para cada una de ellas.

Las cardinalidades de la interrelación en sus tres extremos son (1,n). La interrelación se transformará, por tanto, en una relación a la que se propagarán las

claves primarias de las tres relaciones que enlaza que serán, por tanto, claves ajenas. En principio, la regla general nos dice que la clave primaria de la nueva relación estará formada por la concatenación de estos tres atributos. Sin embargo, la presencia de atributos multivaluados temporales nos obliga a realizar un estudio más detallado de la semántica. Si suponemos que en un mismo día un asegurado no puede ser ingresado más de una vez, la clave primaria de la relación sería únicamente la concatenación de la clave del asegurado junto con la fecha de inicio de la hospitalización. Por lo tanto, para cada una de las dos relaciones HOSPITALIZA_A y HOSPITALIZA_B, la clave primaria será "Cód_P", "Núm" (clave de A1C y A2C, respectivamente, siendo clave ajena de cada una) junto a "Fi" (Fecha Inicio). Además, "Cód_H" será clave ajena que referencia al HOSPITAL (en el caso de HOSPITALIZA_A) o a PROPIO (en el caso de HOSPITALIZA_B), y "Cód_M" será clave ajena que referencia a la relación MÉDICO.

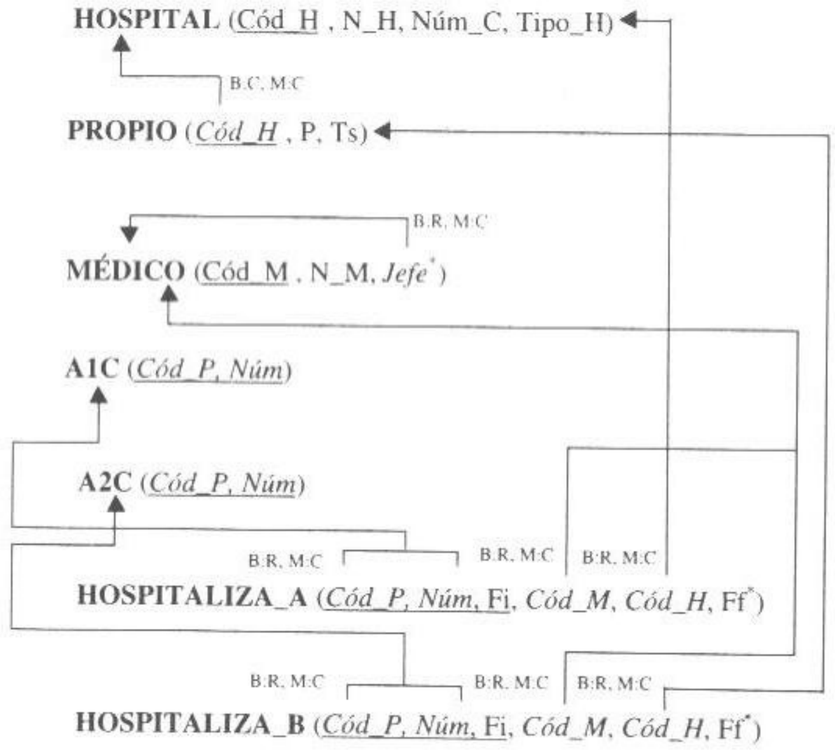


Figura 2.2.74

Propuesta de solución

En la figura 2.2.75 se muestra el grafo relacional con el resultado de la transformación.

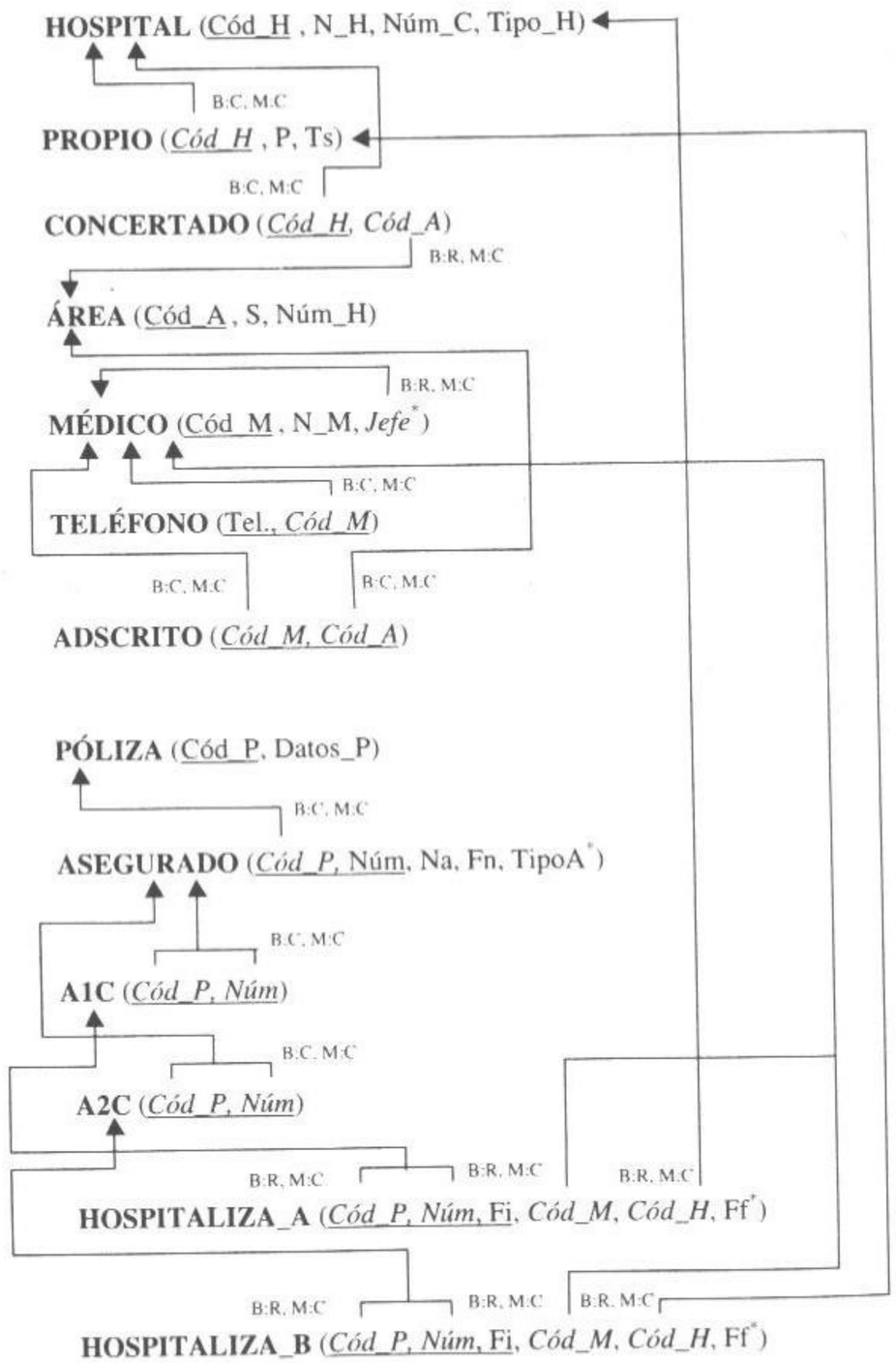


Figura 2.2.75. Esquema relacional solución.

Semántica no recogida

Como supuestos semánticos, además de los ya indicados anteriormente, se ha tenido en cuenta el borrado restringido en las relaciones HOSPITALIZA_A y HOSPITALIZA_B al tratarse de históricos, con lo que no pueden borrarse los hospitales, los médicos y los asegurados que participen en estas relaciones. Para borrarlos podría considerarse que tienen un código especial que indique si son hospitales actuales o pasados.

En el mismo caso se encuentra el borrado restringido en la relación CONCERTADO respecto al área a que pertenece. Con este tipo de borrado no pueden borrarse áreas que tengan hospitales concertados.

Además de lo expuesto, será necesario recoger por medio de disparadores, *checks* y aserciones la exclusividad y la totalidad de las jerarquías y las cardinalidades mínimas 1 de las interrelaciones (1,n) (1,n).

También sería necesario recoger el hecho de que un hospital concertado no puede cambiar el área a la que está asignado.

En el grafo relacional no podemos recoger que el número de asegurado sea correlativo.

PROBLEMA 10: OLIMPIADAS DE INVIERNO

Enunciado

Transformar el esquema E/R obtenido en el problema 15 del capítulo 1 en un esquema relacional en el que se indiquen las claves primarias y candidatas, así como las claves ajenas. Indicar, asimismo, la semántica que aparece recogida en el esquema E/R pero que se pierde en la transformación al modelo relacional.

Discusión del enunciado

Presentamos en la figura 2.2.76 el esquema E/R del que partiremos.

En primer lugar, transformaremos cada entidad del esquema ER en una relación. En el caso de las entidades que presentan una dependencia en identificación (*PISTA* y *PARTICIPANTE*), añadiremos a su clave primaria la de la entidad regular de la que dependen. Ésta será, además, clave ajena de la entidad regular, con borrados y modificaciones en cascada. En cuanto a las jerarquías, seguiremos el caso general,

nsformando cada entidad en una relación: es la solución que conserva más
mántica, y además, como vemos en el esquema, todas las entidades implicadas en
bas jerarquías participan en alguna interrelación. Las entidades hija tendrán como
ve la misma que la padre siendo, además, clave ajena. Los borrados y
odificaciones serán en cascada, y será necesario implementar, además, los
ecanismos adecuados para controlar la totalidad y la exclusividad de cada jerarquía.
momento, no consideraremos los atributos multivaluados ("Contacto" en la entidad
TACIÓN, y los atributos de fecha en las entidades *PRUEBA* y *PARTICIPANTE*).

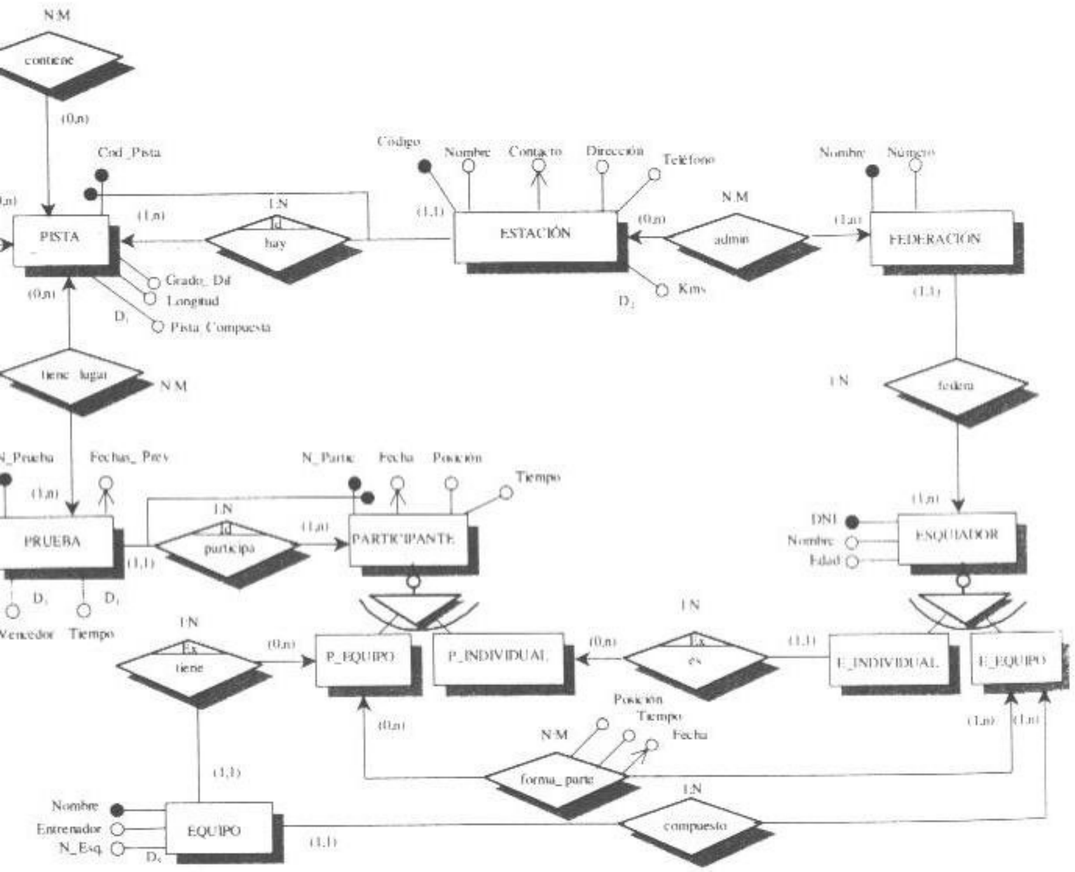


Figura 2.2.76. Esquema E/R a transformar

Transformamos, por tanto, las entidades *ESTACIÓN*, *FEDERACIÓN*, *PRUEBA*, *QUIPO* y *ESQUIADOR* en relaciones cuya clave primaria será atributo identificador rincipal de la entidad correspondiente. Las relaciones PISTA y PARTICIPANTE endrán como clave primaria la concatenación de su identificador más el de la entidad egular de la que dependen (*ESTACIÓN* y *PRUEBA*, respectivamente), la cual será lave ajena con opciones de borrado y modificación en cascada. P_EQUIPO, _INDIVIDUAL, E_INDIVIDUAL y E_EQUIPO serán relaciones cuya clave será la le su entidad padre, que además será clave ajena con borrado y modificación también n cascada. La figura muestra el resultado de estas transformaciones.

FEDERACION (Nombre, Número)

ESTACION (Código, Nombre, Dirección, Teléfono, Kms)
B:C, M:C
PISTA (Cód_Pista, *Codigo*, Grado_dif, Longitud, Pista_compuesta)

PRUEBA (N_Prueba, Vencedor*, Tiempo*)
B:C, M:C
PARTICIPANTE (*N_Prueba*, N_Partic, Posición, Tiempo)
B:C, M:C
P_EQUIPO (*N_Prueba, N_Partic*)
B:C, M:C
P_INDIVIDUAL (*N_Prueba, N_Partic*)

EQUIPO (Nombre, Entrenador, N_Esq)

ESQUIADOR (DNI, Nombre, Edad)
B:C, M:C
E_INDIVIDUAL (*DNI*)
B:C, M:C
E_EQUIPO (*DNI*)

Figura 2.2.77

Pasamos a continuación a transformar las interrelaciones 1:N que no implican una dependencia en identificación, en concreto **Federa**, **Compuesto**, **Es** y **Tiene**. Para cada una de ellas, propagaremos la clave de la relación con cardinalidad máxima 1 a la de cardinalidad máxima N, donde será clave ajena. El hecho de que las cardinalidades en todos los casos sean (1,1) implica que dicha clave ajena no pueda tomar valores nulos. Además, en el caso de **Federa** y **Compuesto**, debido a que la cardinalidad mínima en el lado N es 1, implica que deberemos controlar mediante disparadores que toda ocurrencia de las relaciones FEDERACIÓN y EQUIPO está relacionada con al menos una ocurrencia de, respectivamente, ESQUIADOR y E_EQUIPO. Los borrados y modificaciones serán restringidos o en cascada (no podrán ser con puesta a nulos, ya que ninguna de las claves ajenas que hemos propagado los admite debido a la cardinalidad mínima 1). No permitiremos borrar una federación mientras haya esquiadores federados, ni equipos mientras haya esquiadores de equipo que pertenezcan a él (borrado restringido). El resto de borrados y modificaciones serán en cascada.

La figura siguiente refleja estas transformaciones.

Transformaremos a continuación las interrelaciones N:M. Cada una de ellas se convertirá en una relación cuya clave primaria será la concatenación de los identificadores de las entidades que relaciona. Será necesario controlar las cardinalidades mínimas 1 de la interrelación. Tendremos por tanto las nuevas relaciones ADMINISTRA, FORMA_PARTE, TIENE_LUGAR y CONTIENE. Esta última merece un comentario especial. Como vemos, procede de la interrelación reflexiva que modela el hecho de que una pista puede estar formada por la unión de varias. En teoría, la clave de esta nueva relación sería la concatenación de la clave de

os pistas (la compuesta y la componente), es decir, el código de estación y el código e pista, ambos por duplicado. Sin embargo, el enunciado nos exigía que cuando una ista está formada por otras, éstas (y aquélla) deberían pertenecer a la misma estación. Aunque esta restricción no pudimos recogerla en el esquema E/R, ahora sí podemos acerlo fácilmente, sólo con eliminar uno de los atributos que referencian a la stación. Las claves ajenas de ambas pistas utilizarán este atributo, lo que obligará a que en ambas coincida su estación.

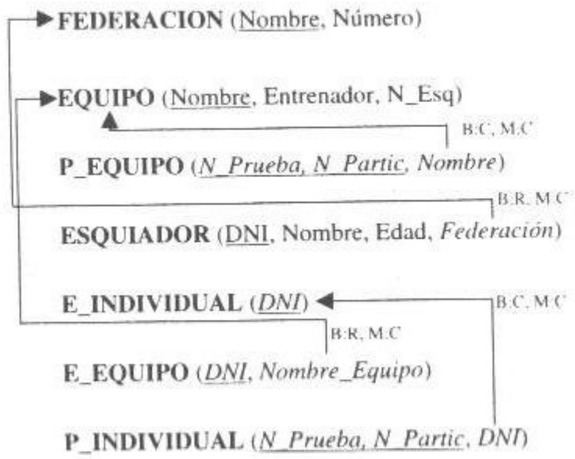


Figura 2.2.78

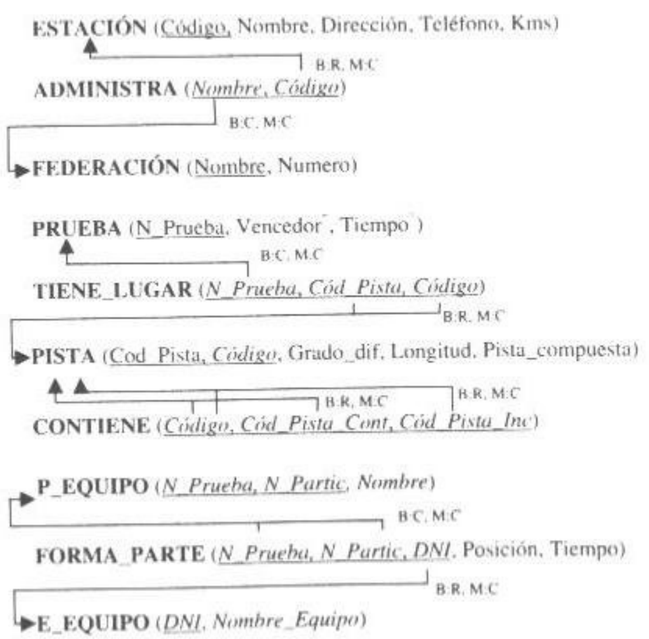


Figura 2.2.79

Al igual que hicimos en el caso de las entidades, de momento no consideraremos el atributo multivaluado "Fecha" de la interrelación **Forma_Parte**.

Por último, nos falta realizar las transformación de los atributos multivaluados. Como sabemos, en el caso general, se creará una nueva relación cuya clave primaria estará formada por el atributo más la clave de la relación de la que es atributo. Sin embargo, en el caso de atributos multivaluados en interrelaciones, es necesario estudiar la semántica, que puede llevar a que no sean necesarias las claves de todas las entidades que enlaza la interrelación. En nuestro caso, el único atributo multivaluado de una interrelación es "Fecha", en la interrelación **Forma_Parte**. Si consideramos, por ejemplo, que, un esquiador de equipo en una fecha determinada formará parte de un único equipo participante, bastaría con las claves de E_EQUIPO y la fecha para identificar a la nueva relación. Además, puesto que ninguno de los atributos es opcional, será necesario controlar que toda ocurrencia de las relaciones PRUEBA, PARTICIPANTE, ESTACIÓN y FORMA_PARTE tiene asociada al menos una ocurrencia en la correspondiente relación que almacena las fechas o, en el caso de ESTACIÓN, una persona de contacto. Los borrados y las modificaciones serán en todos los casos en cascada.

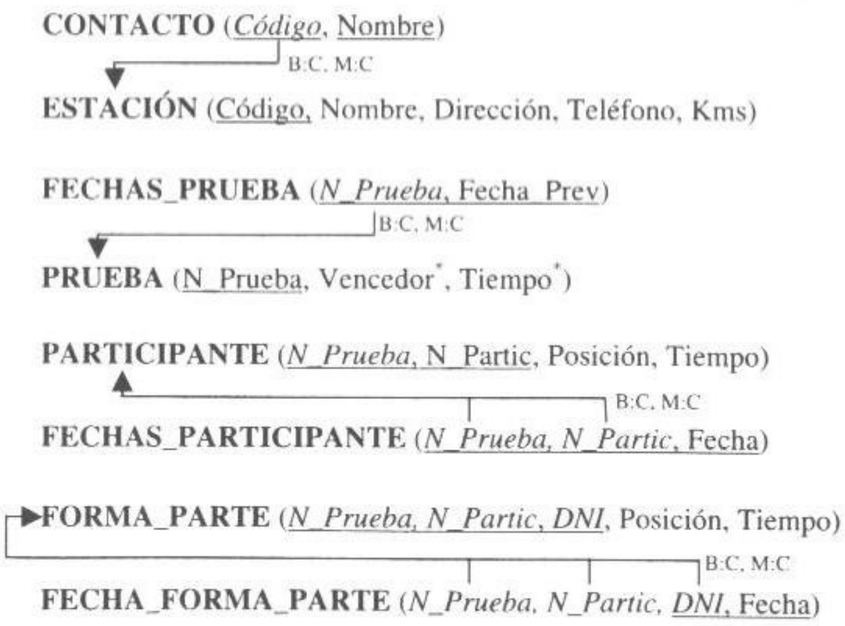


Figura 2.2.80

Propuesta de solución

En la figura 2.2.81 se muestra el grafo relacional con el resultado de la transformación.

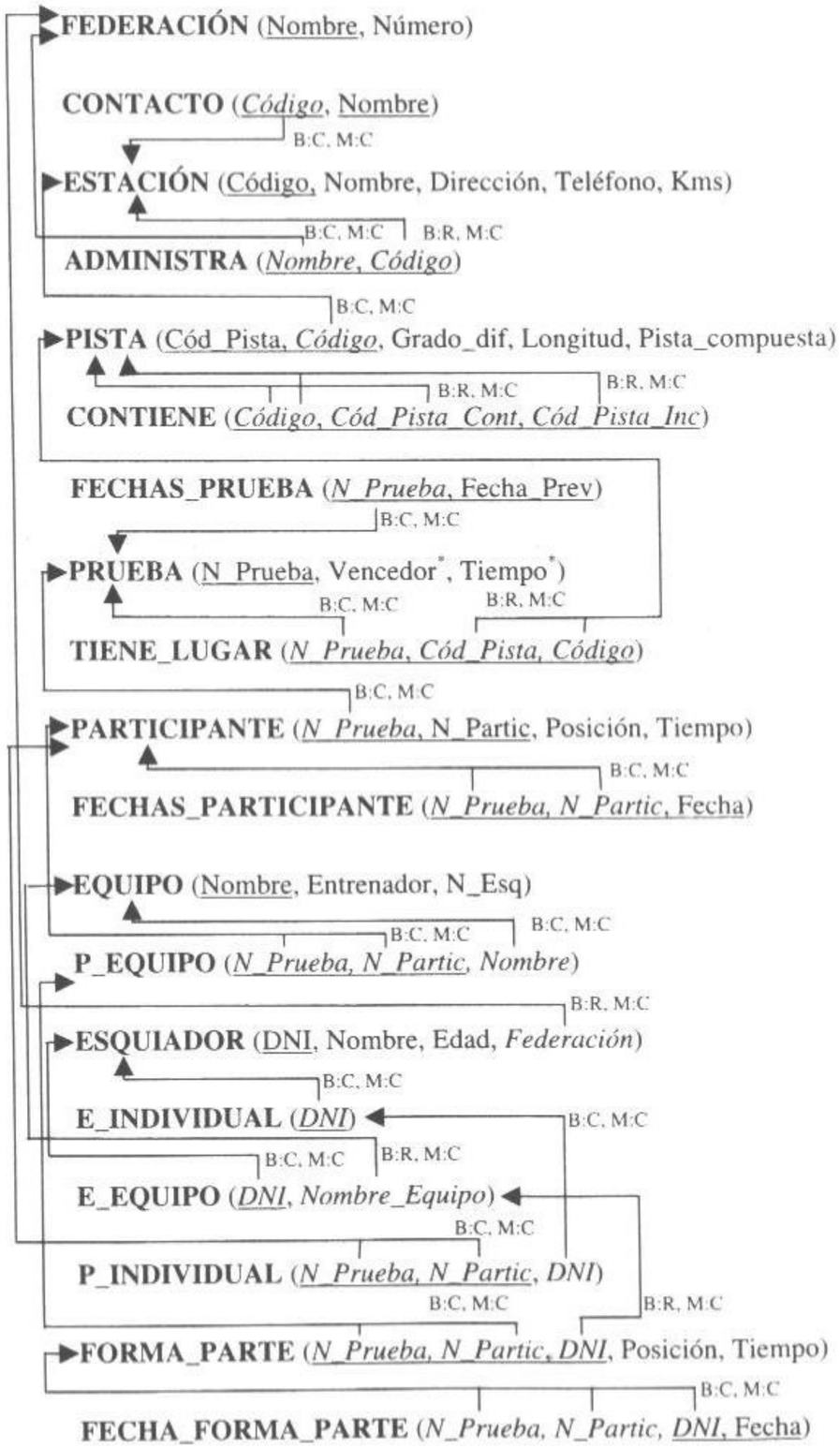


Figura 2.2.81. Esquema relacional solución

Semántica no recogida

Es necesario controlar que un esquiador no pueda cambiar de equipo durante los juegos, así como que, si es individual, no pueda ser de equipo, y al revés.

Atributos derivados.

Totalidad y exclusividad de las jerarquías.

Cardinalidades mínimas 1 en interrelaciones con cardinalidades (1:n) en alguno de sus extremos.

CAPÍTULO 3

NORMALIZACIÓN DE RELACIONES

INTRODUCCIÓN

Guías metodológicas para abordar la resolución de los problemas

La normalización de una relación es un proceso muy definido, donde incluso existen algoritmos de normalización. Sin embargo en la resolución de los problemas de este libro se utilizarán heurísticas, ya que son más intuitivas que los algoritmos, y por lo tanto más fáciles de entender y aplicar al ser problemas con un número reducido de dependencias y atributos.

Se recomienda la utilización de los algoritmos de normalización si en el caso práctico existen un número elevado de dependencia y atributos, con el fin de evitar confusiones y equivocaciones en el cálculo.

La mayoría de las definiciones utilizadas en la introducción de este capítulo, y los algoritmos de normalización, se pueden encontrar en los libros De Miguel et al., (1999) y De Miguel et al., (1995), excepto en las que se indique explícitamente lo contrario. Alguna definición se ha adaptado a los propósitos de este libro, para facilitar la comprensión.

El proceso de normalización se trata las dependencias que existen entre los
butos (campos) de la relación. Los pasos que se realizan para normalizar una
ción son los siguientes:

1. **Cálculo de las dependencias** funcionales, multivaluadas, jerárquicas y en combinación que existen entre los atributos de la relación.

2. **Cálculo del recubrimiento minimal**

3. **Cálculo de las claves candidatas de la relación**, de los atributos principales y de los no principales

4. **Cálculo de la forma normal en la que se encuentra la relación**

5. **Aplicar los métodos de síntesis o análisis** para obtener la forma normal deseada.

Todos estos pasos se explican con mayor grado de detalle en los apartados del 1
5 de la introducción.

CÁLCULO DE LAS DEPENDENCIAS

Entre los atributos de una relación (R) pueden existir dependencias de varios
os. Las dependencias son propiedades inherentes al contenido semántico de los
tos, formando parte de las restricciones de usuario del modelo relacional. Existen
stintos tipos de dependencias: **funcionales, multivaluadas, jerárquicas y en
mbinación**, que se explican a continuación:

ependencias funcionales

Definición: Sea el esquema de relación *R(AT,DEP)*, y *X* e *Y* dos subconjuntos de
T denominados descriptores. ***Y* depende funcionalemente de *X*** (*X* implica o
etermina a *Y*) si para cada valor de *X* solamente existe un único valor posible para *Y*.

Notación: $X \rightarrow Y$

Además, al conjunto *X* se le denomina implicante o determinante y al conjunto *Y*
le denomina implicado.

Un ejemplo de dependencia funcional es la que existe entre los atributos *DNI* de
n profesor y *Nombre* del profesor:

DNI_profesor → *Nombre_profesor*

Si además se cumpliera que no pueden existir dos profesores con el mismo nombre, esto implicaría que el nombre también puede actuar como clave de la relación *Profesor*, y por lo tanto, también determina funcionalmente a su *DNI*, es decir:

Nombre_profesor ←→ *DNI_profesor*

Cuando dos o más atributos se implican funcionalmente mutuamente se dice que son *equivalentes*.

Existen varios tipos de dependencias funcionales, y en ellas se basan algunos de los axiomas de Armstrong, utilizados para hallar el cierre transitivo de un determinante, pero no se entra en detalle en su definición. La definición de los axiomas de Armstrong se puede encontrar en De Miguel et al., (1999).

A continuación, se realizan una serie de definiciones necesarias para el proceso de normalización de relaciones.

La dependencia $X \rightarrow Y$ es:

- Dependencia funcional (DF) trivial si Y es un subconjunto de X (p.e.: *Nombre_profesor* → *Nombre_profesor*)

- DF elemental si Y es un único atributo, se trata de una dependencia plena (no parcial) y no trivial (p.e.: *Nombre_profesor* →*DNI_profesor*)

- DF transitiva si en el esquema de relación R({X,Y,Z},DEP) existen las dependencias X→Y; Y→Z; Y-/->X (p.e: *Nombre_profesor* →*Código_Postal; Código_Postal* →*Ciudad; Código_Postal -/-> Nombre_profesor*)

Dependencias multivaluadas

Definición: Sean *X* e *Y* dos descriptores. ***X multidetermina a Y*** si para cada valor de *X* existe un conjunto bien definido de valores posibles en *Y*, con independencia del resto de los atributos de la relación [Fagin, 1977].

Notación: $X \rightarrow\rightarrow Y$

Por ejemplo, en la tabla siguiente el atributo *DNI* multidetermina a *titulación*:

Nombre	*Titulación*
Felipe	Magisterio Música
Felipe	Ing. Superior en Informática
Antonia	Ing. Caminos

Al igual que ocurría con las dependencias funcionales, las dependencias multivaluadas también pueden ser de varios tipos, pero no se entra en detalle.

Existe un tipo especial de dependencias multivaluadas, que son las dependencias ltivaluadas jerárquicas.

pendencias jerárquicas

Definición: Sean *X*, *Y* y *Z* descriptores. ***X multidetermina jerárquicamente a Y y*** si en la proyección de los atributos *X* multidetermina a *Y* y *X* multidetermina a *Z*.

Notación: $X \rightarrow\rightarrow Y \mid Z$

Un ejemplo de este tipo de dependencias se muestra en la tabla siguiente, donde a persona posee varios *títulos* y trabaja en varios *proyectos*, donde no se exige nguna titulación para trabajar en ningún proyecto (no hay relación entre estos ributos):

Nombre	*Titulación*	*Proyecto*
Felipe	Magisterio en Música	A
Felipe	Ing. Superior en Informática	B
Antonia	Ing. Caminos	A

ependencias en combinación

Definición: Una relación **R tiene dependencia en combinación respecto de sus royecciones (R1, R2, ...Rn)** si R = R1 * R2 * ... * Rn.

Notación: *DJ*(R1,R2, ...,Rn)*

Un ejemplo de esta propiedad es el que aparece en la siguiente tabla, donde existe na relación de "envidia" entre dos personas. Si, por ejemplo *Felipe* trabaja en varios royectos (*A y B*) y además realiza todos los cursos posibles sobre *Bases de Datos* e *geniería del Sw* (las dos primeras tuplas). Si en esos momentos se inserta una nueva pla (la tercera) donde *Antonia* trabaja en el proyecto *B* y realiza un curso de BD, ntonces forzosamente *Felipe*, que trabaja en el mismo proyecto, ha de recibir el ismo curso:

Nombre	*Tema Curso*	*Proyecto*
Felipe	BD	A
Felipe	IS	B
Antonia	BD	B
Felipe	BD	B

Para simplificar el estudio de las dependencias en los problemas de este libro, se onsidera, sin pérdida de generalidad, que el esquema relacional se compone de un

único esquema de relación (una única relación, denominada *relación universal*). En los problemas posteriores se utiliza la notación *R(AT,DEP)* para referenciar a la relación universal *R*, que tiene como atributos el conjunto *AT* y las dependencias que existen entre sus atributos se recogen en el conjunto *DEP*.

2. CÁLCULO DEL RECUBRIMIENTO MINIMAL

Un conjunto de dependencias es mínimo, y se denota como DEP^m, cuando se cumplen las siguientes condiciones:

a) Todas sus dependencias son **elementales,** es decir, que todas sus dependencias sean plenas, no triviales y tengan un único atributo implicado.

b) No existe en ninguna de las dependencias **atributos extraños**. Un atributo *A* perteneciente a *X* es extraño en la dependencia $X \rightarrow Y$ si la dependencia $(X-A) \rightarrow Y$ se deduce del resto de las dependencias de la relación mediante los axiomas de Armstrong.

   Por ejemplo, sea el esquema de relación $R(\{A,B,C\}\{A \rightarrow C,\ AB \rightarrow C\})$. Como el atributo *A* implica funcionalmente a *C*, el atributo *B* es redundante en la segunda dependencia.

c) No existe en la relación ninguna **dependencia redundante**. Una dependencia *dp* se dice que es redundante si sus implicados se deducen a partir del resto de las dependencias de la relación.

   Por ejemplo, sea $R(\{A,B,C\}\{A \rightarrow B,\ B \rightarrow C,\ A \rightarrow C\})$. Como el atributo *A* determina transitivamente al atributo *C*, la última dependencia $(A \rightarrow C)$ es una dependencia redundante.

Puede ocurrir que existan varios conjuntos de dependencias mínimos válidos, ya que todo depende de la elección de las dependencias redundantes y de los atributos extraños que se eliminan en el proceso del cálculo del conjunto minimal.

3. CÁLCULO DE CLAVES CANDIDATAS

Claves candidatas (CC)

Dado un esquema de relación *R(AT,DEP)* se denomina ***superclave*** de la relación R a un conjunto no vacío de atributos pertenecientes a *AT* tal que estos atributos determinan a todo el conjunto *AT*, es decir, que el cierre transitivo de ese conjunto de atributos sea todo *AT*.

El cierre transitivo de *X* se define como el conjunto de atributos determinados por *X* aplicando los axiomas de Armstrong, y se denota como X^+.

Por ejemplo, en el esquema de relación $R(\{A,B,C,D\}\{A \rightarrow B,D;\ B \rightarrow C\})$ $A^+ = \{A,$ $, C\}$, $B^+ = \{B, C\}$, $C^+ = \{C\}$, $D^+ = \{D\}$.

Se denomina ***clave candidata (CC)*** de una relación a una superclave tal que ple que ningún subconjunto de la misma determina a todo *AT*.

Por ejemplo, en *R* la única clave candidata que existe es el atributo *A*, ya que rmina a todo *AT*. El conjunto *{B, A}* no es una clave candidata, aunque determina lo *AT*, porque el subconjunto *{A}* determina a *AT*.

butos principales y no principales

Por otro lado, es necesario para el cálculo de la forma normal de una relación r cuáles son sus atributos principales y no principales.

Un ***atributo es principal (AP)*** cuando forma parte de alguna de las claves lidatas de la relación. Un ***atributo es no principal (ANP)*** cuando no forma parte inguna de las claves candidatas.

Por ejemplo, en la relación $R(\{A,B,C,D\}\{A \rightarrow B,D;B \rightarrow C\})$, donde la única clave lidata que existe es $\{A\}$, el atributo *A* es principal, mientras que el resto de los outos (*B, C, D*) son atributos no principales.

ORMA NORMAL DE UNA RELACIÓN

Existen seis niveles de normalización de una relación, que se muestran en la ra 3.1. Una relación se encuentra en uno u otro grado de normalización si cumple serie de propiedades (restricciones) que se explican a continuación.

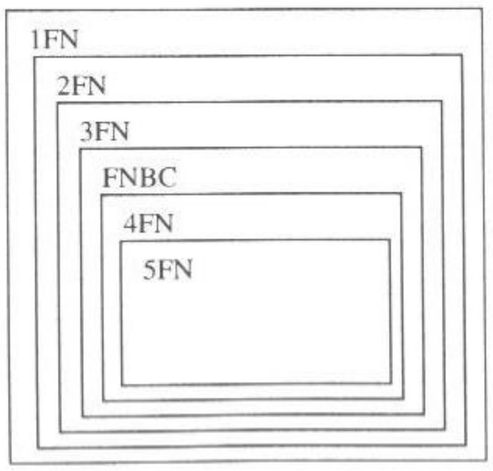


Figura 3.1. Niveles de normalización

Las tres primeras formas normales las definió Codd [Codd, 1970] y se basan en las dependencias funcionales que existen entre los atributos de la relación. Más tarde, en 1974, y debido a que todavía existían anomalías, redefinió la 3FN y la denominó forma normal de Boyce-Codd (FNBC).

Posteriormente, y basándose en las dependencias multivaluadas y en combinación, se definieron dos niveles más de normalización, la 4FN y la 5FN respectivamente.

De esta forma, las relaciones en primera forma normal (1FN) tienen más redundancia de datos que los niveles superiores, y por lo tanto más anomalías de actualización de datos. Y la quinta forma normal (5FN) es el grado de normalización máximo que puede alcanzar una relación.

Primera forma normal (1FN)

Se dice que una relación se encuentra en 1FN cuando cada atributo sólo toma un valor del dominio simple subyacente. Es decir, que no existen grupos repetitivos.

Un ejemplo es el que aparece en las tablas siguientes, donde en la primera tabla no se encuentra en 1FN, sin embargo en la segunda tabla sí que se cumple esta restricción, y por lo tanto se trata de una relación.

Nombre	*Titulaciones*
Felipe	Informática Caminos
Antonia	Magisterio musical

Nombre	*Titulaciones*
Felipe	Informática
Felipe	Caminos
Antonia	Magisterio musical

La primera forma normal es una restricción inherente al modelo relacional, por lo que su cumplimiento es obligatorio para toda relación.

Segunda forma normal (2FN)

Se dice que una relación se encuentra en 2FN si:

- Se encuentra en 1FN
- Cada atributo no principal tiene dependencia funcional completa respecto de alguna de las claves

Por ejemplo, en el esquema de relación $R(\{A,B,C,D\},\{A,B \rightarrow C;\ A \rightarrow D\})$, donde la ve candidata es el conjunto $\{A,B\}$, y por lo tanto los atributos principales son A y B os no principales C y D, el atributo no principal D, depende de A, pero no de una ve. Por lo tanto, el esquema no se encuentra en 2FN.

Sin embargo los esquemas $R1(\{A,B,C\},\{A,B \rightarrow C\})$ y $R2(\{A,D\},\{A \rightarrow D\})$ sí se cuentran en 2FN.

ercera forma normal (3FN)

Se dice que una relación se encuentra en 3FN si:

- Se encuentra en 2FN
- No existe ningún atributo no principal que dependa transitivamente de alguna clave de R

Por ejemplo, en el esquema de relación $R(\{A,B,C\},\{A \rightarrow B;\ B \rightarrow C\})$, la clave andidata de la relación es el atributo A, y los atributos no principales B y C. Como el tributo C depende transitivamente de la clave A, la relación no se encuentra en 3FN. unque sí en 2FN, ya que C depende transitivamente de la clave.

Sin embargo, las relaciones $R1(\{A,B\},\{A \rightarrow B\})$ y $R2(\{B,C\},\{B \rightarrow C\})$ sí que se ncuentran en 3FN.

Forma normal de Boyce-Codd (FNBC)

Se dice que una relación se encuentra en FNBC si y sólo si todo determinante es clave candidata.

Por ejemplo, la relación $R(\{A,B,C,D\},\{A \leftarrow\rightarrow B;\ A,C \rightarrow D\})$ no se encuentra en FNBC, ya que los determinantes A y B no son claves candidatas, sino que lo son los conjuntos $\{A,C\}$ y $\{B,C\}$.

Sin embargo, las relaciones $R1(\{A,B\},\{A \leftarrow\rightarrow B\})$ y $R2(\{A,C,D\},\{A,C \rightarrow D\})$ sí se encuentran en FNBC.

Cuarta forma normal (4FN)

Se dice que una relación se encuentra en 4FN si y sólo si las únicas dependencias multivaluadas no triviales son aquellas en las que una clave multidetermina un atributo.

Por ejemplo, la relación $R(\{A,B,C\},\{A \rightarrow\rightarrow B;\ A \rightarrow\rightarrow C\})$ no se encuentra en 4FN, ya que su clave candidata es el conjunto $\{A, B, C\}$ y en la primera dependencia el atributo C no participa y en la segunda es el atributo B el que no participa.

Sin embargo, las relaciones $R1(\{A,B\},\{A \rightarrow\rightarrow B\})$ y $R2(\{A,C\},\{A \rightarrow\rightarrow C\})$ sí se encuentran en 4FN.

Quinta forma normal (5FN)

Se dice que una relación se encuentra en 5FN si y sólo si:

- Se encuentra en 4FN
- Toda dependencia de combinación está implicada por una clave candidata.

También se puede dar otra definición para la 5FN: "una relación se encuentra en 5FN si y sólo si toda dependencia funcional, multivaluada o de combinación no trivial es consecuencia de las claves candidatas".

6. MÉTODOS DE ANÁLISIS Y SÍNTESIS

Los métodos de análisis (o descomposición) y síntesis tratan de transformar, por medio de sucesivas proyecciones, un esquema de relación en un conjunto de n esquemas resultantes. Estos esquemas siempre han de cumplir la conservación de la información (atributos y tuplas), de las dependencias y que existan mínima redundancia en los datos.

De esta forma siempre los n esquemas resultantes serán equivalentes a la relación inicial y tendrán menor redundancia en los datos.

Se aplican estos métodos para aumentar el nivel de normalización de las relaciones, eliminando redundancias y anomalías en las modificaciones, inserciones y borrados de los datos de la misma.

Análisis

El método de análisis fue propuesto por Codd y posteriormente fue ampliado por Rissanen, Fagir y Zaniolo para incluir las dependencias multivaluadas y en combinación. Trata la descomposición en sucesivas proyecciones de la relación universal (con todos los atributos del universo del discurso y las dependencias existentes entre ellos) paso a paso hasta alcanzar la 5FN.

Los esquemas resultantes son cada vez de menor grado y se encuentran en un nivel de normalización mayor.

El proceso se lleva a cabo mediante el *árbol de descomposición*, que se muestra en la figura 3.2, donde $A_1,...,A_n$ son los atributos y $d_1,...,d_m$ las dependencias que existen entre los mismos.

El proceso de descomposición termina cuando las relaciones se encuentran en la forma normal objetivo o cuando la continuación del proceso supone pérdidas no deseadas de dependencias.

Los pasos del algoritmo de análisis son los siguientes:

1. Hallar el recubrimiento minimal DEP^m.
2. Hallar las claves, los atributos principales y los no principales.
3. Identificar la FN en la que se encuentra la relación.
4. Agrupar las dependencias funcionales (DF) que tienen el mismo implicante o equivalentes y obtener proyecciones independientes con estos grupos de dependencias.
5. Tratar las dependencias multivaluadas (descomponerlas o eliminarlas).
6. Tratar las dependencias en combinación. Si existe alguna que no esté implicada por la clave, habrá que descomponerla.

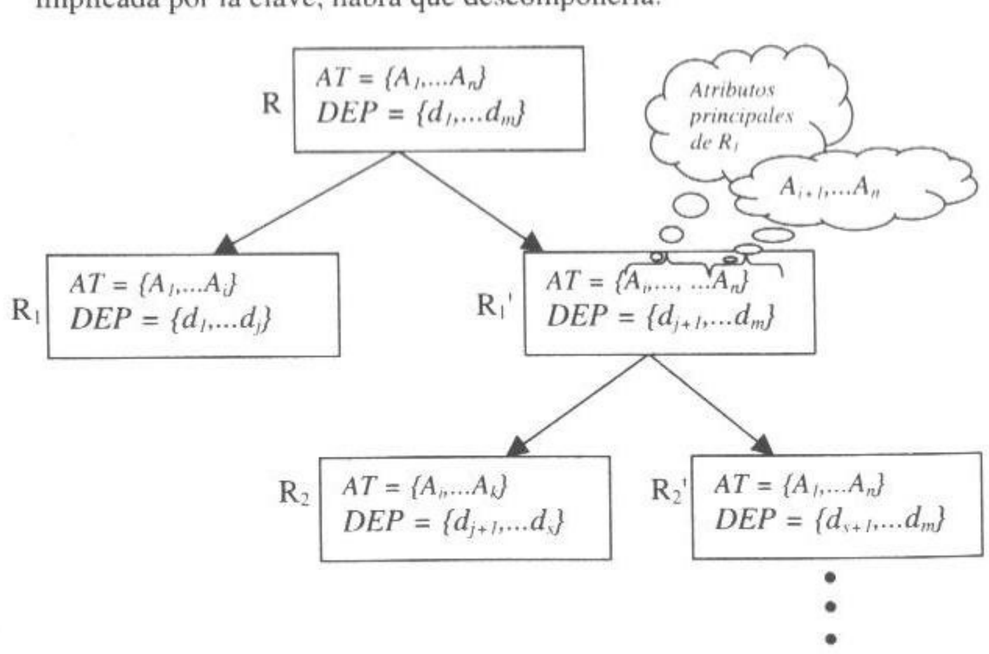


Figura 3.2. Árbol de descomposición general

En el paso cuarto del algoritmo es importante el orden en el que se eligen las dependencias funcionales de una relación para su descomposición en otras relaciones. Esto se debe a que los atributos no principales de la nueva relación creada con la/s dependencia/s elegida/s no se mantienen en la relación temporal creada para continuar la descomposición.

Debido a esto es posible que se pierdan dependencias funcionales en la descomposición de la relación, cosa que no siempre es deseable.

Por otro lado, cuando existen varias dependencias en un momento dado que no inducen a pérdidas, la elección de una u otra es independiente del resultado. Es decir, pueden existir distintos árboles de descomposición, y distintos resultados válidos para una misma relación inicial.

En los árboles de descomposición de los ejercicios resueltos la notación utilizada es la siguiente:

- R: relación universal.
- AT: conjunto de atributos de una relación.
- DEP: conjunto de dependencias de una relación.
- $R_1, R_2, \ldots R_n$: relaciones resultantes de la descomposición de la relación R.
- $R_1', R_2', \ldots R_n', R_{1aux}, R_{2aux}, \ldots R_{naux}$: relaciones temporales necesarias para el proceso de descomposición de una relación.

Un ejemplo del algoritmo de análisis es la descomposición de la relación *R(AT,DEP)*, donde

$AT = \{A,B,C,D,E,F\}$

$DEP^m = \{$

$A,B \rightarrow C$

$A,D \rightarrow F$

$A,C \rightarrow E$

$\}$

En la relación *R*, la clave candidata es el conjunto $\{A,B,D\}$, y por lo tanto, AP=$\{A,B,D\}$ y ANP=$\{C,E,F\}$. Se encuentra en 1FN, ya que, por ejemplo, el atributo *C* en la primera dependencia sólo dependen de los atributos $\{A,B\}$, pero no de toda la clave.

El árbol de descomposición de esta relación se muestra en la figura 3.3.

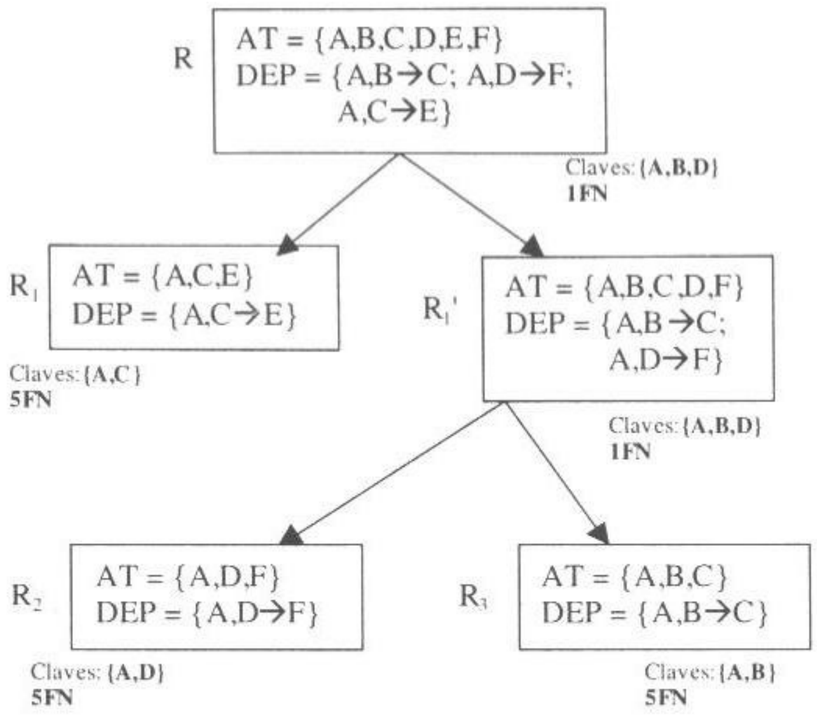


Figura 3.3. Árbol de descomposición de ejemplo

íntesis

El proceso de síntesis recorre el camino inverso al proceso de análisis. Trata de grupar las distintas dependencias que existen entre los atributos del universo del iscurso, sintetizando relaciones.

Es decir, ambos procesos tienen el mismo fin, apoyándose ambos en el mismo oncepto de dependencias y de recubrimiento irredundante.

Sin embargo, el proceso de síntesis no trata las dependencias multivaluadas ni las ependencias en combinación, por lo que solamente asegura que las relaciones esultantes puedan estar en FNBC. Frente al proceso de análisis, que tiene en cuenta stas dependencias, y por lo tanto permite llegar hasta la 5FN. Por supuesto, todo esto eniendo en cuenta que no siempre se pueda llegar sin pérdidas de dependencias ni de nformación.

Los pasos del algoritmo de síntesis se resumen a continuación:

1. Hallar el recubrimiento minimal DEP^m.
2. Agrupar las dependencias funcionales (DF) en particiones que tienen el mismo implicante, uniendo los atributos equivalentes.

3. Crear un esquema de relación (Ri) para cada partición, que tenga como atributos todos los que participen en las dependencias y como grupo de dependencias las del grupo.
4. Si existen atributos que no son implicantes ni implicados en DEP^m, formar un esquema de relación con éstos sin dependencias o alternativamente crear un esquema con la clave de la relación y sin dependencias.

Se realiza el proceso de síntesis para el mismo caso que se resolvió mediante análisis, es decir, para la relación universal *R(AT,DEP)*, donde:

$AT = \{A,B,C,D,E,F\}$

$DEP^m = \{$

$A,B \rightarrow C$

$A,D \rightarrow F$

$A,C \rightarrow E$

$\}$

Se crea un nuevo esquema de relación por cada conjunto de dependencias que poseen el mismo implicante. En este caso los grupos tienen cardinalidad uno, ya que no existen dos dependencias con el mismo determinante. Por lo tanto, el esquema de relación final es:

$R1(\{A,B,C\},\{A,B \rightarrow C\})$

$R2(\{A,D,F\},\{A,D \rightarrow F\})$

$R3(\{A,C,E\},\{A,C \rightarrow E\})$

En este caso las relaciones obtenidas son las mismas que con el proceso de análisis, y todas ellas se encuentran en 5FN.

PROBLEMA 1

Enunciado

Dado el siguiente esquema de relación:

R(AT, DEP)

Donde AT = *{Cod-L, Cod-M, Atr-L, Atr-M, a, b}* y DEP es el conjunto relaciones de los atributos del esquema relacional de la figura 3.4, determinar la na normal en la que se encuentra el anterior esquema de relación, explicando cada de los pasos llevados a cabo, y normalizarlo hasta 4FN (siempre que no se pierdan endencias); Indicar, para cada esquema de relación resultante, las claves y la forma nal en la que se encuentra.

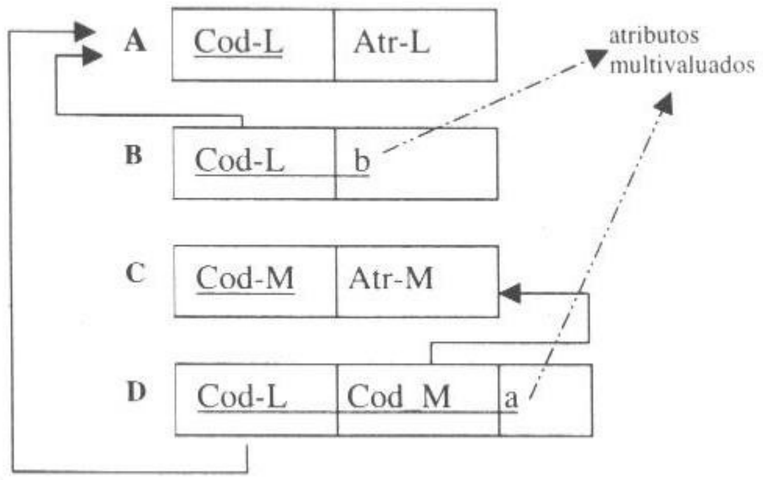


Figura 3.4. Esquema relacional

NOTA: En el esquema relacional de la figura los atributos nombrados con letras minúscula son atributos multivaluados.

scusión del enunciado

PENDENCIAS ENTRE ATRIBUTOS

El primer paso para calcular en qué forma normal se encuentra la relación R es ar qué tipo de dependencias existen entre los atributos del esquema relacional de la ura 3.4.

pendencias funcionales

En la primera relación de la figura 3.4 (relación A), se tiene como clave primaria el atributo *Cod-L*. Toda clave primaria de una relación implica funcionalmente al resto de los atributos de la relación. Ya que por cada valor de la clave primaria, a cada atributo le corresponde un único valor. De esta forma, *Cod-L* implica funcionalmente a *Atr-L:*

$$Cod\text{-}L \rightarrow Atr\text{-}L$$

- Ocurre lo mismo en la relación C, donde *Cod-M* es clave primaria de la relación, y por lo tanto implica funcionalmente al atributo *Atr-M*:

$$Cod\text{-}M \rightarrow Atr\text{-}M$$

Dependencias multivaluadas

- En la relación B no existen dependencias funcionales entre sus atributos pero sí existe una dependencia multivaluada, ya que, para un único valor del determinante (*Cod-L*), existe un conjunto bien definido de valores que puede adoptar el implicado (atributo multivaluado *b*).

$$Cod\text{-}L \rightarrow\rightarrow b$$

- Se puede encontrar otra dependencia multivaluada en la relación D, donde, al igual que ocurre con la relación B, no existe ninguna dependencia funcional. En esta relación son *Cod-L* y *Cod-M* son los que multideterminan al atributo multivaluado *a*.

$$Cod\text{-}L,\ Cod\text{-}M \rightarrow\rightarrow a$$

Recubrimiento irredundante (o minimal)

A continuación, se comprueba que las dependencias halladas en los apartados anteriores son irredundantes, es decir, si el conjunto de dependencias es mínimo (DEP^m):

- Lo primero es comprobar que todas las dependencias son elementales:
  - En todas las dependencias halladas se cumple que tienen un único implicado.
  - Todas las dependencias son plenas, es decir, que ningún subconjunto del implicante podría implicar al implicado.
  - Ninguna dependencia es trivial, es decir, ningún atributo del determinante aparece como implicado.

  Por lo tanto, se cumple que todas las dependencias son elementales.

- A continuación se comprueba que en ninguna dependencia de DEP existen atributos extraños. Un atributo *A* es extraño si dada la dependencia $X \rightarrow Y$ de DEP, la dependencia $(X\text{-}A) \rightarrow Y$ se encuentra en DEP^+.

- Para comprobar que la dependencia *(X-A)* → *Y* pertenece a DEP⁺ se ha de cumplir que *Y* pertenece al cierre de *X-A* ($(X\text{-}A)^+_{DEP}$).

En el conjunto de dependencias (DEP) no existen atributos extraños.

Por último se comprueba que no existe ninguna dependencia redundante, es decir, que se pueda deducir a partir de las demás mediante los axiomas de Armstrong.

Por ser el primer ejercicio de normalización del libro, se realiza el cálculo de las dependencias redundantes aplicando el algoritmo optimizado que aparece en el libro De Miguel et al. (1999) para hallar las dependencias funcionales redundantes: Una dependencia $X \rightarrow Z$ es redundante si $Z \in (X \cup RL_X)^+_{DEP} \cap RO_X$. Se comprueba, entonces, para cada una de las dependencias de DEP^m, si se cumple la condición de redundancia:

- Cod-L → Atr-L:

  - $RL_{Cod\text{-}L}$ = Conjunto de atributos de la parte derecha de la dependencia que pertenecen a la parte izquierda y derecha de la dependencia = { } *(conjunto vacío)*

  - $RO_{Cod\text{-}L}$ = Conjunto de atributos de la parte derecha de la dependencia que pertenecen solamente a la parte derecha de la misma = *{Atr-L}*.

  - *r-cierre:* $(Cod\text{-}L \cup \phi)^+_{DEP'}$ = Conjunto de todos los atributos de las partes derechas de las dependencias derivados de X respecto a DEP' = (DEP – {*Cod-L* → *Atr-L*}). Para calcular el r-cierre de *Cod-L*, primero se calcula el cierre transitivo de *Cod-L* ($Cod\text{-}L^+_{DEP}$) y posteriormente el conjunto de atributos determinados funcionalmente de forma directa a partir del conjunto $Cod\text{-}L^+_{DEP}$.

  - Cierre transitivo ($Cod\text{-}L^+_{DEP}$) = conjunto de todos los atributos implicados funcionalmente por *Cod-L* directa o indirectamente (mediante el axioma de transitividad). En DEP' no existe ninguna dependencia donde *Cod-L* sea determinente, por lo que $Cod\text{-}L^+_{DEP} = Cod\text{-}L$.

  - Por lo tanto, $(Cod\text{-}L)^+_{DEP'}$ = el conjunto vacío (ϕ).

  - ¿Atr-L ∈ ($\phi \cap$ Atr-L)? NO

- Cod-M → Atr-M

  Esta dependencia es similar a la anterior, luego el resultado es el mismo, no se trata de una dependencia redundante.

- Cod-L →→ b y Cod-L, Cod-M →→ a

  Para determinar si una dependencia multivaluada es o no consecuencia lógica de un conjunto de dependencias, es necesario calcular el cierre del conjunto de dependencias y comprobar si dicha dependencia pertenece o no a ese cierre. Sin embargo, el coste y la complejidad del cálculo del cierre de las dependencias multivaluadas es mucho mayor que el de las dependencias funcionales, por lo que no se desarrolla en este libro, afirmando que no existen dependencias multivaluadas redundantes en R.

En los problemas de este libro no se vuelve a realizar el cálculo de las dependencias redundantes mediante este algoritmo, ya que es poco intuitivo. Se aconseja utilizar el algoritmo solamente cuando existe un gran número de atributos y de dependencias, pero cuando el problema es sencillo, es más intuitivo utilizar el siguiente método:

Una dependencia (*dp*) es redundante cuando la dependencia pertenece al cierre de DEP' = DEP-*dp*. Para hallar si alguna de las dependencias de DEP es redundante, hay que comprobar para cada uno de los elementos de DEP si se puede deducir a partir de DEP'.

Como se ha comprobado por el algoritmo que no existen dependencias redundantes, no se vuelve a comprobar por este método.

A partir de todos estos cálculos se puede concluir que el conjunto mínimo de dependencias es el siguiente:

DEP^m = DEP = {

(1) *Cod-L* →*Atr-L*,

(2) *Cod-M* →*Atr-M*,

(3) *Cod-L* →→ *b*,

(4) *Cod-L, Cod-M* →→ *a*

}

CLAVE DE LA RELACIÓN

Claves candidatas

El primer paso es calcular posibles superclaves para la relación R. Una superclave es un subconjunto del conjunto de atributos no vacío de la relación R (AT) tal que determina a todos los atributos de la relación.

Es clave candidata (CC) toda superclave que cumple que no existe ningún subconjunto estricto de la misma que sea a su vez superclave.

Para calcular si un descriptor (subconjunto de atributos de AT) es clave ndidata, el algoritmo se basa en el cálculo de su cierre transitivo.

En primer lugar, se calcula el cierre transitivo de todos los atributos de la ación:

$Cod\text{-}L^+ = \{Cod\text{-}L, Atr\text{-}L\}$

$Atr\text{-}L^+ = \{Atr\text{-}L\}$

$Cod\text{-}M^+ = \{Cod\text{-}M, Atr\text{-}M\}$

$Atr\text{-}M^+ = \{Atr\text{-}M\}$

$a^+ = \{a\}$

$b^+ = \{b\}$

Hasta el momento, ningún atributo simple implica (mediante los axiomas de rmstrong) a todos los atributos de la relación (AT), por lo que se realizan grupos de tributos que puedan determinar a todo el conjunto AT, para ello, normalmente se ligen aquellos cuyos cierres tienen una cardinalidad mayor:

- $(Cod\text{-}L, Cod\text{-}M)^+ = \{Cod\text{-}L, Cod\text{-}M, Atr\text{-}L, Atr\text{-}M\}$

Como aún no se han incluido los atributos *a* y *b* de AT, y ya que se trata de atributos independientes (no se encuentran implicados funcionalmente por ningún otro atributo), éstos tendrán que formar parte de cualquier clave candidata.

- $(Cod\text{-}L, Cod\text{-}M, a, b)^+ = \{Cod\text{-}L, Cod\text{-}M, Atr\text{-}L, Atr\text{-}M, a, b\} = AT$

Por lo tanto, el descriptor $\{Cod\text{-}L, Cod\text{-}M, a, b\}$, al implicar a todo AT, es una superclave. Además es una clave candidata, ya que ningún subconjunto de la misma es superclave. De hecho, se trata de la única clave candidata que existe en la relación R.

Clave = {Cod-L, Cod-M, a, b}

Forma Normal de la relación

- **1FN** (1ª forma normal)**:** Una relación se encuentra en primera forma normal cuando no admite grupos repetitivos como valor de un atributo en un registro. En la relación R no existen grupos repetitivos al tratarse de una relación (propiedad inherente al modelo relacional).
- **2FN:** La relación R no se encuentra en 2FN, ya que existen atributos no principales que no dependen totalmente de la clave. Por ejemplo: en la dependencia $\{Cod\text{-}L \rightarrow Atr\text{-}L\}$, *Atr-L* depende de *Cod-L*, que, aunque forma parte de la clave, no es clave de la relación.

Normalización por análisis

Se normaliza la relación universal R mediante el proceso de descomposición de relaciones hasta llegar al menos a 4FN (objetivo del enunciado).

Se realiza el proceso de descomposición paso a paso, construyendo a su vez el árbol de descomposición:

Paso 1: Recubrimiento minimal (calculado anteriormente)

$DEP^m = \{ Cod\text{-}L \rightarrow Atr\text{-}L,$

$Cod\text{-}M \rightarrow Atr\text{-}M,$

$Cod\text{-}L \rightarrow\rightarrow b,$

$Cod\text{-}L, Cod\text{-}M \rightarrow\rightarrow a\}$

Paso 2: Descomposición de R

Se parte de la relación universal (R), de la que se conocen los siguientes datos:

- Clave: {Cod-L, Cod-M, a, b}
- FN en la que se encuentra: 1FN
- Atributos principales (AP): todos aquellos que forman parte de alguna de las claves = $\{Cod\text{-}L, Cod\text{-}M, a, b\}$
- Atributos no principales (ANP): todos aquellos que no forman parte de ninguna de las claves = $\{Atr\text{-}L, Atr\text{-}M\}$
- $DEP^m = DEP$

Se agrupan todas las dependencias funcionales que poseen el mismo implicante. En este caso no existen dos dependencias funcionales con el mismo implicante, por lo que no se realiza ningún grupo de dependencias.

Se elige una de las DF con cuidado de que el atributo implicado no participe en ninguna otra dependencia de DEP, por ejemplo, se opta por la dependencia $Cod\text{-}L \rightarrow Atr\text{-}L$, ya que *Atr-L* no participa en ninguna otra dependencia de R.

Se crea una nueva relación ($\mathbf{R_1}$) que tendrá como dependencias la elegida y como atributos todos los que participen en la misma, y otra relación temporal necesaria para continuar la descomposición ($\mathbf{R_1}$'), con el resto de las dependencias funcionales y los atributos de R menos los no principales de la relación R_1, es decir, menos el atributo *Atr-L*. Puede observarse la descomposición gráficamente en la figura 3.5.

R_1 AT = {Cod-L, Atr-L}
DEP = {Cod-L → Atr-L}

R_1' AT = {Cod-L, Cod-M, Atr-M, a, b}
DEP = {Cod-M → Atr-M, Cod-L →→ b, Cod-L, Cod-M →→ a}

Figura 3.5. Descomposición de R

A continuación se calcula la forma normal en la que se encuentra la relación R_1. ra ello, primero se hallan las claves candidatas:

La clave de la relación R_1 es *Cod-L*, ya que $(Cod\text{-}L)^+ = AT_{R1}$, y por lo tanto es a superclave, y es mínima, por lo tanto se trata de una clave candidata.

El cálculo de la forma normal en la que se encuentra la relación se detalla a ntinuación:

1FN: Sí, al tratarse de una relación

2FN: Sí, ya que el único atributo no principal (*Atr-L*) que existe en R1 depende funcionalmente de forma completa respecto de la clave, ya que la clave está compuesta por un único atributo (*Cod-L*).

3FN: Sí, ya que no existe ningún atributo no principal que depende transitivamente de alguna clave, ya que sólo existe una dependencia en la relación.

FNBC: Sí, ya que todo determinante (o implicante) es clave candidata, puesto que *Cod-L* es el único determinante de R_1 y se trata de la única clave de la relación.

4FN: Sí, ya que no existen dependencias multivaluadas en R_1.

5FN: Sí, ya que no existen dependencias en combinación en R_1.

Por otro lado, la clave de R_1', es la misma que la de R, donde ahora sólo ha esaparecido una de las dependencias funcionales: se sabe que *Cod-M, Cod-L, a* y *b* o se encuentran implicados por ningún otro atributo de la relación, por lo que tendrán que formar parte de cualquier clave candidata. También se sabe que $(Cod\text{-}M)^+ = Atr\text{-}M$, por lo que el conjunto {*Cod-M, Cod-L, a, b*} implica a todos los atributos de R_1', y por lo tanto es clave candidata.

Con todo esto se puede concluir que:

- Clave(R_1') = Clave(R) = {*Cod-L, Cod-M, a, b*}

El cálculo de la forma normal en la que se encuentra la relación se detalla a continuación:

- 1FN: Sí, al tratarse de una relación
- 2FN: No, ya que aún existe un atributo no principal, *Atr-M*, que depende parcialmente de la clave (sólo depende de *Cod-M*).

De esta forma, y resumiendo todos los cálculos anteriores, el primer nivel del árbol de descomposición se representa gráficamente en la figura 3.6.

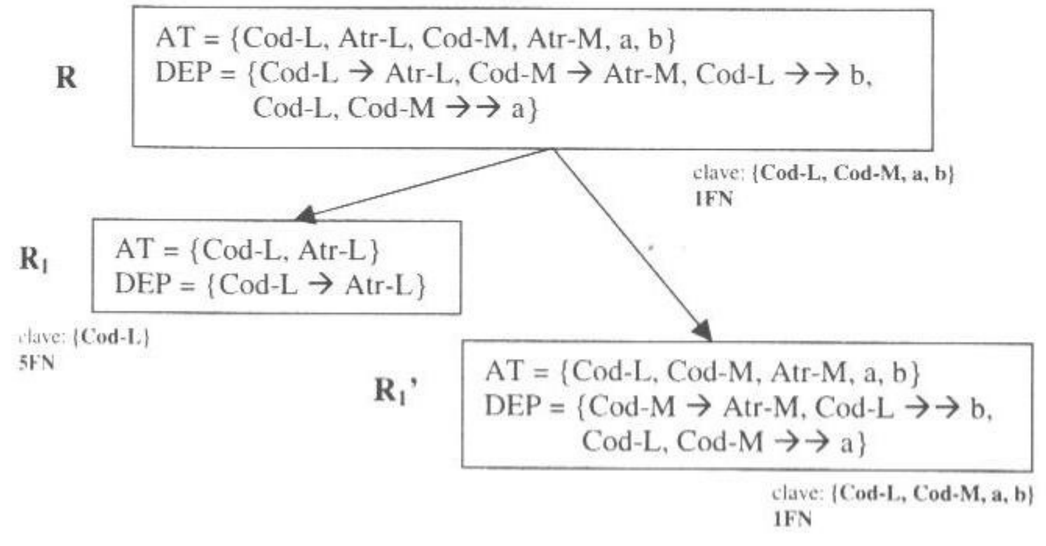


Figura 3.6. Primer nivel del árbol de descomposición

Paso 3: Descomposición de R_1'

Al igual que se hizo con la descomposición de R, se elige una de las dependencias funcionales de R_1', en este caso la única que queda (*Cod-M →Atr-M*), y se crea otra nueva relación (R_2), que tendrá como atributos los que participan en la dependencia funcional y como dependencia la elegida. A su vez se crea una nueva relación temporal para poder continuar con la descomposición (R_2'), que mantendrá los atributos de R_1' menos los atributos implicados en R_2 y las dependencias de R_1' menos las de R_2. Estas relaciones aparecen gráficamente en la figura 3.7.

La relación R_2 es muy parecida a R_1, y por tanto el cálculo de sus claves y de la forma normal en la que se encuentra son similares. De esta forma:

- Clave(R_2): {*Cod-M*}
- Se encuentra en 5FN

R_2 AT = {Cod-M, Atr-M}
DEP = {Cod-M → Atr-M}

R_2' AT = {Cod-L, Cod-M, a, b}
DEP = {Cod-L →→ b, {Cod-L, Cod-M} →→ a}

Figura 3.7. Descomposición de R_1

El cálculo de la clave de la relación R_2' es el siguiente: como ocurría en la ción R_1', al no estar implicados funcionalmente *Cod-L, Cod-M, a* y *b* por ningún buto, han de formar parte de toda clave candidata, y, por lo tanto:

Clave(R_2') = Clave(R_1') = Clave(R) = {*Cod-L, Cod-M, a, b*}

Por otro lado, R_2' sólo posee dependencias multivaluadas, por lo que la forma mal en la que se encuentra ha de ser al menos en la FNBC. A continuación se prueba si se cumplen la cuarta y quinta forma normal:

4FN: No, ya que existen dependencias multivaluadas no determinadas por claves candidatas, por ejemplo *b* está determinado por el atributo *Cod-L*, no por toda la clave {*Cod-L, Cod-M, a, b*}.

El árbol de descomposición hallado hasta el momento aparece en la figura 3.8.

o 4: Descomposición de R_2'

El tratamiento de las dependencias multivaluadas en el algoritmo de composición es similar al de las dependencias funcionales. Se intenta descomponer nero agrupando las dependencias con el mismo implicante y después eligiendo po a grupo hasta que no se puedan descomponer más sin pérdida de dependencias.

En la relación R_2' se puede elegir como primera opción, por ejemplo, la pendencia multivaluada *Cod-L* →→ *b*, de forma que dan lugar a dos nuevas aciones, la relación R_3, con los atributos de la dependencia elegida, y la relación , que contiene el resto de los atributos, que todavía queda por descomponer (de ma similar a apartados anteriores). Gráficamente, estas dos nuevas relaciones se estran en la figura 3.9.

La clave de la relación R_3 es {*Cod-L, b*}, ya que tanto *Cod-L* como *b* son atributos lependientes en la relación

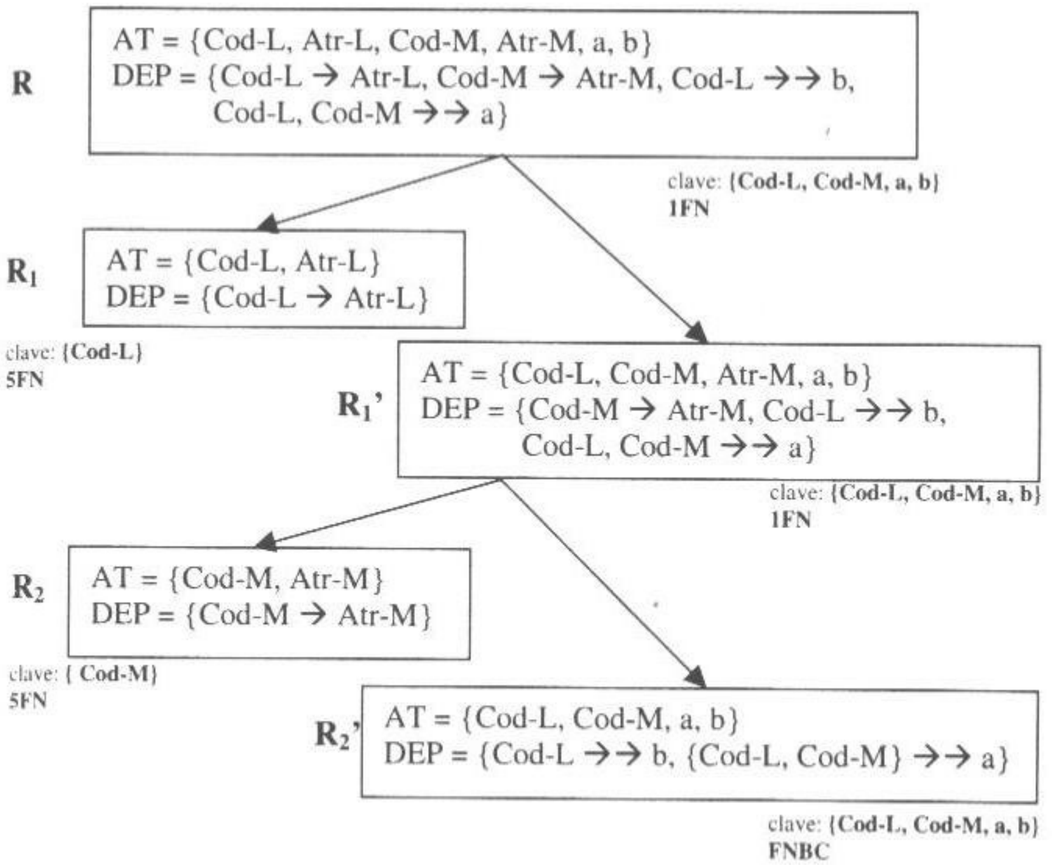


Figura 3.8. Árbol de descomposición parcial

La relación R_3 sí se encuentra en 4FN, ya que se cumple que la única dependencia multivaluada de la relación viene determinada por la clave, teniendo en cuenta que *b* se multidetermina a sí misma (*b* → *b* de forma trivial y por el axioma de Armstrong de replicación *b* →→ *b*). También se encuentra en 5FN, ya que no existen dependencias en combinación.

La clave primaria de la relación R_3' es {*Cod-L, Cod-M, a*}, ya que todos ellos son independientes.

R_3 AT = {Cod-L, b}
DEP = {Cod-L →→ b}

R_3' AT = {Cod-L, Cod-M, a}
DEP = {Cod-L, Cod-M →→ a}

Figura 3.9. Descomposición de R_2

Esta relación se encuentra también a su vez en 5FN, ya que se encuentra en 4FN r el mismo motivo que R_3 y por lo tanto, también en 5FN. Como ya se ha llegado al vel de descomposición pedido en el enunciado (y en este caso el máximo al que se dría llegar), se termina el proceso de descomposición. Se hace notar que, en este so, en el proceso de análisis, no se han perdido dependencias.

El proceso de descomposición se podría haber comenzado en este caso con alquiera de las dependencias (incluso las multivaluadas), ya que no existen atributos nplicados que participasen en otras dependencias, y por lo tanto no existe ningún esgo de pérdidas de dependencias en el proceso de análisis.

El árbol de descomposición final se muestra gráficamente en la figura 3.10.

ormalización por síntesis

El problema también se puede resolver mediante el proceso de síntesis, pero no ompletamente, ya que el algoritmo de síntesis solamente garantiza que la relación ueda estar en FNBC. Por lo tanto, se realiza el proceso de síntesis hasta conseguir ue todas las relaciones obtenidas se encuentren al menos en FNBC. Por último se tiliza el algoritmo de análisis para descomponer las relaciones que aún no se ncuentran en el nivel de normalización deseado.

Al igual que se hizo con el proceso de análisis, se describen paso a paso los letalles del algoritmo:

Paso 1: Recubrimiento minimal (calculado anteriormente)

$DEP^m = \{$

Cod-L → Atr-L,

Cod-M → Atr-M,

Cod-L →→ b,

Cod-L, Cod-M →→ a}

Paso 2: Agrupar las dependencias funcionales que tengan el mismo implicante, unir claves equivalentes y eliminar dependencias transitivas

En este caso no existen dos dependencias funcionales con el mismo implicante, por lo que no se realiza ninguna agrupación. Tampoco existen claves equivalentes ni dependencias transitivas.

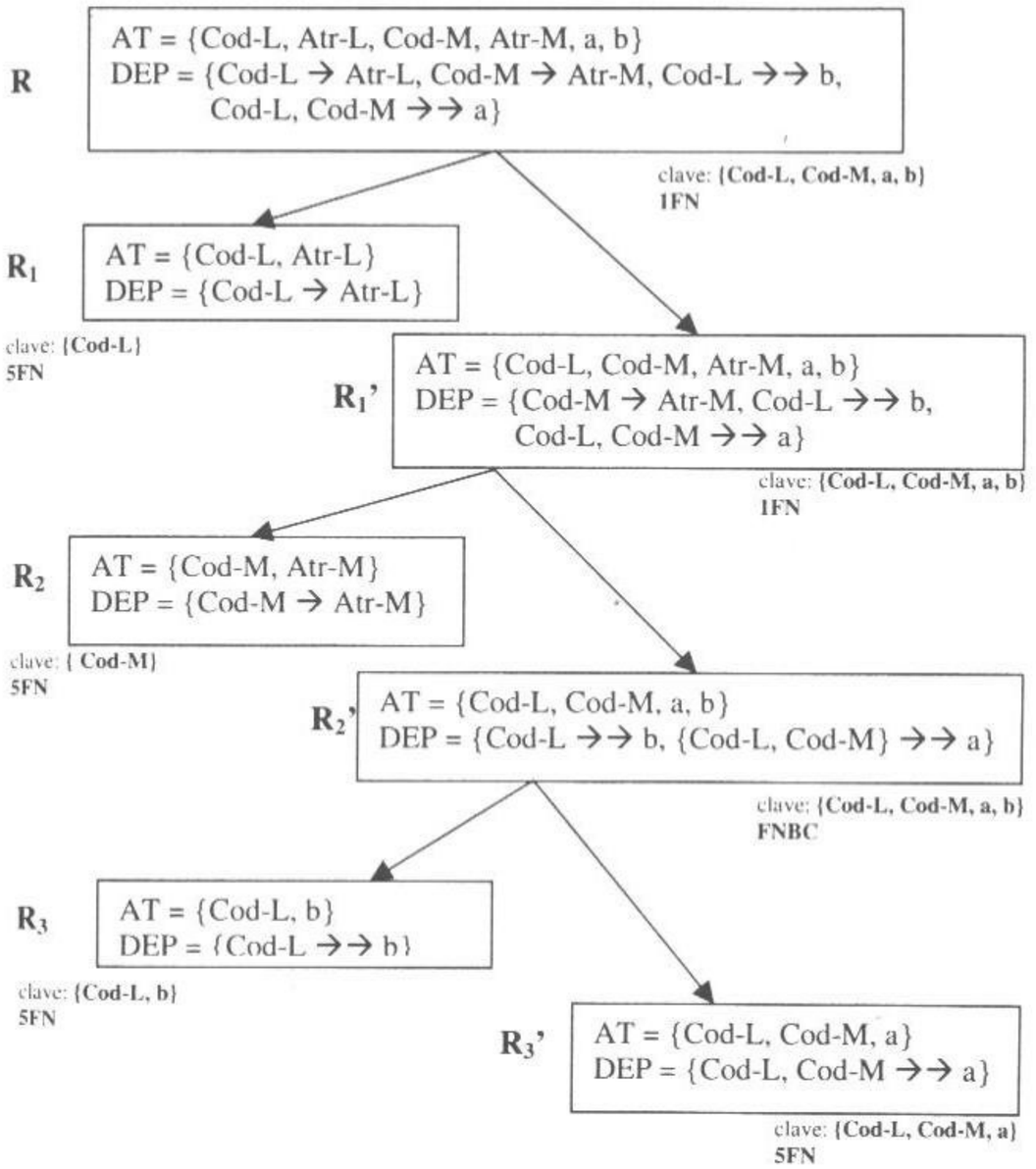


Figura 3.10. Árbol de descomposición

Paso 3: Construir relaciones

Se van a construir nuevas relaciones con los grupos de dependencias funcionales del apartado anterior:

- R_1({Cod-L, Atr-L}, {Cod-L → Atr-L}).
- R_2({Cod-M, Atr-M}, {Cod-M → Atr-M}).

Por otro lado, se crea una nueva relación con todos los atributos que no son aplicados ni implicantes, de forma que se encuentren también en la misma relación s atributos clave de la relación universal:

Raux({Cod-L, Cod-M, a, b}, {Cod-L →→ b; Cod-L, Cod-M →→ a}).

Se puede observar que las relaciones R_1 y R_2 obtenidas mediante el proceso de ntesis son las mismas que se obtuvieron con el proceso de análisis, y que la relación xiliar (Raux) es equivalente a la relación temporal R_2' del proceso de análisis.

Por lo tanto, se sabe que tanto R_1 como R_2 son relaciones que se encuentran en FN, y la relación Raux se encuentra en FNBC.

aso 4: Descomposición de Raux mediante análisis

En este paso se descomponen mediante análisis las relaciones que todavía no se ncuentran en la forma normal deseada (Raux). La descomposición mediante análisis e Raux ya se realizó en el apartado anterior, por lo que se omite en este apartado.

Las relaciones obtenidas en este último paso son las siguientes:

R_3({Cod-L, b}, {Cod-L →→ b}).

R_4({Cod-L, Cod-M, a}, {Cod-L, Cod-M →→ a}).

Para finalizar, se observa que se obtiene el mismo esquema de relaciones ediante el proceso de síntesis y el de análisis, sin pérdida de información ni de ependencias.

ROBLEMA 2

Enunciado

Indicar qué dependencias funcionales existen en el esquema relacional de la gura 3.11, y cuáles son los elementos semánticos del mismo que no se pueden ecoger mediante dependencias funcionales.

Suponiendo que todos los atributos y dependencias del esquema constituyen la elación universal, hallar un recubrimiento minimal y las claves de la misma.

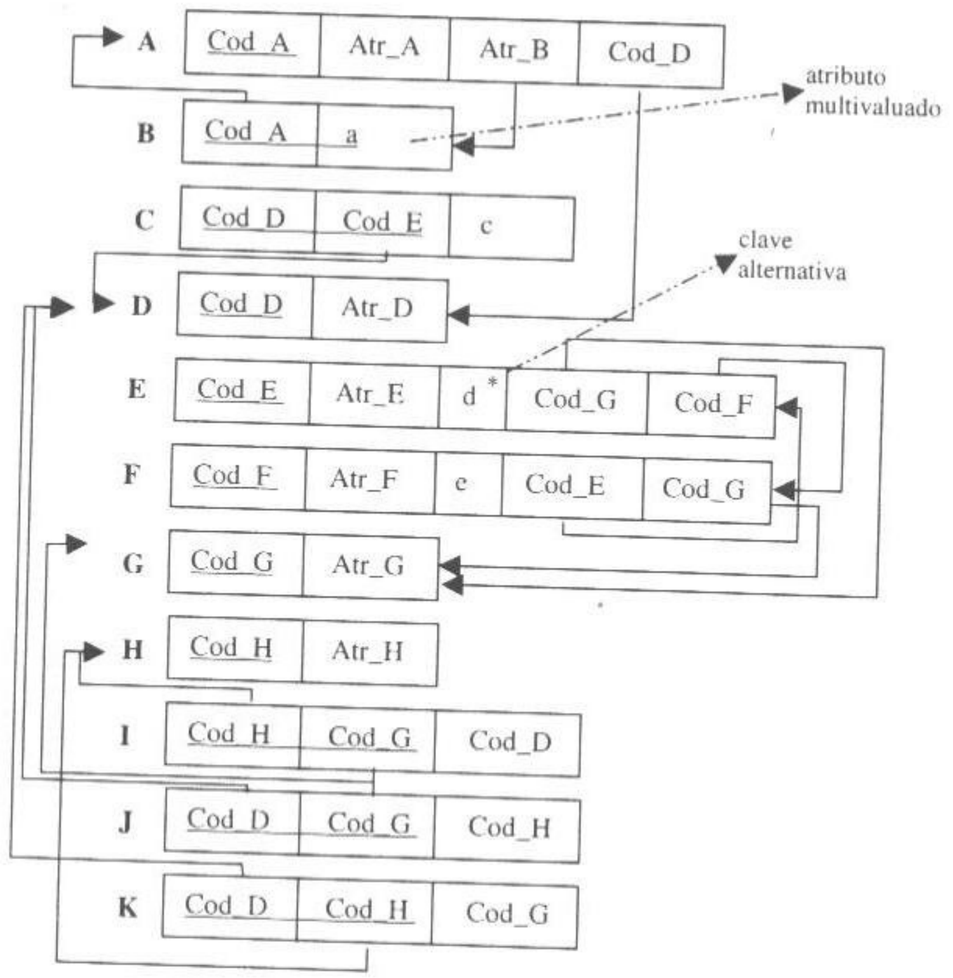


Figura 3.11. Esquema relacional

Discusión del enunciado

DEPENDENCIAS FUNCIONALES

El primer paso es realizar el cálculo de las dependencias funcionales (DF) a partir de las relaciones de la figura 3.11. En este apartado se detalla el procedimiento de búsqueda de las dependencias funcionales a partir de relaciones.

En la relación A se tiene como clave primaria el atributo *Cod_A*. Toda clave primaria determina funcionalmente al resto de los atributos de la relación, y por lo tanto, *Cod_A* determina funcionalmente a *Atr_A*, *Atr-B* y *Cod_D*.

Lo mismo ocurre con las relaciones D, E, F, G, H, donde la clave primaria de cada relación también está compuesta por un único atributo que determina funcionalmente a los atributos no principales de la misma. Y con las relaciones C, I, J

K, donde la clave primaria está formada por dos atributos y también determinan cionalmente al resto.

Por lo tanto, hasta el momento se tienen las siguientes dependencias funcionales:

1) Cod_A → Atr_A, Atr_B, Cod_D

2) Cod_D → Atr_D

3) Cod_E → Atr-E, Cod_G, Cod_F, d

4) Cod_F → Atr_F, Cod_E, Cod_G, e

5) Cod_G → Atr_G

6) Cod_H → Atr_H

7) Cod_E, Cod_D → c

8) Cod_H, Cod_G → Cod_D

9) Cod_D, Cod_G → Cod_H

10) Cod_D, Cod_H → Cod_G

Además, en la relación E se tiene una clave primaria (*Cod_E*) y una alternativa *d*). En teoría de la normalización no se distingue entre estos dos tipos de clave, ambas son claves candidatas, por lo que *d* también podría actuar como clave de la relacion determinando al resto de los atributos de la misma. De esta forma también se puede decir que *d* determina funcionalmente a *Cod_E* por ser clave alternativa:

11) d → Cod_E

Por último, la relación B es distinta a todas las demás, ya que en este caso *a* es un atributo multivaluado por tanto, existe una dependencia multivaluada: *Cod_A* multidetermina a *a* (*Cod_A →→ a*) y por lo tanto, esta restricción semántica no se puede recoger mediante dependencias funcionales.

Por lo tanto, se puede concluir que la relación universal, R(AT, DEP) construida a partir del esquema relacional de la figura 3.11, se define como:

AT = {*Cod_A, Atr_A, Atr_B, Cod_D, Atr_D, Cod_E, Atr_E, Cod_F, Atr_F, Cod_G, Atr_G, Cod_H, Atr_H, a, c, d, e*}

DEP = {

(1) *Cod_A → Atr_A, Atr_B, Cod_D*

(2) *Cod_D → Atr_D*

(3) *Cod_E → Atr_E, Cod_G, Cod_F, d*

(4) *Cod_F → Atr_F, Cod_E, Cod_G, e*

(5) *Cod_G → Atr_G*

(6) *Cod_H → Atr_H*

(7) *Cod_E, Cod_D → c*

(8) *Cod_H, Cod_G → Cod_D*

(9) *Cod_D, Cod_G → Cod_H*

(10) *Cod_D, Cod_H → Cod_G*

(11) *d → Cod_E*

}

RECUBRIMIENTO IRREDUNDANTE (O MINIMAL)

En este apartado se halla el conjunto de dependencias mínimo o irredundante de entre todas las dependencias del conjunto DEP. Para que un conjunto de dependencias sea mínimo ha de cumplir que:

- Todas sus dependencias sean elementales, para ello se han de cumplir las siguientes condiciones:
  - Han de ser plenas: es decir, que no exista un subconjunto del implicante que determine al implicado. En las dependencias anteriores (en general, X→Y) todas las dependencias son plenas, ya que no existe en ellas ningún atributo (At) tal que (X-At)→Y
  - Han de cumplir que el implicado sea un único atributo. En las dependencias del apartado anterior, por comodidad se han resumido varias dependencias en una, de forma que, formalmente en la definición de recubrimiento irredundante, habría que desarrollar cada una de las dependencias. Un ejemplo es la primera dependencia (*Cod_A → Atr_A, Atr_B, Cod_D*), en este caso existen tres dependencias resumidas en una. Si se desarrolla esta

dependencia, resultan las siguientes dependencias funcionales (*Cod_A →Atr_A; Cod_A →Atr_B; Cod_A →Cod_D*). Para continuar con el ejercicio, por comodidad, se sigue utilizando la versión resumida de las dependencias funcionales.

- Han de ser no triviales: es decir, que no exista ninguna dependencia donde un implicante se determine a sí mismo. Esta condición también se cumple en las dependencias anteriores.

Que no existan atributos extraños:

Se comprueba que en ninguna dependencia de DEP existen atributos extraños. Como se indicó en el problema anterior, un atributo *A* es extraño si dada la dependencia $X \rightarrow Y$ de DEP, la dependencia $(X\text{-}A) \rightarrow Y$ se encuentra en DEP^+.

No se detalla el cálculo de atributos extraños en este problema, ya que no existe ninguno, pero en problemas posteriores se puede observar este cálculo.

Que no existan dependencias redundantes:

Una dependencia (dp) es redundante cuando la dependencia pertenece al cierre de DEP' = DEP-dp.

Para hallar si alguna de las dependencias de DEP es redundante, hay que comprobar para cada uno de los elementos de DEP si se puede deducir a partir de DEP'.

El cálculo de redundancia de la primera dependencia (*Cod_A → Atr_A*) se detalla a continuación:

- Se halla $(Cod_A)^+_{DEP'}$ = {Cod_A, Atr_B, Cod_D, Atr_D}

  - Dependencia 1 (1): Cod_A → Atr_B, Cod_D, luego $(Cod_A)^+_{DEP'}$ = {Cod_A, Atr_B, Cod_D}

  - (2): Cod_D → Atr_D, luego $(Cod_A)^+_{DEP'}$ = {Cod_A, Atr_B, Cod_D. Atr_D}

- Como no se determina *Atr_A* a partir de DEP', la dependencia no es redundante.

No se explica detalladamente todo el cálculo para ver si las dependencias son irredundantes o no, debido a que es repetitivo, pero sí se detalla cómo se detecta la única dependencia redundante del conjunto DEP:

La dependencia *Cod_E → Cod_G* es una dependencia redundante:

- $(Cod_E)^+_{DEP'}$ = {Cod_E, Atr_E, d, Cod_F, Atr_F, Cod_G, e, Atr_G}

  - (3): Cod_E → Atr-E, Cod_F, d, luego $(Cod_E)^+_{DEP'}$ = {Cod_E, Atr_E. Cod_F, d}

  - (4): Cod_F → Atr_F, Cod_E, Cod_G, e, luego $(Cod_E)^+_{DEP'}$ = {Cod_E, Atr_E, Cod_F, d, Atr_F, Cod_G, e}

  - (5): Cod_G → Atr_G, luego $(Cod_E)^+_{DEP'}$ = {Cod_E, Atr_E, Cod_F, d, Atr_F, Cod_G, e, Atr_G}

- En este caso, como *Cod_G* se determina a partir de DEP', la dependencia sí es redundante.

Por lo tanto, se puede concluir que el conjunto de dependencias siguiente es un conjunto de dependencias mínimo (DEP^m):

DEP^m = {

(1) *Cod_A → Atr_A, Atr_B, Cod_D*

(2) *Cod_D → Atr_D*

(3) *Cod_E → Atr_E, Cod_F, d*

(4) *Cod_F → Atr_F, Cod_E, Cod_G, e*

(5) *Cod_G → Atr_G*

(6) *Cod_H → Atr_H*

(7) *Cod_E, Cod_D → c*

(8) *Cod_H, Cod_G → Cod_D*

(9) *Cod_D, Cod_G → Cod_H*

(10) *Cod_D, Cod_H → Cod_G*

(11) *d → Cod_E*

}

CLAVES CANDIDATAS

El cálculo del conjunto de claves candidatas de DEP^m se basa en el cálculo del cierre transitivo de los determinantes.

En primer lugar, el atributo *a* es un atributo multivaluado y además no se cuentra determinado funcionalmente por ningún conjunto de atributos de la relación. por ello que ha de formar parte de todas las claves candidatas de R.

Por otro lado también se tiene que *Cod_E* → *d*; *d* → *Cod_E*; *Cod_E* → *Cod_F*; *d_F* → *Cod_E*, por lo que *Cod_E*, *Cod_F* y *d* son atributos equivalentes y su cierre nsitivo será el mismo.

Se calcula el cierre de cada uno de los atributos individuales de la relación iversal:

$(Cod_A)^+_{DEP}$= {Cod_A, Atr_A, Atr_B, Cod_D, Atr_D}

Cod_D se encuentra implicado funcionalmente por *Cod_A*, luego no se calcula su cierre, al estar implícito en el cálculo de $(Cod_A)^+_{DEP}$.

$(Cod_E)^+_{DEP}$= {Cod_E, Atr_E, d, Cod_F, Atr_F, Cod_G, e, Atr_G}

Cod_F y *d* tienen el mismo cierre que *Cod_E*, y además se encuentran implicados por él, luego no se calcula su cierre. A *Cod_G* también se encuentra implicado funcionalmente por *Cod_E*, luego tampoco se calcula su cierre.

$(Cod_H)^+_{DEP}$= {Cod_H, Atr_H}

Hasta el momento no existe ningún determinante que implique funcionalmente a dos los atributos de la relación (AT), de modo que se realiza el cálculo del cierre de njuntos de atributos, normalmente eligiendo aquellos cuya cardinalidad del cierre es ayor (*Cod_A* y *Cod_E*):

$(Cod_A, Cod_E)^+_{DEP}$= {Atr_A, Atr_B, Cod_D, Atr_D, Atr_E, d, Cod_F, Atr_F, Cod_G, e, Atr_G, Cod_H, Atr_H, c}, ya que:

- $(Cod_A, Cod_E)^+_{DEP}$= $(Cod_A)^+_{DEP}$ U $(Cod_E)^+_{DEP}$ U (cierre de las combinaciones de los elementos de $(Cod_A)^+_{DEP}$ y $(Cod_E)^+_{DEP}$), como por la dependencia (7) (*Cod_E, Cod_D* → *c*) se implica al atributo *c*, y con la dependencia (9) (*Cod_D, Cod_G* → *Cod_H*) se implica a *Cod_H*, (Cod_A, $Cod_E)^+_{DEP}$ implica a todos los atributos de la relación menos al atributo *a*.

Por lo tanto se puede concluir que el conjunto CC1 = {*Cod_A, Cod_E, a*} es una perclave de la relación. Como además no existe ningún subconjunto de CC1 que mplique a todos los atributos de la relación, se puede concluir que es clave candidata

Como *Cod_E*, *d* y *Cod_F* son equivalentes, también los conjuntos {*Cod_A, Cod_F, a*} y {*Cod_A, d, a*} son claves candidatas, y en este caso, estas tres claves son as únicas de la relación.

En el desarrollo completo de este problema se debería seguir el proceso para comprobar que no existe ninguna otra clave candidata para esta relación. Se deja como ejercicio al lector esta comprobación.

Para finalizar, se indican cuáles son los atributos principales y no principales de la relación R:

- Atributos principales (forman parte de alguna clave candidata): {*Cod_E, Cod_F, d, Cod_A, a*}
- Atributos no principales (no forman parte de ninguna clave candidata): {Atr_A, Atr_B, Cod_D, Atr_D, Atr_E, Atr_F, Cod_G, Atr_G, Cod_H, Atr_H, c, e}

PROBLEMA 3

Enunciado

Dado el siguiente esquema de relación R (AT, DEP), donde:

AT = {*O, R, U, V, W, X, Y, Z*} y

DEP = {

(1) *X,Y* → *Z*

(2) *Z* → *U*

(3) *X,Y,Z* → *V*

(4) *R* → *X*

(5) *X* → *R*

(6) *W* → *O*

(7) *O* → *W*

}

Hallar el cierre transitivo del descriptor {*X,Y*}, determinar las claves de R, indicar n qué forma normal se encuentra R y descomponerla en un conjunto de relaciones en NBC indicando las claves primarias de las relaciones resultantes.

Discursión del enunciado

CIERRE TRANSITIVO DE XY

El cálculo del cierre transitivo de un descriptor se basa en los axiomas de Armstrong, y devuelve como resultado el conjunto de los atributos de la relación que son implicados por el descriptor directa o indirectamente (mediante el axioma de transitividad).

En este, como en los demás problemas de normalización, se utiliza indistintamente dos notaciones para los descriptores. Por poner un ejemplo, para que se vea con más claridad, el descriptor formado por los atributos *X* y *Y* se puede denotar por {*X*,*Y*} o simplemente *XY*.

Se muestra a continuación el resultado de cada iteración del cálculo del cierre transitivo de XY según las dependencias de DEP:

- $XY^+ = \{X, Y\}$
- Dependencia 1 (1): $X,Y \rightarrow Z$, luego $XY^+ = \{X, Y, Z\}$
- (2): $Z \rightarrow U$, luego $XY^+ = \{X, Y, Z, U\}$
- (3): $X,Y,Z \rightarrow V$, luego $XY^+ = \{X, Y, Z, U, V\}$
- (5): $X \rightarrow R$, luego $XY^+ = \{X, Y, Z, U, V, R\}$

No existe ninguna otra dependencia de DEP que modifique XY^+, por lo tanto, se concluye que:

- $XY^+ = \{X, Y, Z, U, V, R\}$

CLAVES CANDIDATAS DE R

El algoritmo de cálculo de las claves candidatas de R también se basa en el cálculo del cierre transitivo de descriptores. Se detalla el procedimiento a continuación:

En primer lugar, se halla el cierre transitivo de todos los atributos de R:

- $X^+ = \{X, R\}$
- $R^+ = \{X, R\}$, los atributos X y R son equivalentes.
- $Y^+ = \{Y\}$
- $Z^+ = \{Z, U\}$
- $U^+ = \{U\}$

- $V^+ = \{V\}$
- $W^+ = \{W, O\}$
- $O^+ = \{W, O\}$, los atributos W y O también son equivalentes.

Hasta el momento, no existe ningún atributo simple que determine a todos los atributos de R, luego se realizan conjuntos de atributos que puedan determinarlos y se halla su cierre transitivo:

- En el apartado anterior ya se ha calculado el cierre transitivo de *XY* ($XY^+ = \{X, Y, Z, U, V, R\}$), pero los atributos *W* y *O* se encuentran incluidos en el conjunto, luego se incluye, por ejemplo, *W*, ya que $W \rightarrow O$. Por lo tanto, $XYW^+ = \{X, Y, Z, U, V, R, W, O\}$ = AT, de forma que el conjunto CC1 = {*X*,*Y*,*W*} es una superclave. Como ningún subconjunto de CC1 implica a todo AT, CC1 es clave candidata.
- Por otro *W* es equivalente a *O*, por lo que el conjunto CC2 = {*X*,*Y*,*O*} es otra clave candidata.
- De igual forma, los atributos *R* y *X* son equivalentes, por lo que se puede concluir que los conjuntos CC3 = {*R*, *Y*, *W*}y CC4 = {*R*, *Y*, *O*} también son claves candidatas.

FORMA NORMAL DE R

Para calcular en qué forma normal se encuentra la relación R, se comprueban las restricciones que cumple R, comenzando por el nivel de normalización menos restrictivo:

- 1FN: R sí se encuentra en primera forma normal al no existir grupos repetitivos (restricción inherente de toda relación).
- 2FN: R no se encuentra en 2FN, ya que existen atributos no principales que dependen funcionalmente de un subconjunto de la clave. Por ejemplo, el atributo no principal *U* depende funcionalmente de *Z*, pero no de la clave completa de R.

Por tanto, se puede concluir que la relación R se encuentra en primera forma normal (1FN).

DESCOMPOSICIÓN DE R

En este apartado se realiza la descomposición de R mediante el algoritmo de Tesis, ya que en el enunciado sólo se pide llegar a FNBC y este algoritmo lo permite.

De la misma forma se puede realizar la descomposición mediante el algoritmo de álisis, proporcionando una solución igualmente correcta. Se deja como ejercicio a el lector la descomposición de R mediante análisis.

El algoritmo de síntesis parte del conjunto de dependencias de la relación R EP) y realiza los siguientes pasos:

Hallar el conjunto de dependencias mínimo de DEP (DEP^m).

- Para ello se ha de comprobar que todas las relaciones son elementales (con un único atributo implicado, donde el implicado no depende de ningún subconjunto del determinante). En la dependencia $X,Y,Z \rightarrow V$ no se cumple la segunda condición de dependencia elemental, ya que $X,Y \rightarrow Z$, de forma que V depende de Z o de $\{X,Y\}$, luego se puede concluir que $Z \rightarrow V$.

Debido a esto, el conjunto de dependencias mínimo hasta el momento es el siguiente:

$DEP1^m$ = {

(1) $X,Y \rightarrow Z$

(2) $Z \rightarrow U$

(3) $Z \rightarrow V$

(4) $R \rightarrow X$

(5) $X \rightarrow R$

(6) $W \rightarrow O$

(7) $O \rightarrow W$

}

- Por otro lado, también se ha de comprobar que no existen atributos extraños. Se dice que un atributo A perteneciente a X es extraño en la dependencia X→Y si la dependencia (X-A)→Y se encuentra en DEP^+. Si se realiza el cálculo en este problema, se puede comprobar que en ningún elemento del conjunto $DEP1^m$ existen atributos extraños. Se deja esta commprobación para el lector.

- Por último se comprueba que no existen atributos redundantes. Para hallar si alguna de las dependencias (dp) de $DEP1^m$ es redundante, se comprueba que

no se puede deducir a partir del resto de las dependencias de $DEP1^m$ ($DEP' = DEP1^m - dp$).

Por ejemplo, para la primera de las dependencias ($X,Y \rightarrow Z$):

- Se halla $(XY)^+_{DEP'} = \{X, Y, R\}$:

Según la dependencia 5 de DEP' ($X \rightarrow R$), el atributo X determina funcionalmente al atributo R luego $(XY)^+_{DEP'} = \{X, Y, R\}$

- Como el atributo Z no se determina a partir de DEP', la dependencia no es redundante.

En este ejercicio no se desarrolla completamente el cálculo de las dependencias redundantes, sino que se concluye afirmando que no existe ninguna dependencia redundante en DEP, luego $DEP^m = DEP1^m$

- Agrupar las dependencias que tengan el mismo implicante, uniendo las claves equivalentes.

  - En este caso no existen dos dependencias que tengan el mismo implicante, luego no se realizarán grupos de dependencias con el mismo implicante.
  - Por otro lado sí que existen atributos equivalentes:
    - W es equivalente a O ($W=O$): luego se agrupan las dependencias $\{W \rightarrow O, O \rightarrow W\}$
    - $X=R$: donde también se agrupan las dependencias $\{X \rightarrow R, R \rightarrow X\}$

- Construir relaciones con los grupos de dependencias:

  - R1 ({X, Y, Z}, {X, Y→Z})
  - R2 ({Z, U}, {Z→U})
  - R3 ({Z, V}, {Z→V})
  - R4 ({W, O}, {W→O; O→W})
  - R5 ({X, R}, {X→R; R→X})

- Por último se construye una relación con los atributos clave de R.

  - R6 ({X, Y, W}, {})

FORMA NORMAL DE (R1...R6)

Para finalizar la resolución del problema falta aún indicar en qué forma normal se encuentran las seis relaciones resultantes de la descomposición de R (R1...R6) y cuáles son sus claves candidatas.

- R1 ($\{X, Y, Z\}$, $\{X, Y \rightarrow Z\}$):
  - Clave: $\{X, Y\}$, se halla de forma inmediata.
  - Atributos principales: $\{X, Y\}$; atributos no principales: $\{Z\}$.
  - FN de R1 (5FN): Se encuentra en 1FN al no tener grupos repetitivos; en 2FN también, ya que todos los atributos no principales dependen de la clave de la relación; en 3FN, ya que no existen dependencias transitivas; en FNBC al cumplirse que todo determinante es clave candidata, y, por último, se encuentra en 4FN y 5FN al no existir dependencias multivaluadas ni en combinación.

R2 ($\{Z, U\}$, $\{Z \rightarrow U\}$):

- Clave: $\{Z\}$.
- Atributos principales: $\{Z\}$; Atributos no principales: $\{U\}$.
- Se encuentra en 5FN siguiendo el mismo razonamiento que en R1.

R3 ($\{Z, V\}$, $\{Z \rightarrow V\}$):

- Clave: $\{Z\}$.
- Atributos principales: $\{Z\}$; Atributos no principales: $\{V\}$.
- Se encuentra en 5FN (similar a R2).

R4 ($\{W, O\}$, $\{W \rightarrow O; O \rightarrow W\}$):

- Claves: $\{W\}$ y $\{O\}$.
- Atributos principales: $\{W, O\}$; Atributos no principales: $\{\}$.
- Se encuentra en 5FN (similar a R3).

- R5 ($\{X, R\}$, $\{X \rightarrow R; R \rightarrow X\}$):
  - Claves: $\{R\}$ y $\{X\}$.
  - Atributos principales: $\{R, X\}$; Atributos no principales: $\{\}$.
  - Se encuentra en 5FN (similar a R4).
- R6 ($\{X, Y, W\}$, $\{\}$):
  - Claves: $\{X, Y, W\}$.
  - Atributos principales: $\{X, Y, W\}$; Atributos no principales: $\{\}$.
  - Se encuentra en 5FN.

De esta forma se puede concluir que se ha llegado a una descomposición de la relación inicial (R), en seis relaciones (R1, ..., R6) que se encuentran en 5FN sin pérdidas de dependencias funcionales ni de información.

PROBLEMA 4

Enunciado

Describir de forma detallada (preferiblemente mediante un diagrama), todos los pasos que permitan determinar el nivel de normalización de un esquema de relación y avanzar lo máximo posible en su normalización mediante el proceso de análisis.

Aplique el proceso descrito anteriormente para determinar el nivel de normalización del siguiente supuesto, así como para obtener un esquema relacional en la forma normal que considere más conveniente.

Supuesto

Se desea diseñar una base de datos en el modelo relacional para una universidad, teniendo los siguientes supuestos semánticos:

Un profesor se identifica por un código de profesor (CP) y todos los profesores tienen nombres (NP) distintos. Un profesor puede tener varios títulos (T) e intervenir en distintos proyectos (P), no exigiéndose ningún título determinado para intervenir en un cierto proyecto.

Cada asignatura (A) tiene un único profesor como responsable, si bien un mismo profesor puede ser responsable de más de una asignatura. Las asignaturas se dividen en uno o más grupos (G). Todo alumno (AL), en cada asignatura, pertenece a un único grupo.

Cada profesor depende siempre y únicamente de un departamento (D). Así ismo, toda asignatura está ligada a un único departamento, el del profesor sponsable de la misma.

iscusión del enunciado

IVEL DE NORMALIZACIÓN

Al enfrentarse a un problema en el que se pide hallar el nivel de normalización de laciones, el primer paso, antes de calcular las claves primarias de la relación, es omprobar que el conjunto de atributos y dependencias está en, al menos, primera rma normal, es decir, que se trata de una relación. Esto se debe a que una de las stricciones inherentes al modelo relacional es que el conjunto de atributos y ependencias se encuentre en 1FN.

Una vez que se tiene la relación en 1FN, se comprueba si se encuentra en los guientes niveles de normalización. Para ello se necesita saber primero cuáles son sus ributos principales y no principales. Por lo tanto, se han de calcular las dependencias ncionales, multivaluadas y de combinación existentes entre sus atributos, el cubrimiento minimal (o irredundante) de la relación, las claves y los atributos incipales y no principales de la misma.

La relación se encontrará en 2FN si se encuentra en 1FN y cumple que cada ributo no principal tiene dependencia funcional completa respecto de cada una de las aves.

Se encuentra en 3FN cuando está en 2FN y no existe ningún atributo no principal ue dependa transitivamente de alguna clave de la relación.

Se encuentra en FNBC cuando todo determinante es clave candidata de la elación.

La relación se encuentra en 4FN cuando está en 3FN y cumple que las únicas ependencias multivaluadas no triviales vienen determinadas por claves candidatas.

Por último, la relación se encuentra en 5FN cuando toda dependencia funcional ultivaluada o de combinación no trivial es consecuencia de las claves candidatas.

Cuando una relación no se encuentra en la forma normal deseada, se puede escomponer (mediante las técnicas de análisis o síntesis) en varias con un nivel de ormalización más alto (aunque en ocasiones esto puede suponer pérdidas de ependencias entre los atributos de la relación).

Para resumir todo lo anterior de forma esquemática, se presenta en forma de iagrama en la figura 3.12 el proceso de cálculo de la forma normal de cualquier elación.

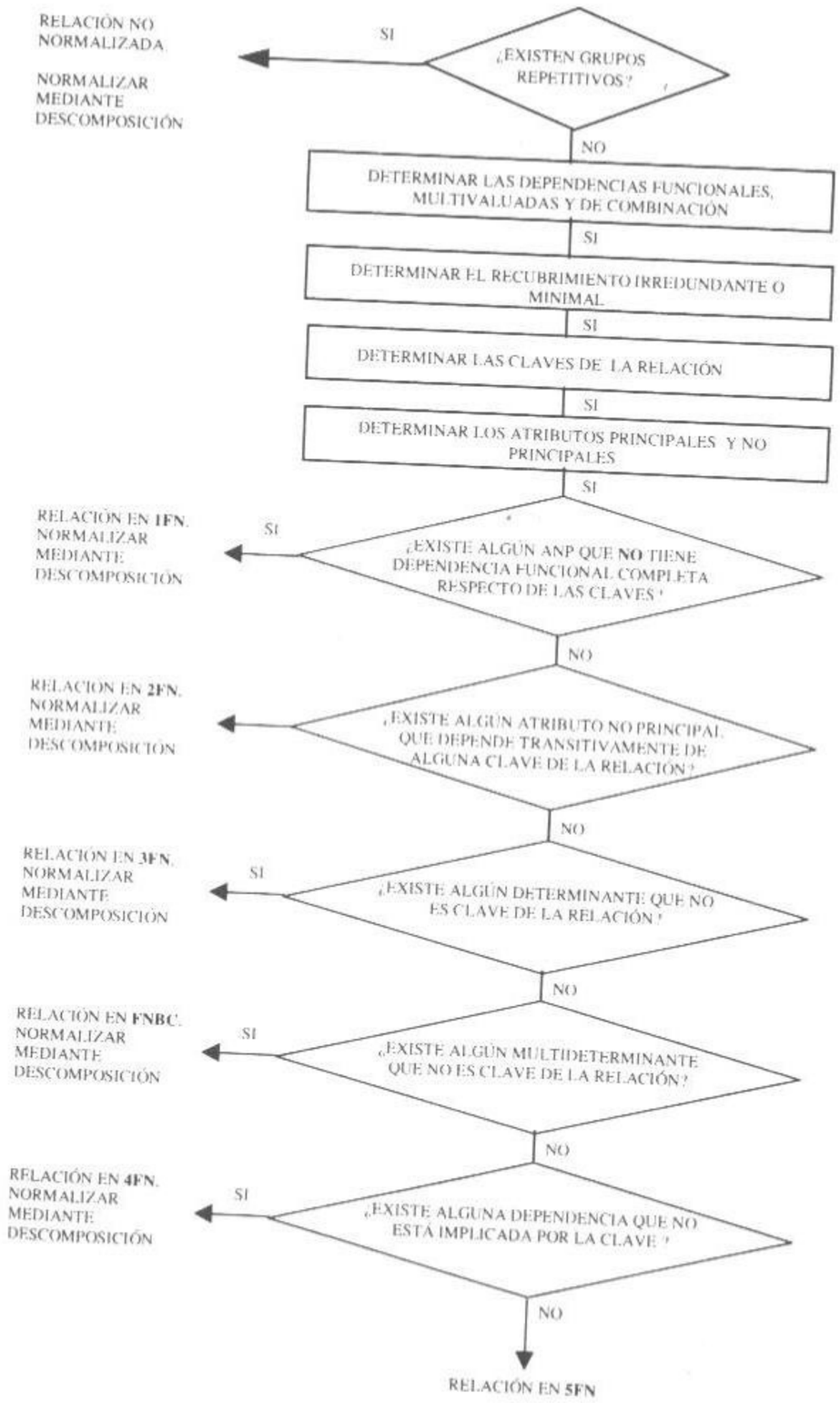


Figura 3.12. Diagrama de comprobación del nivel de normalización

NORMALIZACIÓN DEL SUPUESTO

En este apartado se procede a determinar el nivel de normalización del supuesto del enunciado y se normaliza mediante análisis (o descomposición) hasta la forma normal que se cree más conveniente.

Grupos repetitivos

En el supuesto no existen grupos repetitivos, luego, la relación se encuentra al menos en 1FN.

Dependencias entre atributos

En primer lugar, se parte de una base de datos de la universidad, donde, en principio, solamente existe una relación con todos los atributos y dependencias entre los mismos (relación universal). Para ello, y como se ha visto en ejercicios de otros capítulos, se analizan los atributos de la relación. Como este capítulo se centra en la normalización, no se detalla el proceso de análisis de atributos.

Se analiza el caso, determinando los atributos y las dependencias funcionales, multivaluadas y de combinación que existen entre los atributos de la relación. Para realizar un análisis detallado, se introducen referencias al enunciado.

> *Un profesor se identifica por un código de profesor (CP) y todos los profesores tienen nombres (NP) distintos.*

Esta frase indica que de la entidad *profesor* se desean guardar los atributos *CP* y *NP*. Donde *CP* identifica al profesor, tratándose de su clave primaria, y por tanto determina funcionalmente al nombre:

- $CP \rightarrow NP$

Pero también se indica que no existen dos profesores con el mismo nombre. Por lo tanto, el nombre también determina funcionalmente al código (actuaría como una clave alternativa en el modelo relacional). De esta forma, también se da la dependencia funcional:

- $NP \rightarrow CP$

> *Un profesor puede tener varios títulos (T) e intervenir en distintos proyectos (P), no exigiéndose ningún título determinado para intervenir en un cierto proyecto.*

En este caso, indica que los profesores pueden poseer varios títulos (conjunto bien definido de valores), luego, en este caso no existe ninguna dependencia funcional entre *CP* y *T*, sino que se trata de una dependencia multivaluada. También multivalúa

a los distintos proyectos en los que interviene el profesor. Como no tiene ninguna relación el título y el proyecto para un determinado profesor, se tiene la dependencia multivaluada jerárquica $CP \rightarrow\rightarrow T \mid P$, donde entre el proyecto y el título no existe ninguna dependencia funcional (T -/-> P y P-/->T):

- $CP \rightarrow\rightarrow T \mid P$

> *Cada asignatura (A) tiene un único profesor como responsable, si bien un mismo profesor puede ser responsable de más de una asignatura.*

Por otro lado, para cada asignatura existe un único profesor que se responsabiliza de la misma, por lo tanto, como para cada valor de la asignatura existe un único valor para profesor responsable, existe la dependencia funcional $A \rightarrow CP$. También se da que el profesor puede ser responsable de más de una asignatura, de forma que, para un único valor de P no existe un único valor de A, por lo tanto, no existe la dependencia funcional entre los proyectos y las asignaturas (CP -/-> A):

- $A \rightarrow CP$

> *Las asignaturas se dividen en uno o más grupos (G). Todo alumno (AL), en cada asignatura, pertenece a un único grupo.*

Las asignaturas se dividen en grupos, un grupo sólo puede pertenecer a una asignatura, de forma que, para cada valor de grupo existe un único valor de asignatura, luego existe otra dependencia funcional entre asignatura y grupo:

- $G \rightarrow A$

Además, todo alumno en una asignatura pertenece a un único grupo, luego, para cada valor del par (AL, A), se tiene un único valor de G, de forma que existe otra dependencia funcional entre estos tres atributos:

- $AL, A \rightarrow G$

> *Cada profesor depende siempre y únicamente de un departamento (D). Así mismo, toda asignatura está ligada a un único departamento, el del profesor responsable de la misma.*

Por último, se sabe que un profesor pertenece a un único departamento, luego ($CP \rightarrow D$) y toda asignatura pertenece a un departamento, de forma que ($A \rightarrow D$):

- $CP \rightarrow D$
- $A \rightarrow D$

Resumiendo, la relación universal resultante del análisis del caso es R(AT, DEP), de:

AT = {*CP, NP, T, P, A, G, AL, D*}

DEP = {

(1) $CP \rightarrow NP$,

(2) $NP \rightarrow CP$,

(3) $A \rightarrow CP$,

(4) $G \rightarrow A$,

(5) $AL, A \rightarrow G$,

(6) $CP \rightarrow D$

(7) $A \rightarrow D$

(8) $CP \rightarrow\rightarrow T \mid P$

}

ubrimiento minimal

En este apartado, se comprueba que el conjunto de las dependencias halladas en partado anterior (DEP) es irredundante. Para ello se ha de comprobar que todas las ndencias son elementales, no existe ningún atributo extraño en ninguna de ellas y xisten dependencias redundantes.

endencias elementales

Lo primero es comprobar que todas las dependencias son elementales:

- En todas las dependencias se cumple que tienen un único atributo implicado
- Todas las dependencias son plenas, es decir, que ningún subconjunto del determinante puede implicar al implicado.
- Ninguna dependencia es trivial, es decir, ningún atributo del determinante aparece como implicado.

Por lo tanto, todas las dependencias son elementales.

Atributos extraños

A continuación se comprueba que no existen atributos extraños en las dependencias. Un atributo A es extraño si dada la dependencia $X \rightarrow Y$ de DEP, la dependencia $(X\text{-}A) \rightarrow Y$ se encuentra en DEP$^+$.

Para comprobar que la dependencia $(X\text{-}A) \rightarrow Y$ pertenece a DEP$^+$ se ha de cumplir que Y pertenece al cierre de $X\text{-}A$ ($(X\text{-}A)^+_{DEP}$).

En el conjunto de dependencias, DEP, en la única dependencia donde podrían existir atributos extraños es en $(AL, A \rightarrow G)$, ya que tiene más de un atributo como determinante.

Para ver si A es un atributo extraño, se comprueba que $AL \rightarrow G$ no se encuentra en DEP$^+$. Para ello se calcula el cierre de AL:

- $AL^+ = \{AL\}$, ya que el atributo por sí sólo no determina a ningún otro.

Como el atributo G no se encuentra en el cierre de AL, se puede concluir que A no es un atributo extraño. Para comprobar si AL es atributo extraño se halla el cierre de A:

- $A^+ = \{A, CP, D\}$, pero no determina a G

Por lo tanto, AL tampoco es un atributo extraño.

Dependencias redundantes

Por último se comprueba si existen dependencias redundantes en el conjunto DEP. Se dice que una dependencia dp es redundante si puede derivarse del conjunto DEP' = {DEP – dp}.

Por lo tanto, por cada dependencia de DEP, se determina si se puede derivar del resto (DEP'). No se resuelve completamente el ejercicio, debido a que el cálculo es repetitivo, se deja como ejercicio para el lector. Solamente se indican las dependencias redundantes y por qué lo son:

La dependencia $A \rightarrow D$ es una dependencia redundante, ya que se puede deducir a partir del conjunto de dependencias {DEP – $\{A \rightarrow D\}$}:

- $(A)^+_{DEP'} = \{A, CP, NP, D\}$
  - (3): $A \rightarrow CP$, luego $(A)^+_{DEP'} = \{A, CP\}$
  - (1): $CP \rightarrow NP$, luego $(A)^+_{DEP'} = \{A, CP, NP\}$
  - (6): $CP \rightarrow D$, luego $(A)^+_{DEP'} = \{A, CP, NP, D\}$

- Como *D* se determina a partir de DEP', la dependencia (7) es redundante.

A partir de todos estos cálculos se puede concluir que uno de los conjuntos mínimos de dependencias posibles es el siguiente:

DEP^m = {

$CP \rightarrow NP$,

$NP \rightarrow CP$,

$A \rightarrow CP$,

$G \rightarrow A$,

$AL, A \rightarrow G$,

$CP \rightarrow D$

$CP \rightarrow\rightarrow T \mid P$

}

Claves candidatas de la relación

El algoritmo de cálculo de si un descriptor (subconjunto de atributos de AT) es clave candidata se basa en el cálculo de su cierre transitivo.

En primer lugar se observa que existen dos atributos multivaluados, *T* y *P*, por lo que forman parte de cualquier clave candidata al no ser implicados funcionalmente por ningún otro atributo.

A continuación, se calcula el cierre transitivo de todos los atributos de la relación:

- $(CP)^+_{DEP} = \{CP, NP, D\}$
- $(NP)^+_{DEP} = \{CP, NP, D\}$, *CP* es equivalente a *NP*.
- $(A)^+_{DEP} = \{A, CP, NP, D\}$
- $(G)^+_{DEP} = \{G, A, CP, NP, D\}$
- $(AL)^+_{DEP} = \{AL\}$
- $(D)^+_{DEP} = \{D\}$

- $(T)^+_{DEP} = \{T\}$
- $(P)^+_{DEP} = \{P\}$

Hasta el momento, ningún atributo simple implica transitivamente a todos los atributos de la relación (AT), por lo que se realizan grupos de atributos que puedan determinar a todo el conjunto AT, eligiendo aquellos atributos con mayor cardinalidad en su cierre transitivo:

- $(T, P, G)^+ = \{T, P, G, A, CP, NP, D\}$, donde todavía no se recoge el atributo *AL*

Por lo tanto, una clave candidata es el conjunto formado por $\{T, P, G, AL\}$, ya que no existe ningún subconjunto de la misma que a su vez implique a todo AT. Por otro lado, como en la dependencia (5) de DEP ($AL, A \rightarrow G$), donde los atributos *AL* y *A* implican a *G*, el conjunto $\{T, P, A, AL\}$ también es clave candidata.

Resumiendo los cálculos anteriores:

- Claves candidatas: $\{T, P, G, AL\}$ y $\{T, P, A, AL\}$
- Atributos principales (forman parte de alguna clave candidata): $\{T, P, G, A, AL\}$
- Atributos no principales (AT – atributos principales): $\{CP, NP, D\}$

Forma Normal de la relación

- **1FN:** La relación se encuentra en 1FN al no poseer grupos repetitivos.
- **2FN:** La relación R no se encuentra en 2FN, ya que existen atributos no principales que no dependen plenamente de la clave. Por ejemplo: en la dependencia $\{CP \rightarrow NP\}$, el atributo *NP* no depende de ninguna clave principal (ni siquiera transitivamente).

Normalización por análisis

Se va a normalizar la relación R mediante el proceso de análisis, construyendo al mismo tiempo el árbol de descomposición. Se detalla paso a paso el algoritmo:

Paso 1: Recubrimiento minimal (calculado anteriormente)

DEP^m = {

$CP \rightarrow NP$,

$NP \rightarrow CP$,

$A \rightarrow CP$,

$G \rightarrow A$,

$AL, A \rightarrow G,$

$CP \rightarrow D$

$CP \rightarrow\rightarrow T \mid P$

}

Paso 2: Descomposición de R

El primer paso es identificar los atributos equivalentes. Los atributos *CP* y *NP* on equivalentes, ya que $CP \rightarrow NP$ y $NP \rightarrow CP$. Por lo tanto, se agrupan estas ependencias funcionales y se crea una relación R_1 cuyos atributos son los que articipan en ambas dependencias. Al mismo tiempo, se crea otra relación (R_1'), esta ez con carácter temporal, necesaria para continuar el proceso de descomposición. Los tributos de R_1' son los de R menos los atributos no principales de R_1, y las ependencias son las de $\{R - R_1\}$. Gráficamente se muestran descomposición en la gura 3.13.

R_1

AT = {CP, NP}
DEP = {CP → NP, NP → CP}

R_1'

AT = {CP, T, P, A, G, AL, D}
DEP = {A→CP, G→A, (AL, A)→G, CP→D, A→D, CP→→ T | P}

Figura 3.13. Descomposición de R

A continuación se realiza el cálculo de la forma normal en la que se encuentra la elación R_1, para ello, primero se hallan las claves candidatas:

Las claves de la relación son {*CP*} y {*NP*}

El cálculo de la forma normal en la que se encuentra la relación se detalla a ontinuación:

1FN: Sí, al tratarse de una relación

2FN y 3FN: Sí, ya que no existen atributos no principales.

FNBC: Sí, ya que todo determinante (o implicante) es clave candidata.

4FN y 5FN: Sí, ya que no existen dependencias multivaluadas ni en combinación en R_1.

A continuación se calculan las claves candidatas y forma normal en la que se encuentra la relación R_1'.

La clave de R_1' se calcula de forma similar a la clave de R. Como *T* y *P* no se encuentran implicados por ningún otro atributo, forman parte de toda clave candidata. El cierre transitivo de *G*, cuya cardinalidad sigue siendo la mayor, es $(G)^+_{DEP} = \{G, A, CP, D\}$, por lo que se repite la situación del cálculo en R, y por lo tanto sus claves candidatas son las mismas:

- Claves candidatas: {*T, P, G, AL*} y {*T, P, A, AL*}
- Atributos principales: {*T, P, G, A, AL*}
- Atributos no principales: {*CP, D*}

El cálculo de la forma normal en la que se encuentra la relación se detalla a continuación:

- 1FN: Sí, al tratarse de una relación
- 2FN: No, ya que aún existen atributos no principales que dependen parcialmente de la clave. Por ejemplo en la dependencia $A \rightarrow CP$.

De esta forma el primer nivel del árbol de descomposición se representa gráficamente en la figura 3.14.

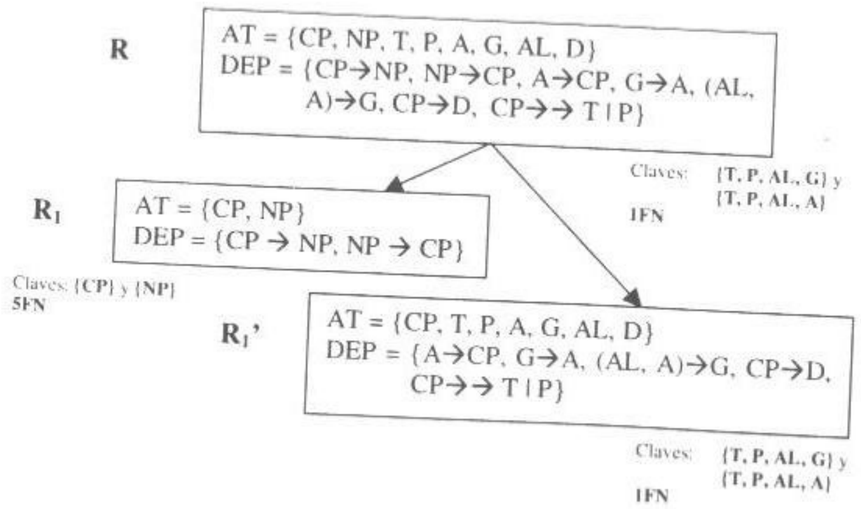


Figura 3.14. Primer nivel del árbol de descomposición

Paso 3: Descomposición de R_1'

Se elige otro grupo de dependencias funcionales, en R_1' no se realizan más grupos de dependencias, ya que no existen dos dependencias con el mismo implicante,

por lo que se elige, entonces, otra dependencia funcional cualquiera, por ejemplo *CP* → *D*, ya que *D* no participa en ninguna otra dependencia.de R_1'. Se descompone entonces en dos relaciones nuevas R_2 y R_2', con los atributos y dependencias que aparecen en la figura 3.15.

R_2
AT = {CP, D}
DEP = {CP → D}

R_2'
AT = {CP, T, P, A, G, AL}
DEP = {A→CP, G→A, (AL, A)→G, CP→→ T | P}

Figura 3.15. Descomposición de R_1

La clave de R_2 es *CP*, ya que determina a D, además la relación se encuentra en 5FN, ya que su atributo no principal (D) se encuentra determinado por toda la clave, su único determinante (CP) es clave y no existen nada más que dependencias funcionales.

La clave de R_2', al igual que ocurría con R_1', está formada por *T* y *P*. Además $(AL, G)^+$={*G, A, CP, AL*} y $(AL, A)^+$={*G, A, CP,* AL}, luego las claves son las mismas que en R y que en R_1'.

- Clave(R_2') = Clave(R_1') = Clave(R) = {*T, P, AL, G*} y {*T, P, AL, A*}
- R_2', al igual que R_1', también se encuentra en 1FN, ya que *CP* depende funcionalmente de *A*, que no es clave candidata, aunque forma parte de una de ellas.

El árbol de descomposición hallado hasta el momento aparece en la figura 3.16

Paso 4: Descomposición de R_2'

A continuación se realiza el tratamiento de las dependencias multivaluadas, ya que si se eligiera cualquiera de las otras dependencias, donde el implicado se encuentra involucrado en otras, se podría perder alguna.

En principio, se crea una relación con la dependencia multivaluada jerárquica

- *R_3 ({CP, T, P}, {CP →→ T | P}).*

Donde la clave está compuesta por todos los atributos de la relación. Pero esta relación no se encuentra en 4FN, ya no se encuentra determinada por toda la clave de la relación ({*CP, T, P*}). Se ha demostrado que toda dependencia multivaluada

jerárquica se puede descomponer en nuevas relaciones sin pérdidas de información. La descomposición que se realiza en este caso es la siguiente:

- *R_{31} ({CP, T}, {}).*
- *R_{32} ({CP, P}, {}).*

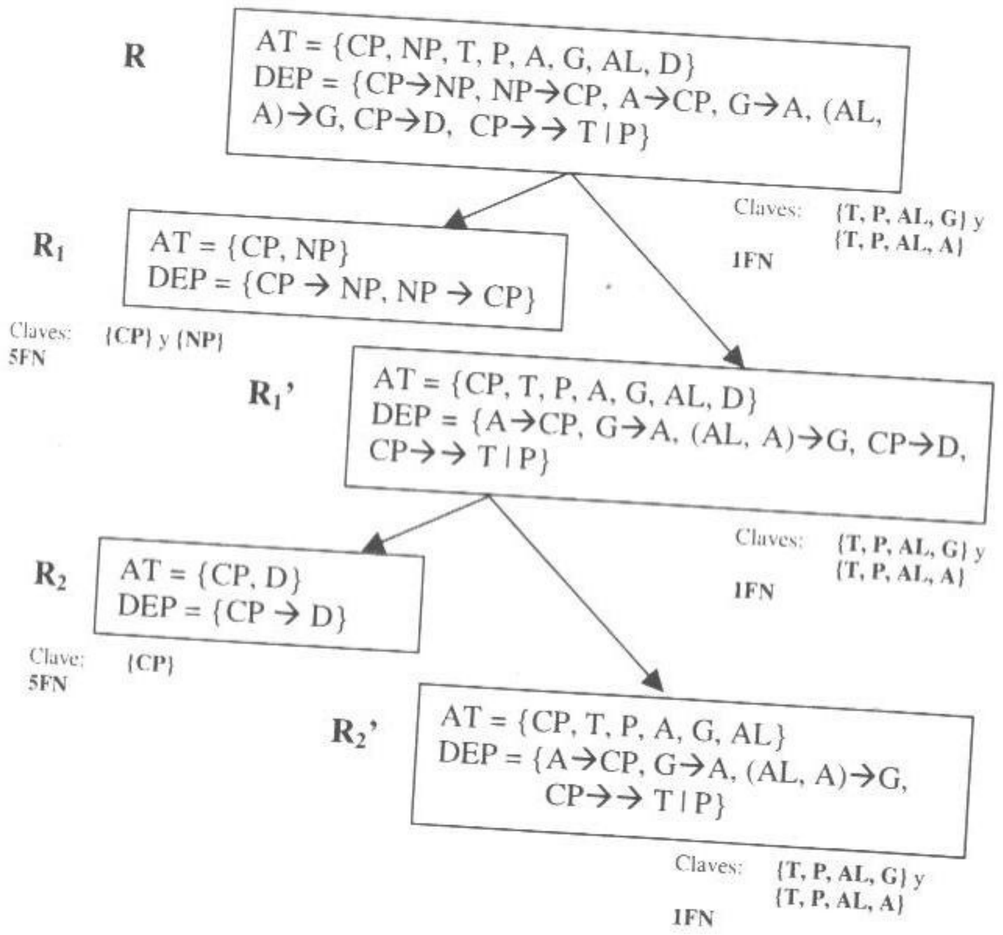


Figura 3.16. Árbol de descomposición parcial II

Esta vez las relaciones R_{31} y R_{32} sí se encuentran en 4FN (todos los atributos forman parte de la clave) y también en 5FN, ya que no existen dependencias en combinación.

Por otro lado, se crea la relación temporal R_3' para poder continuar con la descomposición. El árbol de descomposición parcial hallado hasta el momento es el que se muestra en la figura 3.18.

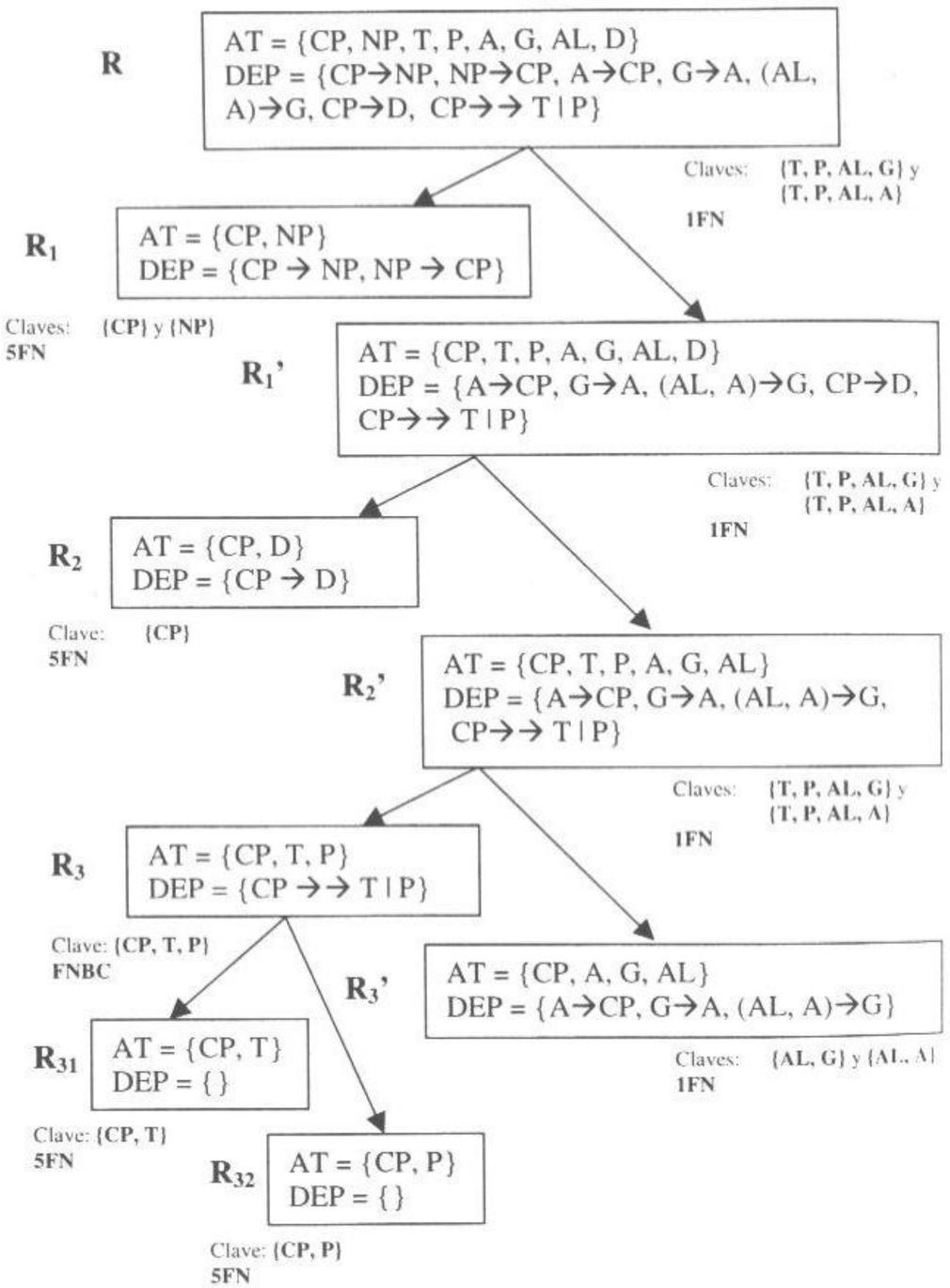


Figura 3.17. Árbol de descomposición parcial III

Paso 5: Descomposición de R_3'

De nuevo se continúa descomponiendo la relación temporal, en este caso se vuelve a la descomposición de las dependencias funcionales, ya que ya se han resuelto todas las multivaluadas que existen. En estos instantes se puede elegir la dependencia $A \rightarrow CP$, ya que CP no participa en ninguna otra dependencia de R_3.

Las relaciones que se crean son las que se muestran en la figura 3.18.

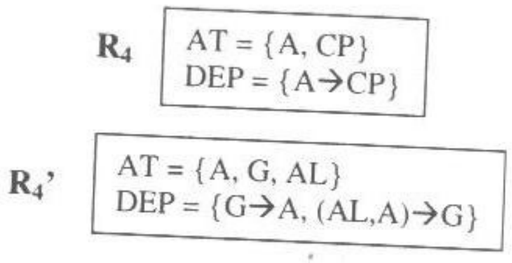


Figura 3.18 Descomposición de R_3

La relación R_4 se encuentra en 5FN, y su clave es el atributo A, mientras que la relación R_4' tiene como claves $\{AL, G\}$ y $\{AL, A\}$, que se encuentra en 3FN, pero no en FNBC, ya que no todo determinante es clave candidata (en la dependencia $G \rightarrow A$).

Paso 6: Descomposición de R_4'

Si se intenta descomponer la relación R_4' en dos relaciones se pueden perder dependencias funcionales, ya que, si se eligiera la primera dependencia ($G \rightarrow A$), el implicado, A, participa en la segunda dependencia, y en el caso de que se eligiera la segunda, G participa en la primera, luego, la relación no se puede descomponer sin pérdidas de dependencias.

Se supone (aunque no se detalla en el enunciado) que es requisito indispensable el mantenimiento de todas las dependencias de la relación. Por lo tanto, el árbol de descomposición final se muestra en la figura 3.19

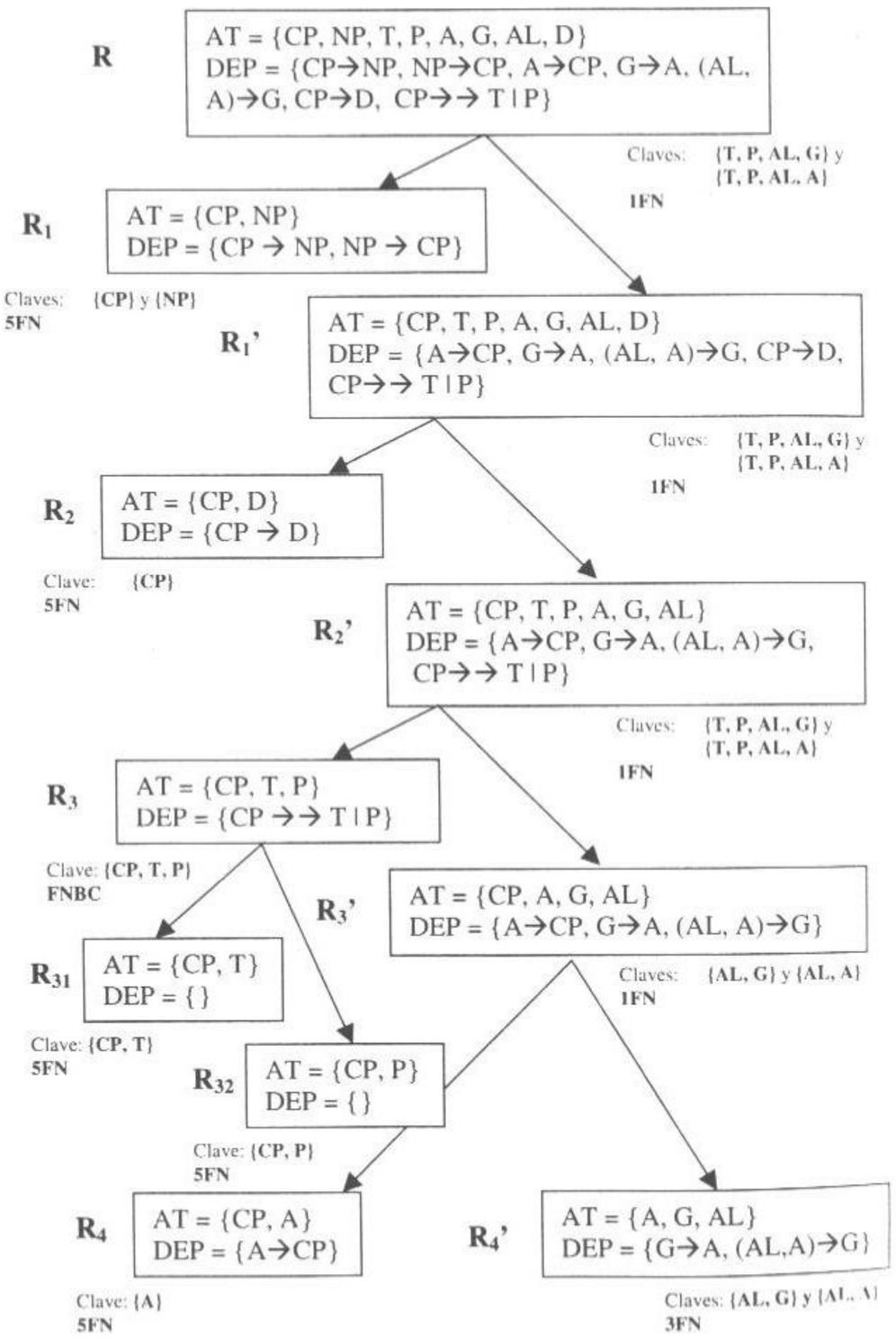


Figura 3.19 Árbol de descomposición total

PROBLEMA 5

Enunciado

Se desea diseñar una base de datos para una empresa de ventas que tiene representantes (R) en las distintas áreas (A) donde vende sus productos (P). Se supone que:

a) Los representantes tienen un código (CR) y un conjunto de atributos (AR). Análogamente las áreas tienen CA y AA y los productos CP y AP.

b) En cada A hay varios R y cada R trabaja en varias A.

c) En todas las A se venden todos los P.

d) Un R puede vender varios P y cada P se vende por varios R

e) Nunca dos R venden el mismo P en la misma A.

f) Todo R vende el mismo conjunto de P en cada A donde trabaja.

Se pide:

1) Las dependencias que se deducen de cada uno de los supuestos del enunciado. Si de algún supuesto se deduce la no existencia de alguna dependencia indíquelo también.

2) Un conjunto mínimo de dependencias válido (recubrimiento minimal o irredundante).

3) Estructura relacional que considere más adecuada, analizando el nivel de normalización de cada una de las relaciones.

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

En este apartado se analiza paso a paso el enunciado para hallar los atributos de la relación universal de la empresa de ventas y las dependencias funcionales, multivaluadas y de combinación que existen entre los atributos de la misma. Para realizar un análisis detallado, al igual que se hizo en el problema 4, se introducen referencias al enunciado.

Los representantes tienen un código (CR) y un conjunto de atributos (AR). Análogamente las áreas tienen CA y AA y los productos CP y AP.

Esta frase del enunciado indica que la entidad representante se identifica mediante un código (*CR*), y un conjunto de atributos (*AR*). Se supone que para un único valor de *CR* se tiene un único valor de *AR*, luego *CR* determina funcionalmente a *AR*.

- $CR \rightarrow AR$

Por otro lado, las áreas y los productos, al igual que los representantes, se identifican por su código y atributos:

- $CA \rightarrow AA$
- $CP \rightarrow AP$

En cada A hay varios R y cada R trabaja en varias A.

En la frase anterior se indica que en cada área existen varios representantes y cada representante trabaja en varias áreas, luego no existe ninguna dependencia funcional entre estos dos atributos (*CA -/-> CR* y *CR -/-> CA*), o lo que es lo mismo:

- $¢ \rightarrow\rightarrow CA \mid CR$

En todas las A se venden todos los P.

Esto indica que en cada área se venden todos los productos, y que los productos se pueden vender en cada área, por lo tanto se trata de una relación similar a la anterior, donde no existe ninguna relación entre los atributos (*CA -/-> CP* y *CP -/-> CA*), lo que es igual a que:

- $¢ \rightarrow\rightarrow CA \mid CP$

Un R puede vender varios P y cada P se vende por varios R.

Como en las dos frases anteriores, entre estos dos atributos no existe ningún tipo de relación (*CP -/-> CR* y *CR -/-> CP*), es decir:

- $¢ \rightarrow\rightarrow CR \mid CP$

Nunca dos R venden el mismo P en la misma A.

Por otro lado, se indica que un representante siempre vende el mismo producto en la misma área. Es decir, siempre se sabe qué representante suministró un producto dado en un área concreta. Esto indica que existe una dependencia funcional de la forma:

- $CP, CA \rightarrow CR$

Todo R vende el mismo conjunto de P en cada A donde trabaja.

Por último se indica que todo representante vende siempre los mismos productos en todas las áreas. Por lo tanto, el representante vende un conjunto bien determinado de productos en un conjunto de áreas bien determinadas, sin que exista ninguna relación entre el producto y el área, ya que suministra todos los productos en todas las áreas. De esta forma, el representante multidetermina tanto a los productos como a las áreas (multidependencia jerárquica):

- $CR \rightarrow\rightarrow CA \mid CP$

Resumiendo, la relación universal resultante del análisis del enunciado, obviando las "no-dependencias" es R(AT,DEP), donde:

AT = {*CR, AR, CA, AA, CP, AP*}

DEP = {

(1) $CR \rightarrow AR$,

(2) $CA \rightarrow AA$,

(3) $CP \rightarrow AP$,

(4) $CP, CA \rightarrow CR$,

(5) $CR \rightarrow\rightarrow CP \mid CA$,

}

RECUBRIMIENTO MINIMAL

En este apartado, se comprueba que el conjunto de las dependencias halladas en el apartado anterior (DEP) es irredundante. Para ello se tiene que cumplir que todas las dependencias sean elementales, no exista ningún atributo extraño y no existan dependencias redundantes.

Dependencias elementales

Lo primero es comprobar que todas las dependencias son elementales:

- En todas las dependencias se cumple que tienen un único atributo implicado.

- Todas las dependencias son plenas, es decir, que ningún subconjunto del determinante puede implicar al implicado.
- Ninguna dependencia es trivial, es decir, ningún atributo del determinante aparece como implicado.

Por lo tanto, todas las dependencias son elementales.

Atributos extraños

A continuación se comprueba que no existen atributos extraños en las dependencias.

Del conjunto de dependencias DEP, en la única dependencia donde podrían existir atributos extraños es en (CP, $CA \rightarrow CR$), ya que el determinante está formado por más de un atributo.

Para ver si CA es un atributo extraño, se comprueba s $CP \rightarrow CR$ se encuentra en DEP^+. Para ello se calcula el cierre de CP:

- $CP^+ = \{CP, AP\}$, pero no determina a CR, luego CA no es un atributo extraño.

Para comprobar si CP es atributo extraño se calcula el cierre de CA:

- $CA^+ = \{CA, AA\}$, pero no determina a CR, luego CP no es un atributo extraño.

Dependencias redundantes

Por último se comprueba si existen dependencias redundantes en el conjunto DEP. Se dice que una dependencia dp es redundante si puede derivarse del conjunto $DEP' = \{DEP - dp\}$.

Por lo tanto, por cada dependencia de DEP, se calcula si se puede derivar del resto (DEP'). No existe ninguna dependencia redundante en DEP. En este problema no se detalla el cálculo de la existencia de alguna dependencia redundante, ya que la forma de calcularlo se ha explicado en problemas anteriores, se deja como ejercicio al lector.

Concluyendo, el conjunto de dependencias DEP es un conjunto de dependencias mínimo:

$DEP^m = DEP$

NORMALIZACIÓN

Se realiza la normalización de la relación R mediante el algoritmo de síntesis, indicando la forma normal en la que se encuentra cada relación en cada momento.

Paso 1: Recubrimiento minimal (calculado anteriormente)

$DEP^m = \{$

(1) $CR \rightarrow AR$,

(2) $CA \rightarrow AA$,

(3) $CP \rightarrow AP$,

(4) $CP, CA \rightarrow CR$,

(5) $CR \rightarrow\rightarrow CP \mid CA$,

$\}$

Paso 2: Claves de la relación y FN en la que se encuentra

En primer lugar se han de calcular las claves candidatas de la relación R, para ello es necesario calcular el cierre transitivo de los predicados:

- $CR^+ = \{CR, AR\}$
- $CA^+ = \{CA, AA\}$
- $CP^+ = \{CP, AP\}$

Se sabe que los atributos multivaluados CP y CA no se encuentran implicados funcionalmente por ningún otro atributo, luego forman parte de todas las claves candidatas. El cierre transitivo del conjunto $\{CP, CA\}$ es:

- $\{CP, CA\}^+ = \{CP, AP, CA, AA, CR, AR\}$

De esta forma, se tiene que el conjunto $\{CP, CA\}$ implica a todos los atributos de R, luego será la clave candidata. Resumiendo:

- Clave candidata: $\{CP, CA\}$
- Atributos principales: $\{CP, CA\}$
- Atributos no principales: $\{AP, AA, CR, AR\}$

Por otro lado, la forma normal en la que se encuentra la relación es en 1FN, ya que en 2FN no se encuentra al existir atributos no principales determinados funcionalmente por parte de la clave. Por ejemplo en la dependencia $CP \rightarrow AP$, el atributo no principal AP se encuentra determinado por CP, no por la clave completa.

Paso 3: Síntesis de DEP

El primer paso es agrupar todas las dependencias funcionales que tienen el mismo implicante o atributos equivalentes. Solamente se crea un grupo, la dependencia multivaluada $CR \rightarrow\rightarrow CP|CA$ y la dependencia $(CP, CA) \rightarrow CR$, ya que CR es el implicado de una y el implicante de la otra.

Posteriormente se crean relaciones con los grupos de dependencias hallados. Las relaciones que se forman son las siguientes:

- $R_1(\{\underline{CR}, AR\}, \{CR \rightarrow AR\})$
- $R_2(\{\underline{CA}, AA\}, \{CA \rightarrow AA\})$
- $R_3(\{\underline{CR}, AP\}, \{CR \rightarrow AP\})$
- $R_4(\{\underline{CP, CA}, CR\}, \{(CP,CA) \rightarrow CR, CR \rightarrow\rightarrow CP|CA\})$

La relación R_1 se encuentra en 5FN, ya que todo determinante es clave candidata (FNBC) y además no tiene dependencias multivaluadas y en combinación. Lo mismo ocurre con las relaciones R_2 y R_3, que se encuentran también en 5FN.

Sin embargo, la relación R_4 no se encuentra nada más que en FNBC, ya que cumple que todo determinante de las dependencias funcionales es clave candidata (es este caso $\{CP, CA\}$), pero al tener la relación multivaluada, no cumple la 4FN.

Para poder continuar con la descomposición de la relación R_4, es necesario utilizar el algoritmo de análisis, ya que el algoritmo de síntesis sólo habla de dependencias funcionales.

La dependencia multivaluada jerárquica $CR \rightarrow\rightarrow CP|CA$ se puede descomponer en dos relaciones sin pérdidas de información:

- $R_{4a}(\{\underline{CR, CA}\}, \{\})$
- $R_{4b}(\{\underline{CR, CP}\}, \{\})$

Ambas relaciones se encuentran en 5FN, ya que no existen dependencias entre sus atributos.

Por otro lado, la dependencia de R_4 que falta por analizar $(CP,CA \rightarrow CR)$ formaría una nueva relación:

- $R_{4c}(\{\underline{CP, CA, CR}\}, \{CP,CA \rightarrow CR\})$

En este caso, esta relación, cuya clave candidata es $\{CP,CA\}$, se encuentra en 5FN, ya que se encuentra en 3FN al haber solamente un atributo no principal, en FNBC ya que todo determinante es clave candidata, y en 4FN y 5FN al no existir dependencias funcionales ni en combinación.

PROBLEMA 6

Enunciado

Un departamento universitario desea diseñar una base de datos para la gestión de los cursos que imparte durante un cuatrimestre. En la base de datos quiere almacenar los profesores (P), los estudiantes (E), la nota (N) con la que se califica a un alumno en cada asignatura (AS), así como los días de la semana/hora (H) en las que se imparte una asignatura y el aula (AU) (se supone que ni el día/hora ni el aula en los que se imparte una asignatura varían de una semana a otra). Se desea almacenar también el teléfono (TL) y el despacho (D) de cada profesor (se supone que no existen teléfonos compartidos por dos profesores y que en cada despacho sólo hay un profesor y un teléfono). Se sabe asimismo que un profesor imparte clases en varios grupos (G) y en todos ellos utiliza los mismos textos (T).

Además de los anteriores se dan los siguientes supuestos semánticos:

a) En un momento dado tanto un estudiante como un profesor sólo pueden estar en un aula.

b) En un momento dado en un aula sólo se puede impartir una asignatura.

c) En cada despacho hay un solo teléfono.

d) Un estudiante no puede asistir a las clases de dos asignaturas en una misma hora.

e) Todas las asignaturas están divididas en los mismos grupos, utilizándose en todos los grupos de la misma asignatura los mismos textos.

Se pide:

1) Determinar las dependencias que existen entre los atributos.
2) Recubrimiento minimal, determinando si existe alguna dependencia redundante o algún atributo ajeno.
3) Forma normal en la que se encuentra la relación.
4) Esquema relacional normalizado con las observaciones que estime pertinentes, sin tener en cuenta las dependencias multivaluadas o en combinación.
5) ¿Debería añadir al esquema anterior la relación Rx(As, G), a fin de tener en cuenta alguna posible dependencia multivaluada?

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

Se analizan detalladamente los supuestos semánticos del enunciado con el objetivo de determinar los atributos de la base de datos y las dependencias funcionales, multivaluadas y de combinación que existen entre ellos. Para realizar un análisis detallado, al igual que se hizo en problemas anteriores, se introducen referencias al enunciado.

En el enunciado se nombran los atributos profesor (*P*), estudiante (*E*), nota (*N*), asignatura (*As*), una determinada hora (*H*), aula (*Au*), teléfono (*Tl*) y despacho (*D*) del profesor, grupo de alumnos (*G*) y textos (*T*).

En primer lugar se analizan los supuestos semánticos que se encuentran enumerados en el enunciado, y a continuación se comprueba si existe algún supuesto semántico más "oculto" en el enunciado:

> *En un momento dado tanto un estudiante como un profesor sólo pueden estar en un aula.*

Es decir, conociendo la hora y el estudiante, se sabe en cada momento en qué aula se encuentra; por lo tanto, los atributos *hora* y *estudiante* implican funcionalmente al atributo *aula* ($H, E \rightarrow Au$). De igual forma, conociendo la hora y el profesor, se sabe en qué aula está el profesor ($H, P \rightarrow Au$). Por lo tanto, las relaciones funcionales encontradas hasta el momento son las siguientes:

- $H, E \rightarrow Au$
- $H, P \rightarrow Au$

> *En un momento dado en un aula sólo se puede impartir una asignatura.*

Por otro lado, conociendo la hora exacta y el aula, se sabe en todo momento qué asignatura se imparte en el aula ($H, Au \rightarrow As$).

- $H, As \rightarrow Au$

> *En cada despacho hay un solo teléfono.*

Esto implica que el atributo *despacho* implica el atributo *teléfono* ($D \rightarrow Tl$).

- $D \rightarrow Tl$

> *Un estudiante no puede asistir a las clases de dos asignaturas en una misma hora.*

Esto indica que un estudiante a una hora determinada sólo puede asistir a una asignatura. Es decir, conociendo el valor del atributo *estudiante* y del atributo *hora*, existe un único valor posible para el atributo *asignatura* ($H, E \rightarrow As$).

- $H, E \rightarrow As$

> *Todas las asignaturas están divididas en los mismos grupos, utilizándose en todos los grupos de la misma asignatura los mismos textos.*

Por último, se tiene que todas las asignaturas de la base de datos están divididas en los mismos grupos, es decir, que una asignatura no implica funcionalmente un grupo, o, lo que es lo mismo, $\not\in \rightarrow\!\!\rightarrow As \mid G$. Por otro lado, se tiene que en todos los grupos de la misma asignatura se explican los mismos textos, es decir, que una asignatura multidetermina jerárquicamente el grupo y los textos que se dan en ella ($As \rightarrow\!\!\rightarrow G \mid T$).

- $\not\in \rightarrow\!\!\rightarrow As \mid G$
- $As \rightarrow\!\!\rightarrow G \mid T$

A continuación se comprueba si existen más supuestos semánticos en el resto del enunciado, referenciándolo cuando se cree oportuno.

> *En la base de datos quiere almacenar los profesores (P), los estudiantes (E), la nota (N) con la que se califica a un alumno en cada asignatura (AS).*

Por lo tanto, a un alumno, en una asignatura concreta, se le califica con una única nota. Es decir, para una pareja de valores (*estudiante, asignatura*), existe un único valor posible para el atributo *nota*, luego ($E, As \rightarrow N$).

- $E, As \rightarrow N$

> *Se desea almacenar también el teléfono (TL) y el despacho (D) de cada profesor (se supone que no existen teléfonos compartidos por dos profesores y que en cada despacho sólo hay un profesor y un teléfono).*

Al no existir teléfonos compartidos por dos profesores, esto indica que el *profesor* determina funcionalmente al *teléfono* que le pertenece ($P \rightarrow Tl$) y que conociendo un número de teléfono se sabe a qué profesor pertenece ($Tl \rightarrow P$). En cada despacho sólo existe un teléfono ($D \rightarrow Tl$), y cada teléfono sólo pertenece a un despacho ($Tl \rightarrow D$). También se sabe que en cada despacho existe un único profesor ($D \rightarrow P$), y que un profesor tiene un único despacho ($P \rightarrow D$). Resumiendo:

- $P \rightarrow Tl$
- $Tl \rightarrow P$
- $D \rightarrow Tl$ (dependencia repetida en un supuesto anterior)
- $Tl \rightarrow D$
- $P \rightarrow D$
- $D \rightarrow P$

Se sabe asimismo que un profesor imparte clases en varios grupos (G) y en todos ellos utiliza los mismos textos (T).

Al igual que ocurría con las asignaturas, un profesor imparte clases en varios grupos (conjunto bien determinado), utilizando en todos ellos los mismo textos, luego existe una dependencia multivaluada entre el profesor, los grupos y los textos ($P \rightarrow\rightarrow G \mid T$).

- $P \rightarrow\rightarrow G \mid T$

Para terminar con este apartado, se puede concluir que la relación universal con todos los atributos y dependencias de la base de datos del departamento sería la siguiente (nota: se han eliminado las dependencias cuyo implicante es el conjunto vacío, ya que indica la "no-dependencia" entre atributos):

R (AT, DEP), donde:

AT = {*H, E, Au, P, Tl, D, As, N, G, T*}

DEP = {

(1) $H, E \rightarrow Au$

(2) $H, P \rightarrow Au$

(3) $H, Au \rightarrow As$

(4) $H, E \rightarrow As$

(5) $E, As \rightarrow N$

(6) $As \rightarrow\rightarrow G \mid T$

(7) $P \rightarrow Tl$

(8) $Tl \rightarrow P$

(9) $D \rightarrow Tl$

(10) $Tl \rightarrow D$

(11) $P \rightarrow D$

(12) $D \rightarrow P$

(13) $P \rightarrow\rightarrow G \setminus T$

}

RECUBRIMIENTO MINIMAL

En este apartado, se comprueba que el conjunto de las dependencias halladas en el apartado anterior (DEP) es irredundante. Por lo tanto, se va a determinar si todas las dependencias son elementales, si existe ningún atributo extraño y si existen dependencias redundantes, para eliminarlas del conjunto de dependencias mínimo.

Dependencias elementales

- En todas las dependencias se cumple que tienen un único atributo implicado.
- Todas las dependencias son plenas, es decir, que ningún subconjunto del determinante puede implicar al implicado.
- Ninguna dependencia es trivial, es decir, ningún atributo del determinante aparece como implicado.

Por lo tanto, se puede concluir que todas las dependencias son elementales.

Atributos extraños

Se comprueba si existen atributos extraños en alguna de las dependencias del conjunto DEP. Sólo pueden existir atributos extraños en las dependencias funcionales cuyo determinante está compuesto por más de un atributo, luego, se ha de comprobar las cinco primeras dependencias del conjunto DEP (numeradas como (1)...(5)).

Para ver si en la dependencia funcional *H, E → Au* si el atributo *E* es un atributo traño, se comprueba si *H → Au* se encuentra en DEP⁺. Para ello se calcula el cierre *H*:

- $H^+ = \{H\}$, pero no determina a *Au*, luego *E* no es un atributo extraño.

Para comprobar si *H* es atributo extraño se calcula el cierre de *E*:

- $E^+ = \{E\}$, pero no determina a *Au*, luego *E* no es un atributo extraño.

Lo mismo ocurre con el resto de las dependencias de DEP, donde el cierre de los ributos implicantes tiene cardinalidad uno, ya que sólo se determinan a sí mismos ediante los axiomas de Armstrong). Luego, se puede concluir que no existe ningún ributo extraño en ninguna de las dependencias de DEP.

ependencias redundantes

Por último se comprueba si existen dependencias redundantes en el conjunto EP. Se dice que una dependencia *dp* es redundante si puede derivarse del conjunto EP' = {DEP – *dp*}.

En la solución de este problema no se detallan todos los cálculos realizados, se eja como ejercicio al lector. Se detalla únicamente el proceso de determinación de las ependencias redundantes:

La dependencia (9): *D → Tl* es una dependencia redundante, ya que se puede educir a partir del conjunto de dependencias DEP' = {DEP – {*D→Tl*}}. No se realiza do el cálculo del cierre, solamente hasta el punto que interesa:

$(D)^+_{DEP'} = \{D, P, Tl, ...\}$

- (12): *D → P*, luego $(D)^+_{DEP'} = \{D, P\}$
- (7): *P → Tl*, luego $(D)^+_{DEP'} = \{D, P, Tl\}$
- ...

Como *Tl* se determina a partir de DEP', la dependencia (9) es redundante.

De igual forma, la dependencia (10) *Tl → D* es redundante, ya que se puede educir a partir del resto (DEP' = {DEP – {*D→Tl; Tl→D*}}):

$(Tl)^+_{DEP'} = \{Tl, P, D ...\}$

- (8): *Tl → P*, luego $(Tl)^+_{DEP'} = \{Tl, P\}$
- (11): *P → D*, luego $(Tl)^+_{DEP'} = \{Tl, P, D\}$
- ...

Como *Tl* implica a *D* según las dependencias del conjunto DEP', la dependencia (10): *Tl→D* es redundante.

Lo mismo ocurre con la dependencia (4) *H, E → As*:

- $(H, E)^+_{DEP'} = \{H, E, Au, As ...\}$
  - (1): *H, E → Au*, luego $(H,E)^+_{DEP'} = \{H, E, Au\}$
  - (3): *H, Au → As*, luego $(H,E)^+_{DEP'} = \{H, E, Au, As\}$
  - ...

Como la dependencia (4) *H,E→*As se puede deducir a partir de DEP' = {DEP – {*D→Tl; Tl→D; H,E→As*}}, es redundante.

Nota: este conjunto mínimo de dependencias no es el único, ya que se podrían haber tachado de dependencias redundantes otras parejas entre el profesor, despacho y teléfono, ya que se trata de atributos equivalentes.

Para la solución detallada en este libro, el conjunto mínimo de dependencias hallado es el que se presenta a continuación:

$DEP^m = \{$

H, E → Au

H, P → Au

H, Au → As

E, As → N

P → Tl

Tl → P

P → D

D → P

As →→ G | T

P →→ G \ T

}

FORMA NORMAL DE LA RELACIÓN

Para poder calcular en qué nivel de normalización se encuentra la relación R(AT, DEP^m) se necesita saber cuáles son sus atributos principales y no principales, y por lo tanto se necesita realizar el cálculo de las claves candidatas de la relación.

El cálculo de las claves de una relación se basa en el cierre transitivo de sus descriptores, por lo tanto, en primer lugar, se realiza el cálculo de los cierres transitivos de los atributos de R de forma individual:

- $H^+ = \{H\}$
- $E^+ = \{E\}$
- $Au^+ = \{Au\}$
- $P^+ = \{P, D, Tl\}$
- $Tl^+ = \{P, D, Tl\}$
- $D^+ = \{P, D, Tl\}$
- $As^+ = \{As\}$
- $N^+ = \{N\}$
- $G^+ = \{G\}$
- $T^+ = \{T\}$

Del cálculo anterior se puede deducir que los atributos *P, Tl* y *D* son equivalentes

Por otro lado, los atributos no implicados por ningún otro han de formar parte de todas las claves de la relación. Se sabe que los atributos implicantes son *H, E, P, Au* y *As*, mientras que los atributos implicados son *Au, As* y *N*. Además, también se sabe que los atributos *G* y *T* no forman parte de ninguna dependencia funcional, por lo que forman parte de toda clave.

Como $\{H, P, G, T\}^+ = \{H, P, D, Tl, Au, As, G, T\}$, aún faltan los atributos *E* y *N* para formar todo el conjunto AT. Pero, según la dependencia *E, As* $\rightarrow N$, luego, si se incluye el atributo *E* al conjunto, se determina a todo AT. Por lo tanto se trata de una clave candidata de R:

- $\{H, E, P, G, T\}^+ = \{H, P, D, Tl, Au, As, G, T, E, N\} = \text{AT}$

Debido a que los atributos *P, Tl* y *D* son equivalentes, en la clave candidata anterior se puede sustituir, sin temor a la equivocación, el atributo *P* por cualquiera de sus atributos equivalente *Tl* y *D*. De esta forma serán tres las claves candidatas para la relación R, $\{H, E, P, G, T\}$, $\{H, E, D, G, T\}$ y $\{H, E, Tl, G, T\}$.

Resumiendo:

- Claves candidatas: {H, E, P, G, T}, {H, E, D, G, T} y {H, E, Tl, G, T}
- Atributos principales: *{H, E, P, D, Tl, G, T}*
- Atributos no principales: *{ As, Au, N}*

Por otro lado, la relación no se encuentra en 2FN, ya que existen atributos no principales que dependen de parte de alguna clave. Por ejemplo, el atributo *N* depende del conjunto *{E, As}*, de ninguna clave candidata. Por lo tanto, la relación se encuentra en 1FN.

NORMALIZACIÓN

Dado que en el enunciado del problema nos indica que no se han de tener en cuenta las dependencias multivaluadas ni las de combinación, se puede utilizar el algoritmo de síntesis para realizar la normalización de R.

Por lo tanto, se aplica el método de síntesis al conjunto de dependencias DEP'', donde:

DEP'' = {

(a) *H, E* $\rightarrow Au$

(b) *H, P* $\rightarrow Au$

(c) *H, Au* $\rightarrow As$

(d) *E, As* $\rightarrow N$

(e) *P* $\rightarrow Tl$

(f) *Tl* $\rightarrow P$

(g) *P* $\rightarrow D$

(h) *D* $\rightarrow P$

}

El primer paso es agrupar todas las dependencias funcionales que tienen el mismo implicante y/o atributos equivalentes. Por lo tanto, las dependencias (e), (f), (g) y (h) se agrupan, ya que sus implicantes son equivalentes, y se crea una relación con todas ellas:

- R1({*P*, *D*, *Tl*}, {*P* →*Tl*, *Tl* →*P*, *P* →*D*, *D* →*P*})

Esta es la única agrupación de dependencias que se puede realizar para este problema, ya que no existen implicados iguales ni equivalentes. Por lo tanto, se crea una relación más por cada una de las dependencias funcionales que existen:

- R2({*H, E, Au*}, {*H, E* →*Au*})
- R3({*H, P, Au*}, {*H, P* →*Au*})
- R4({*H, Au, As*}, {*H, Au* →*As*})
- R5({*E, As, N*}, {*E, As* →*N*})

Por último, se crea una nueva relación, con los atributos de la clave, donde se podrían incluir, si quisiéramos, las dependencias multivaluadas que existían en DEP[m]:

- R6({*H, E, P, G, T*},{*As* →→ *G*|*T* ; *P* →→ *G*|*T*})

Las cinco primeras relaciones (R1, ...,R5) se encuentran en 5FN, ya que todo determinante es clave de la relación y existe un único implicado, mientras que la relación R6 se encuentra en FNBC, debido a las dependencias multivaluadas jeráquicas.

Se ha demostrado que toda dependencia multivaluada jerárquica se puede descomponer, sin pérdida de información ni de dependencias [De Miguel et al., 1999]. De esta forma, la relación R6 se puede descomponer mediante el método de análisis en cuatro nuevas relaciones:

- R6a({*As, G*}, {})
- R6b({*As, T*}, {})
- R6c({*P, G*}, {})
- R6d({*P, T*}, {})

De esta forma todas las relaciones se encuentran en 5FN, incluyendo las nuevas halladas a partir de R6.

Respondiendo a la última cuestión del enunciado, no es necesario incluir la relación Rx(*As, G*) en el esquema de relación anterior, ya que ¢ →→*As*|*G*, y por lo tanto, no existe ningún tipo de relación entre los atributos *As* y *G*.

PROBLEMA 7

Enunciado

En un centro de experimentación farmacéutica sobre animales se desea comprobar la eficacia de determinados medicamentos (**M**) así como de los tratamientos (**T**) que se aplican a animales enfermos. En su fase inicial el centro decide imponer normas muy restrictivas en su funcionamiento. Se desea diseñar una base de datos aplicando la teoría de la normalización, para lo que nos dan las siguientes especificaciones:

a) Los animales se identifican indistintamente, según las aplicaciones, por un código alfabético (**C**) o por un número de orden (**N**).

b) Todos los veterinarios que trabajan en el centro tienen apellidos (**A**) distintos y, además, se les identifica también por su DNI (**D**).

c) Un tratamiento sólo se le puede aplicar a un animal (se comprueba mediante el número de animal), y nunca un tratamiento lo pueden aplicar dos veterinarios con distintos DNI.

d) Un veterinario (sus apellidos) no esta nunca asociado a más de un medicamento por cada animal.

e) Un veterinario sólo le puede aplicar un tratamiento a cada animal.

f) En un tratamiento, a cada animal sólo se le puede aplicar un medicamento.

Se pide:

1) Determinar las dependencias funcionales.

2) Hallar un recubrimiento minimal válido.

3) Determinar la forma normal en la que se encuentra la relación, explicando el proceso que le ha permitido llegar a la correspondiente conclusión.

4) Aplicar el proceso de análisis (árbol de descomposición) para obtener un esquema relacional en FNBC realizando las observaciones que se estimen pertinentes.

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

En este problema, al igual que se hizo con problemas anteriores, se analizan detalladamente los supuestos semánticos del enunciado con el objetivo de determinar los atributos y las dependencias que existen entre los mismos. Observar que esta vez en el enunciado sólo piden explícitamente las dependencias funcionales.

Los atributos que se desean guardar en la base de datos son: los medicamentos (*M*) y los tratamientos (*T*); el código alfabético (*C*) y el número de orden (*N*) de un animal; DNI (*D*) y apellidos (*A*) de los veterinarios.

A continuación se hallan las dependencias funcionales que existen entre estos atributos:

> *Los animales se identifican indistintamente, según las aplicaciones, por un código alfabético (**C**) o por un número de orden (**N**).*

Un animal se puede identificar por su código o por su número de orden. Esto implica que los dos atributos podrían ser clave principal de una relación donde se guardan todos la información de los animales, y por lo tanto, cualquiera de los dos determina funcionalmente al resto de sus atributos ($C \rightarrow N$) y viceversa ($N \rightarrow C$):

- $C \rightarrow N$
- $N \rightarrow C$

> *Todos los veterinarios que trabajan en el centro tienen apellidos (**A**) distintos y, además, se les identifica también por su DNI (**D**).*

Por otro lado, se desea guardar tanto el DNI como el apellido de los veterinarios del centro, donde se sabe que el DNI del veterinario implica funcionalmente el valor del apellido del mismo ($D \rightarrow A$). Pero además se sabe que no existen en el centro dos veterinarios con el mismo apellido, por lo que el apellido también puede actuar como clave primaria, por lo tanto también determina al resto de los atributos ($A \rightarrow D$).

- $A \rightarrow D$
- $D \rightarrow A$

> *Un tratamiento sólo se le puede aplicar a un animal (se comprueba mediante el número de animal), y nunca un tratamiento lo pueden aplicar dos veterinarios con distintos DNI.*

Esta frase del enunciado indica que un tratamiento sólo se le puede aplicar a un animal, por lo que, conociendo el valor del atributo *tratamiento*, existe un único valor para el atributo *número de orden* del animal ($T \rightarrow N$). Además un tratamiento sólo puede ser indicado por un veterinario ($T \rightarrow D$).

- $T \rightarrow N$
- $T \rightarrow D$

> *Un veterinario (sus apellidos) no esta nunca asociado a más de un medicamento por cada animal.*

Un veterinario sólo le puede recetar un medicamento a un animal concreto, de forma que conociendo el apellido (*A*) del veterinario y el número de orden del animal (*N*), se conoce el nombre del medicamento (*M*) que le recetó a ese animal (en el caso de que exista):

- $A, N \rightarrow M$

> *Un veterinario sólo le puede aplicar un tratamiento a cada animal.*

Además, existe una relación similar entre el veterinario, el animal y el medicamento, donde un veterinario sólo le puede aplicar un único tratamiento a un animal determinado:

- $A, N \rightarrow T$

> *En un tratamiento, a cada animal sólo se le puede aplicar un medicamento.*

Por último, el enunciado indica que aplicando un determinado tratamiento sólo se le puede recetar un medicamento a un animal concreto. Es decir, el atributo *tratamiento* y el *número de orden* del animal implican funcionalmente al *medicamento*:

- $T, N \rightarrow M$

Para finalizar con este apartado, se resume todo lo visto en él mediante la creación de la relación universal para la base de datos del centro farmacéutico. Esta relación se puede definir como R(AT, DEP), donde AT es el conjunto de atributos que la forman y DEP un conjunto de dependencias funcionales entre ellos:

AT = {*C, N, A, D, M, T*}

DEP = {

(1) $C \rightarrow N$

(2) $N \rightarrow C$

(3) $A \rightarrow D$

(4) $D \rightarrow A$

(5) $T \rightarrow N$

(6) $T \rightarrow D$

(7) $A, N \rightarrow M$

(8) $A, N \rightarrow T$

(9) $T, N \rightarrow M$

}

RECUBRIMIENTO MINIMAL

Se comprueba, con el mismo procedimiento que en problemas anteriores, si todas las dependencias son elementales, si existe algún atributo extraño, y, por último, si existe alguna dependencia redundante.

No se realiza todo el cálculo de forma detallada, simplemente se concluye que en este problema todas las dependencias son elementales y no existe ningún atributo extraño en ellas. Pero sí existe alguna dependencia redundante.

De hecho pueden hallarse distintos conjuntos de dependencias minimales, dependiendo de qué dependencia del conjunto DEP se considera redundante, pero sólo se detalla el cálculo de las dependencias redundantes para uno de estos conjuntos minimales.

El primer paso es identificar los atributos equivalentes. En este caso, las dependencias (1) y (2) indican que los atributos C y N son equivalentes. Igualmente, las dependencias (3) y (4) indican que lo son los atributos A y D.

Una vez que se han determinado qué atributos son equivalentes, para facilitar el cálculo de las dependencias redundantes, se pueden realizar sustituciones entre ellos. El "truco" se encuentra en elegir uno de los dos atributos equivalentes como representante, y sustituir todas las apariciones del otro atributo por el representante en las demás dependencias funcionales.

Por ejemplo, se elige el atributo N y el atributo D como representantes. De esta forma, el conjunto de dependencias (DEP) de R se puede sustituir por el conjunto DEP1 siguiente:

DEP1 = {

(a) $C \rightarrow N$

(b) $N \rightarrow C$

(c) $A \rightarrow D$

(d) $D \rightarrow A$

(e) $T \rightarrow N$

(f) $T \rightarrow D$

(g) $D, N \rightarrow M$ (sustitución de A por D)

(h) $D, N \rightarrow T$ (idem)

(i) $T, N \rightarrow M$

}

Además, se comprueba que la dependencia (i)$T, N \rightarrow M$ es una dependencia redundante, ya que se puede deducir a partir del conjunto de dependencias DEP' = {DEP1 – $\{T,N \rightarrow M\}$}. En este caso es la única dependencia redundante de DEP1. Se deja su comprobación al lector.

- $(T, N)^+_{DEP'} = \{T, N, D, M, \ldots\}$
  - (f): $T \rightarrow D$, luego $(T, N)^+_{DEP'} = \{T, N, D\}$
  - (g): $D, N \rightarrow M$, luego $(T, N)^+_{DEP'} = \{T, N, D, M\}$
  - ...
- Como M se determina a partir de DEP', la dependencia (i) es redundante.

Con todo lo visto hasta el momento en este apartado se concluye que uno de los posibles conjuntos mínimos de dependencias funcionales puede ser DEP^m, donde:

DEP^m = {

(a) $C \rightarrow N$

(b) $N \rightarrow C$

(c) $A \rightarrow D$

(d) $D \rightarrow A$

(e) $T \rightarrow N$

(f) $T \rightarrow D$

(g) $D, N \rightarrow M$

(h) $D, N \rightarrow T$

}

FORMA NORMAL DE LA RELACIÓN

El cálculo del nivel de normalización en el que se encuentra una relación se basa en los atributos principales y no principales de la relación, y para hallar cuáles son estos atributos es necesario el cálculo de las claves candidatas de la relación.

En primer lugar, se sabe que existen dos parejas de atributos equivalentes.

Por otro lado, calculando los cierres transitivos de los atributos de R se tiene:

- $C^+ = \{C, N\}$
- $N^+ = \{N, C\}$ (equivalente a C)
- $A^+ = \{A, D\}$
- $D^+ = \{D, A\}$ (equivalente a A)
- $T^+ = \{T, N, C, D, A, M\}$
  - (e): $T \rightarrow N$, luego $T^+ = \{T, N\}$
  - (f): $T \rightarrow D$, luego $T^+ = \{T, N, D\}$
  - (b): $N \rightarrow C$, luego $T^+ = \{T, N, D, C\}$
  - (d): $D \rightarrow A$, luego $T^+ = \{T, N, D, C, A\}$
  - (g): $D, N \rightarrow M$, luego $T^+ = \{T, N, D, C, A, M\}$
- $M^+ = \{M\}$

Del cálculo del cierre de los atributos se puede observar que el atributo T implica a todos los atributos de R (AT), luego T es clave candidata.

Por otro lado, se tiene la relación $A, N \rightarrow T$, por lo que también el conjunto $\{A, N\}$ implica a todos los atributos de R, y no existe ningún subconjunto de $\{A,N\}$ que determina a todo AT, por lo que se trata de otra clave candidata.

Debido a que permutar los atributos equivalentes no altera el resultado, se puede sustituir el atributo A por D y el atributo N por C, por lo que los conjuntos $\{D, N\}$, $\{A, C\}$ y $\{D, C\}$ también son claves candidatas.

Resumiendo:

- Claves candidatas: $\{T\}$, $\{A, N\}$, $\{D, N\}$, $\{A, C\}$ y $\{D, C\}$
- Atributos principales: $\{T, A, N, D, C\}$
- Atributos no principales: $\{M\}$

La relación se encuentra en 3FN, debido a que el único atributo no principal (M) tiene dependencia funcional completa respecto de la clave $\{D,N\}$ (2FN) y no existen dependencias transitivas (3FN), pero no se encuentra en FNBC, ya que no todo determinante es clave candidata. Por ejemplo, $C \rightarrow D$, $D \rightarrow C$,...

NORMALIZACIÓN

Por último, se realiza la normalización de la relación R mediante el proceso de análisis, construyendo a su vez el árbol de descomposición.

Paso 1: Recubrimiento minimal y claves candidatas

Los primeros pasos son hallar el recubrimiento minimal de R y calcular las claves candidatas, atributos principales y no principales. Todo esto se ha hallado en apartados anteriores.

Paso 2: Descomposición de R

Para realizar la descomposición de una relación, en primer lugar se agrupan todas las dependencias funcionales que poseen el mismo implicante o equivalentes. Como se comentó en apartados anteriores, en este problema existen dos parejas de atributos equivalentes, por lo tanto se agrupan.

Se elige uno de estos grupos para comenzar la descomposición, por ejemplo, el grupo formado por las dependencias $\{A \rightarrow D, D \rightarrow A\}$. Se crea una relación con ellas (R_1) y otra relación con el resto de las dependencias de R y con el resto de los

atributos de R más el atributo clave de la nueva relación. Gráficamente se muestran las relaciones resultantes de esta descomposición en la figura 3.20

La relación R_1 (cuyas claves candidatas son $\{A\}$ y $\{D\}$) se encuentra en 5FN, ya que no existen atributos no principales (3FN), todo determinante es clave (FNBC) y no existen más que dependencias funcionales. Sin embargo, la relación R_1' se encuentra, al igual que R, en 3FN, por lo que se continúa su descomposición.

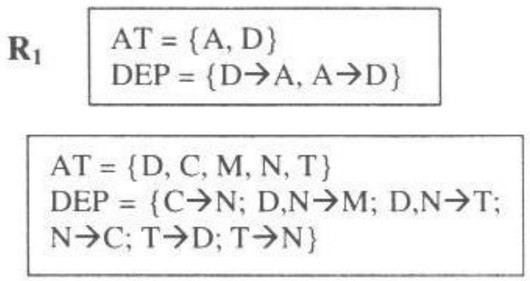


Figura 3.20 Descomposición de R

Paso 3: Descomposición de R_1'

Para descomponer la relación R_1', se elige el otro grupo de dependencias con los atributos equivalentes, el formado por el conjunto $\{C \rightarrow N, N \rightarrow C\}$. Se crea una nueva relación, R_2, con estas dependencias, y otra relación temporal R_2', con el resto de las dependencias y el resto de los atributos de R_1' más el atributo representante de R. Gráficamente se muestra en la figura 3.21

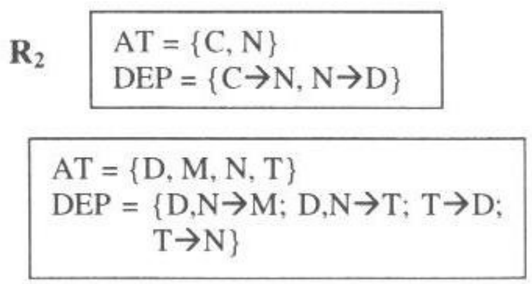


Figura 3.21. Descomposición de R_1'

La relación R_2, al igual que ocurre con la relación R_1, se encuentra en 5FN. La relación R_2', cuyas claves son $\{T\}$, $\{D, N\}$, también se encuentra en 5FN, ya que se encuentra en 3FN, todo determinante es clave (FNBC), y no existen dependencias multivaluadas ni en combinación. Por lo tanto, se ha llegado al nivel de normalización máximo de las relaciones. El árbol de descomposición final es el que aparece en la figura 3.22.

Como se apuntó en apartados anteriores, al no tratarse del único conjunto minimal de dependencia válido, existen otras descomposiciones de R igualmente correctas como solución para este problema.

Por otro lado, se observa que la normalización no ha implicado ninguna pérdida de dependencias ni de información.

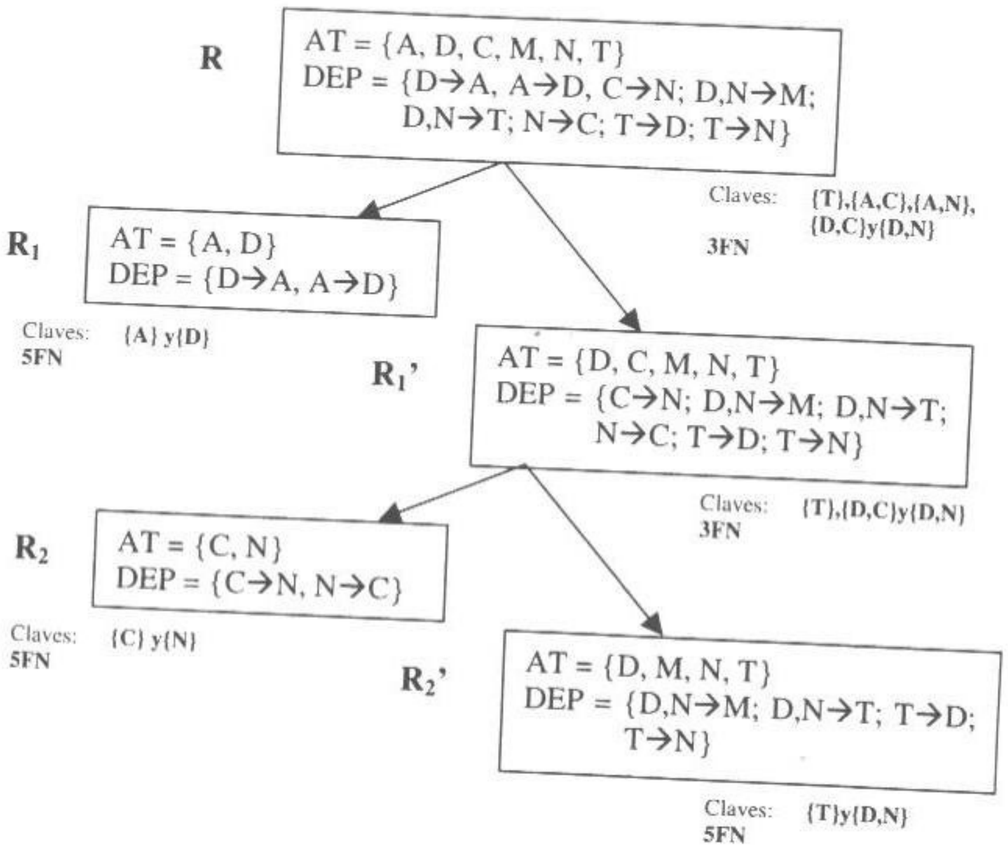


Figura 3.22. Árbol de descomposición de R

PROBLEMA 8

Enunciado

Una empresa de producción de automóviles desea crear una base de datos relacional aplicando la teoría de la normalización. Las especificaciones que tiene son las siguientes:

a) Un suministrador se identifica por su código (Cs) y por su nombre (N).

b) Toda pieza tiene un número de pieza (P) y un nombre (Np) únicos.

c) Para cada modelo de vehículo (M), un suministrador suministra una determinada cantidad (C) de cada pieza.

d) Cada pieza de un suministrador está situada en una única dirección (D) de una determinada ciudad (Ci).

e) A una dirección nunca le corresponde más de un distrito postal (Dp) en cada ciudad.

f) Una pieza de un suministrador que se suministra para un modelo no puede situarse en dos distritos postales distintos.

g) Cada distrito postal está asignado a una única ciudad, aunque a una misma ciudad le pueden corresponder varios distritos postales.

h) El precio (Pr) de las piezas es independiente del suministrador que las provee.

i) Interesa almacenar el número de serie (Ns) de cada vehículo (que es único) con el color (Co) del vehículo. A efectos de producción, los vehículos se identifican por un número consecutivo (Nu) dentro de cada modelo.

Se pide:

1) Determinar las dependencias funcionales.
2) Hallar un recubrimiento minimal válido.
3) Determinar la forma normal en la que se encuentra la relación, explicando el proceso que le ha permitido llegar a la correspondiente conclusión.

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

El análisis de los supuestos semánticos del enunciado se detallan a continuación.

Un suministrador se identifica por su código (Cs) y por su nombre (N).

En la base de datos se desea guardar la información del código y nombre de cada suministrador de la empresa de automóviles, de los que se sabe que el suministrador se puede identificar tanto por su código (*Cs*) como por su nombre (*N*), luego, tanto uno como el otro implicarán funcionalmente al resto de los atributos del suministrador, y por consiguiente se tienen las siguientes dependencias:

- $Cs \rightarrow N$
- $N \rightarrow Cs$

Toda pieza tiene un número de pieza (P) y un nombre (Np) únicos.

Por otro lado, de cada pieza del automóvil se desea guardar su número y nombre, que, al igual que ocurre con los suministradores, ambos identifican la pieza, y por lo tanto:

- $P \rightarrow Np$
- $Np \rightarrow P$

Para cada modelo de vehículo (M), un suministrador (Cs) suministra una determinada cantidad (C) de cada pieza (P).

También se sabe que un suministrador sólo puede suministrar una cierta cantidad de piezas de un modelo de un vehículo determinado. Entonces, se tiene que los atributos *Cs*, *M* y *P* implican la cantidad de la pieza de un modelo vendida por el suministrador:

- $Cs, M, P \rightarrow C$

Cada pieza (P) de un suministrador (Cs) está situada en una única dirección (D) de una determinada ciudad (Ci).

Esto indica que para cada valor del conjunto de atributos {*P*, *Cs*} se tiene un único valor de *D* y *Ci*, luego existe la dependencia funcional $\{Cs,P \rightarrow D,Ci\}$, o, lo que es lo mismo y es más indicado para su descomposición (para que una dependencia sea elemental es necesario que exista un único atributo implicado por cada dependencia):

- $Cs, P \rightarrow D$
- $Cs, P \rightarrow Ci$

A una dirección nunca le corresponde más de un distrito postal (Dp) en cada ciudad.

También se sabe que una dirección (*D*) en una cierta ciudad (*Ci*) tiene un único distrito postal (*Dp*), luego existe otra dependencia funcional entre estos atributos:

- $D, Ci \rightarrow Dp$

Una pieza de un suministrador que se suministra para un modelo no puede situarse en dos distritos postales distintos.

Por otro lado, un suministrador sólo puede suministrar una pieza para un modelo en un único distrito postal, por lo tanto, para cada valor del conjunto de atributos {*P*, *Cs* y *M*} existe un único valor posible para *Dp*, por lo que:

- $Cs, M, P \rightarrow Dp$

  Cada distrito postal está asignado a una única ciudad, aunque a una misma ciudad le pueden corresponder varios distritos postales.

Este supuesto semántico indica que el distrito postal determina funcionalmente a la ciudad, pero la ciudad no determina funcionalmente a los distritos postales, ya que pueden existir varios en la misma ciudad. Por lo tanto:

- $Dp \rightarrow Ci$

  El precio (Pr) de las piezas es independiente del suministrador que las provee.

Por lo tanto, una determinada pieza siempre tiene el mismo precio, luego:

- $P \rightarrow Pr$

  Nos interesa almacenar el número de serie (Ns) de cada vehículo (que es único) con el color (Co) del vehículo. A efectos de producción, los vehículos se identifican por un número consecutivo (Nu) dentro de cada modelo.

Por último, interesa guardar el número de serie y el color de cada vehículo, donde el número de serie identifica completamente al vehículo, y por lo tanto, implica funcionalmente al resto de sus atributos ($Ns \rightarrow Co, M, Nu$). Además, el número *Nu* del vehículo en cada modelo (*M*) también identifica el número de serie del vehículo, por lo que $Nu, M \rightarrow Ns$. Si estas dependencias se transforman en dependencias funcionales elementales, se tiene:

- $Ns \rightarrow Co$
- $Ns \rightarrow M$
- $Ns \rightarrow Nu$
- $Nu, M \rightarrow Ns$

Para finalizar este apartado, se pueden resumir todos los cálculos anteriores mediante la relación universal (R) de la base de datos de la empresa de producción de vehículos. R está compuesta por un conjunto de atributos (AT) y un conjunto de dependencias (DEP) que se detallan a continuación:

R (AT, DEP), donde:

AT = {*Cs, N, P, Np, M, C, D, Ci, Dp, Pr, Co, Un, Ns*}

DEP = {

(1) $Cs \rightarrow N$

(2) $N \rightarrow Cs$

(3) $P \rightarrow Np$

(4) $Np \rightarrow P$

(5) $Cs, M, P \rightarrow C$

(6) $P, Cs \rightarrow D$

(7) $P, Cs \rightarrow Ci$

(8) $D, Ci \rightarrow Dp$

(9) $P, Cs, M \rightarrow Dp$

(10) $Dp \rightarrow Ci$

(11) $P \rightarrow Pr$

(12) $Ns \rightarrow Co$

(13) $Ns \rightarrow Nu$

(14) $Ns \rightarrow M$

(15) $M, Nu \rightarrow Ns$

}

RECUBRIMIENTO MINIMAL

En este apartado, se comprueba si el conjunto de las dependencias halladas anteriormente (DEP) es irredundante, es decir, que todas las dependencias sean elementales, no redundantes y no existe ningún atributo extraño en ninguna de ellas.

Dependencias elementales

Con respecto a las dependencias elementales, en los apartados anteriores se ha tenido cuidado en definir las dependencias para que sean elementales, con un único atributo implicado, no triviales y plenas, por lo que no se detalla su comprobación.

Atributos extraños

En una de las dependencias de DEP existe un atributo extraño, en concreto, en las dependencias (9) $\{P, Cs, M \rightarrow Dp\}$ el atributo *M* es un atributo extraño, ya que se

uede deducir el atributo *Dp* a partir de *P* y *Cs*. A continuación se muestra el cálculo etallado:

$(P,Cs)^+ = \{ P, Np, Pr, Cs, N, D, Ci, Dp, ... \}$

- $P^+ = \{P, Np, Pr\}$, luego $(P,Cs)^+ = \{P, Np, Pr\}$
- $Cs^+ = \{Cs, N\}$, luego $(P,Cs)^+ = \{P, Np, Pr, Cs, N\}$
- (6): P, Cs → D, luego $(P,Cs)^+ = \{P, Np, Pr, Cs, N, D\}$
- (7): P, Cs → Ci, luego $(P,Cs)^+ = \{P, Np, Pr, Cs, N, D, Ci\}$
- (8): D, Ci → Dp, luego $(P,Cs)^+_{DEP'} = \{ P, Np, Pr, Cs, N, D, Ci, Dp \}$
- ...

- Como el atributo *Dp* pertenece al cierre transitivo de $\{P, Cs\}$, el atributo *M* es un atributo extraño en la relación, por lo que, la dependencia (9) se transforma en $\{P,Cs \rightarrow Dp\}$.

Dependencias redundantes

La dependencia (9) $\{P,Cs \rightarrow Dp\}$ es una dependencia redundante, ya que se puede deducir a partir del conjunto de dependencias DEP' = {DEP1 – $\{P,Cs \rightarrow Dp\}$} (nota: no se realiza todo el cálculo del cierre, solamente hasta el punto que interesa):

- $(P,Cs)^+_{DEP'} = \{P, Cs, D, Ci, Dp, ...\}$
  - (6): $P,Cs \rightarrow D$, luego $(P,Cs)^+_{DEP'} = \{P, Cs, D\}$
  - (7): $P,Cs \rightarrow Ci$, luego $(P,Cs)^+_{DEP'} = \{P, Cs, D, Ci\}$
  - (8): $D, Ci \rightarrow Dp$, luego $(P,Cs)^+_{DEP'} = \{P, Cs, D, Ci, Dp\}$
  - ...
- Como *Dp* se determina a partir de DEP', la dependencia (9) es redundante.

Por lo tanto, para concluir con este apartado y a modo de resumen, se indica el conjunto de dependencias mínimo hallado:

$DEP^m = \{$

$Cs \rightarrow N$

$N \rightarrow Cs$

$P \rightarrow Np$

$Np \rightarrow P$

$Cs, M, P \rightarrow C$

$P, Cs \rightarrow D$

$P, Cs \rightarrow Ci$

$D, Ci \rightarrow Dp$

$Dp \rightarrow Ci$

$P \rightarrow Pr$

$Ns \rightarrow Co$

$Ns \rightarrow Nu$

$Ns \rightarrow M$

$M, Nu \rightarrow Ns$

$\}$

FORMA NORMAL DE LA RELACIÓN

El cálculo de la forma normal de una relación se basa en los atributos principales y no principales de la misma, por lo que, el primer paso es el cálculo de la clave de la relación. Este cálculo se basa en el cierre transitivo de sus descriptores, por lo tanto, en primer lugar, se realiza el cálculo de los cierres transitivos de cada atributo de R de forma individual:

- $Cs^+ = \{Cs, N\}$
- $N^+ = \{Cs, N\}$, son equivalentes los atributos *Cs* y *N*.
- $P^+ = \{P, Np, Pr\}$
- $Np^+ = \{P, Np, Pr\}$, también son equivalente *P* y *Np*
- $M^+ = \{M\}$
- $C^+ = \{C\}$
- $D^+ = \{D\}$

- $Ci^+ = \{Ci\}$
- $Dp^+ = \{Dp, Ci\}$
- $Pr^+ = \{Pr\}$
- $Ns^+ = \{Ns, Co, Nu, M\}$
- $Co^+ = \{Co\}$
- $Nu^+ = \{Nu\}$

Se eligen los atributos cuyos cierres tienen una cardinalidad mayor (*Ns, Np, Cs*), que pueden formar parte de alguna clave, y se halla su cierre transitivo (se realizar algunos pasos intermedios en el cálculo):

- $(Np, Cs, Ns)^+_{DEP} = AT$, luego es clave candidata al no existir ningún subconjunto del mismo que determine a todo AT.
  - Suma de sus cierres transitivos: $(Np, Cs, Ns)^+_{DEP} = \{P, Np, Pr, Cs, N, Ns, Co, Nu, M\}$
  - (6): P, Cs → D, luego $(Np, Cs, Ns)^+_{DEP} = \{P, Np, Pr, Cs, N, Ns, Co, Nu, M, D\}$
  - (7): P, Cs → Ci, luego $(Np, Cs, Ns)^+_{DEP} = \{P, Np, Pr, Cs, N, Ns, Co, Nu, M, D, Ci\}$
  - (8): D, Ci → Dp, luego $(Np, Cs, Ns)^+_{DEP} = \{P, Np, Pr, Cs, N, Ns, Co, Nu, M, D, Ci, Dp\}$
  - (5): Cs, M, P → C, luego $(Np, Cs, Ns)^+_{DEP} = \{P, Np, Pr, Cs, N, Ns, Co, Nu, M, D, Ci, Dp, C\}$

Por otro lado, las parejas de atributos $\{Cs, N\}$ y $\{P, Np\}$ son equivalentes, por lo que también son claves candidatas los conjuntos $\{Np, N, Ns\}$, $\{P, Cs, Ns\}$ y $\{P, N, Ns\}$, que se hallan permutando los atributos equivalentes.

Otra clave candidata de R es la que se calcula a partir del conjunto de atributos que implica al que tiene el cierre con una cardinalidad mayor, es decir, en este caso, los atributos $\{M, Nu\}$ implican al atributo *Ns*, que implica a un número mayor de atributos. Se sustituye en todas las claves anteriores el atributo *Ns* por el conjunto $\{M, Nu\}$, de forma que se tienen las siguientes superclaves: $\{Np, Cs, M, Nu\}$, $\{Np, N, M, Nu\}$, $\{P, Cs, M, Nu\}$ y $\{P, N, M, Nu\}$. Además son claves candidatas al no existir ningún subconjunto de las mismas que determina a todo AT.

Por último, se concluye (aunque no se explica de forma detallada), que éstas son todas las claves de R.

A modo de resumen:

- Claves candidatas: {Np, Cs, Ns}, {Np, N, Ns}, {P,Cs, Ns}, {P, N, Ns}, {Np, Cs, M, Nu}, {Np, N, M, Nu}, {P, Cs, M, Nu} y {P, N, M, Nu}
- Atributos principales: $\{Np, P, Cs, N, Ns, M, Nu\}$
- Atributos no principales: $\{Pr, Co, D, Ci, Dp, C\}$

El nivel de normalización de la relación R es 1FN. No se encuentra en 2FN, ya que existen atributos no principales que dependen funcionalmente de parte de alguna clave. Por ejemplo, el atributo *Pr* depende de *P*, pero no de la clave completa.

NORMALIZACIÓN

Se aplica el método de análisis o descomposición para dividir la relación inicial, R, en varias con un nivel de normalización mayor.

Paso 1: Recubrimiento minimal y claves

El primer paso del método es hallar el recubrimiento minimal de la relación y la forma normal en la que se encuentra (por si no es necesario la descomposición). Estos puntos ya se han resuelto en apartados anteriores.

Paso 2: Descomposición de R

El siguiente paso es descomponer la relación inicial, R. Para ello se agrupan los atributos equivalentes y las dependencias con el mismo implicante. Se elige uno de estos grupos, con cuidado de que el implicado no se encuentre en ninguna otra dependencia, y se crea una nueva relación con el grupo.

Por ejemplo, se crea una nueva relación (R_1) con el grupo formado por los atributos equivalentes $\{Cs, N\}$ y las dependencias $\{Cs \rightarrow N, N \rightarrow Cs\}$.

Se crea al mismo tiempo una nueva relación temporal (R_1') que se sigue descomponiendo hasta llegar a la forma normal deseada con todos los atributos de R menos los que no son principales en R_1, en este caso, en R_1 todos los atributos son principales, por lo que en principio todos los atributos de R se encuentran en R_1', pero debido a que *N* no participa en ninguna dependencia de R_1', se omite. Gráficamente estas dos relaciones se muestran en la figura 3.23.

R_1

AT = {Cs, N}
DEP = {Cs→N, N→Cs}

R_1'

AT = {Cs, P, Np, M, C, D, Ci, Dp, Pr, Ns, Co, Nu}
DEP = {P→Np; Np→P; Cs,M,P→C; P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci; P→Pr; Ns→Co; M,Nu→Ns; Ns→Nu; Ns→M}

Figura 3.23. Descomposición de R

Las claves candidatas de la relación R_1 son *{Cs}* y *{N}*. Además, se sabe que R_1 se encuentra en 5FN, ya que se encuentra en 3FN por no tener atributos no principales, en FNBC, ya que todos los determinantes son clave, y en 5FN al no poseer dependencias multivaluadas ni en combinación.

Por otro lado, la relación R_1' se encuentra, por la misma razón que la relación R, en 1FN, conservándose igualmente las claves candidatas.

Paso 3: Descomposición de R_1'

En este paso se elige el otro grupo formado por los otros dos atributos equivalentes, donde las relaciones se construyen como en el paso anterior. Las claves de R_2 son *{P}* y *{Np}*, y se encuentra en 5FN. Gráficamente se muestra la descomposición en la figura 3.24.

R_2

AT = {P, Np}
DEP = {Np→P, P→Np}

R_2'

AT = {Cs, P, M, C, D, Ci, Dp, Pr, Ns, Co, Nu}
DEP = {Cs,M,P→C; P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci; P→Pr; Ns→Co; M,Nu→Ns; Ns→Nu; Ns→M}

Figura 3.24. Descomposición de R_1'

Por otro lado, la relación R_2' tiene las mismas claves candidatas que R y que R_1', y, al igual que las anteriores, también se encuentra en 1FN, pero no en 2FN, ya que aún existen atributos no principales que dependen funcionalmente de parte de la clave, por ejemplo, en la dependencia *Cs,M,P →C*. Al igual que en el paso 2, se ha omitido el atributo *Np* en R_2' al no participar en ninguna dependencia de la relación.

Paso 4: Descomposición de R_2'

En este paso, se elige una nueva dependencia, donde el implicado no se encuentre en ninguna otra, por ejemplo, la dependencia *Cs,M,P →C*, ya que *C* no participa en el resto de las dependencias. Se crean dos nuevas relaciones (R_3 y R_3') que se muestran en la figura 3.25.

R_3

AT = {Cs, M, P, C}
DEP = {Cs,M,P → C}

R_3'

AT = {Cs, P, M, D, Ci, Dp, Pr, Ns, Co, Nu}
DEP = {P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci; P→Pr; Ns→Co; M,Nu→Ns; Ns→Nu; Ns→M}

Figura 3.25. Descomposición de R_2'

La clave de la relación R_3 es el conjunto de los atributos *{Cs,M,P}*, y por lo tanto se encuentra en 5FN, ya que sólo existe un atributo no principal, que depende completamente de la clave (3FN), todo los determinantes son claves candidatas (FNBC) y sólo existen dependencias funcionales.

Por otro lado, la relación R_3' se encuentra en 1FN, pero no en superiores, por la misma razón que R, R_1' y R_2'. Por lo tanto, se continúa descomponiendo en otras relaciones con mayor grado de normalización.

Paso 4: Descomposición de R_3'

Se elige una nueva dependencia para crear otra relación, por ejemplo, la dependencia *P →Pr*, ya que el implicado no participa en ninguna otra. De esta forma, se crean las relaciones de la figura 3.26.

R_4

AT = {P, Pr}
DEP = {P→Pr}

R_4'

AT = {Cs, P, M, D, Ci, Dp, Pr, Ns, Co, Nu}
DEP = {P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci; Ns→Co; M,Nu→Ns; Ns→Nu; Ns→M}

Figura 3.26. Descomposición de R_3'

En este caso, la relación R_4 se encuentra en 5FN y su única clave candidata es $\{P\}$, mientras que la relación R_4' se encuentra aún en 1FN por la misma razón que R, R_1', R_2' y R_3', por lo tanto, se continúa su descomposición.

Paso 5: Descomposición de R_4'

La siguiente dependencia que se elige para la descomposición es $Ns \rightarrow Co$, donde su implicado no pertenece a ninguna otra. Se crean las dos relaciones que se muestran en la figura 3.27, donde R_5 se encuentra en 5FN y su clave es $\{Ns\}$, por la misma razón que R_4, y la relación R_5' se encuentra en 1FN, al igual que las anteriores relaciones temporales.

Paso 6: Descomposición de R_5'

La elección de la siguiente dependencia funcional no es tan simple como en los pasos anteriores, ya que no existe ninguna cuyo determinante no pertenezca a ninguna otra. De esta forma, si se eligiera la dependencia $P,Cs \rightarrow D$, no se podría separar de la dependencia $D,Ci \rightarrow Dp$, ya que D se encuentra como determinante. Por otro lado, esta última dependencia no se puede separar de $Dp \rightarrow Ci$, por la misma razón, ni ésta de $P,Cs \rightarrow Ci$.

R_5

AT = {Ns, Co}
DEP = {Ns→Co}

R_5'

AT = {Cs, P, M, D, Ci, Dp, Pr, Ns, Nu}
DEP = {P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci; M,Nu→Ns; Ns→Nu; Ns→M}

Figura 3.27. Descomposición de R_4'

Por lo tanto, se crea una nueva relación con estas cuatro dependencias y otra con el resto. Se muestran en la figura 3.28.

R_6

AT = {M, Nu, Ns}
DEP = {M,Nu→Ns; Ns→Nu; Ns→M}

R_7

AT = {P, Cs, D, Ci, Dp}
DEP = {P,Cs→D; P,Cs→Ci; D,Ci→Dp; Dp→Ci}

Figura 3.28. Descomposición de R_5'

La relación R_6 tiene como claves candidatas los conjuntos $\{Ns\}$ y $\{M, Nu\}$, y, por lo tanto, todos los atributos son principales. De esta forma, la relación se encuentra en 5FN.

La clave candidata de la relación R_7 es el conjunto de atributos $\{P,Cs\}$, por lo tanto, la relación se encuentra en 3FN, ya que aún existen atributos no principales que dependen transitivamente de parte de la clave. Si se quisiera descomponer esta relación en otras con un nivel de normalización mayor se perderían dependencias.

Por ejemplo, si se quisiera descomponer eligiendo la dependencia $P,Cs \rightarrow D$ se tendrían las siguientes relaciones:

- $R_{7a}(\{P, Cs, D\},\{P, Cs \rightarrow D\})$
- $R_{7a}'(\{P, Cs, Ci, Dp\},\{P, Cs \rightarrow Ci; Dp \rightarrow Ci\})$.

Se observa que se han perdido las dependencias donde participaba el atributo D, en concreto, se ha perdido la dependencia $D, Ci \rightarrow Dp$.

La relación R_{7a} se encuentra en 5FN, mientras que la relación R_{7a}' se encuentra en 1FN, ya que su clave candidata es $\{P, Cs, Dp\}$ y el atributo no principal Ci depende parcialmente de ella. Si se quisiera seguir descomponiendo esta relación se perdería de nuevo otra dependencia, por ejemplo:

- $R_{7b}(\{P, Cs, Ci\},\{P, Cs \rightarrow Ci\})$
- $R_{7b}'(\{P, Cs, Dp\},\{\phi\})$, donde se ha perdido la dependencia $Dp \rightarrow Ci$.

En las figuras 3.29 y 3.30 se muestra el árbol de descomposición sin pérdidas de dependencias funcionales, indicando en cada relación cuál es su clave y en qué forma normal se encuentra.

En el enunciado se pide que se realice el proceso de descomposición mediante el método de análisis, pero de igual forma se podría haber realizado mediante el de síntesis, ya que no existen dependencias multivaluadas ni de combinación, y, por lo tanto, el resultado sería igualmente válido.

© RA-MA

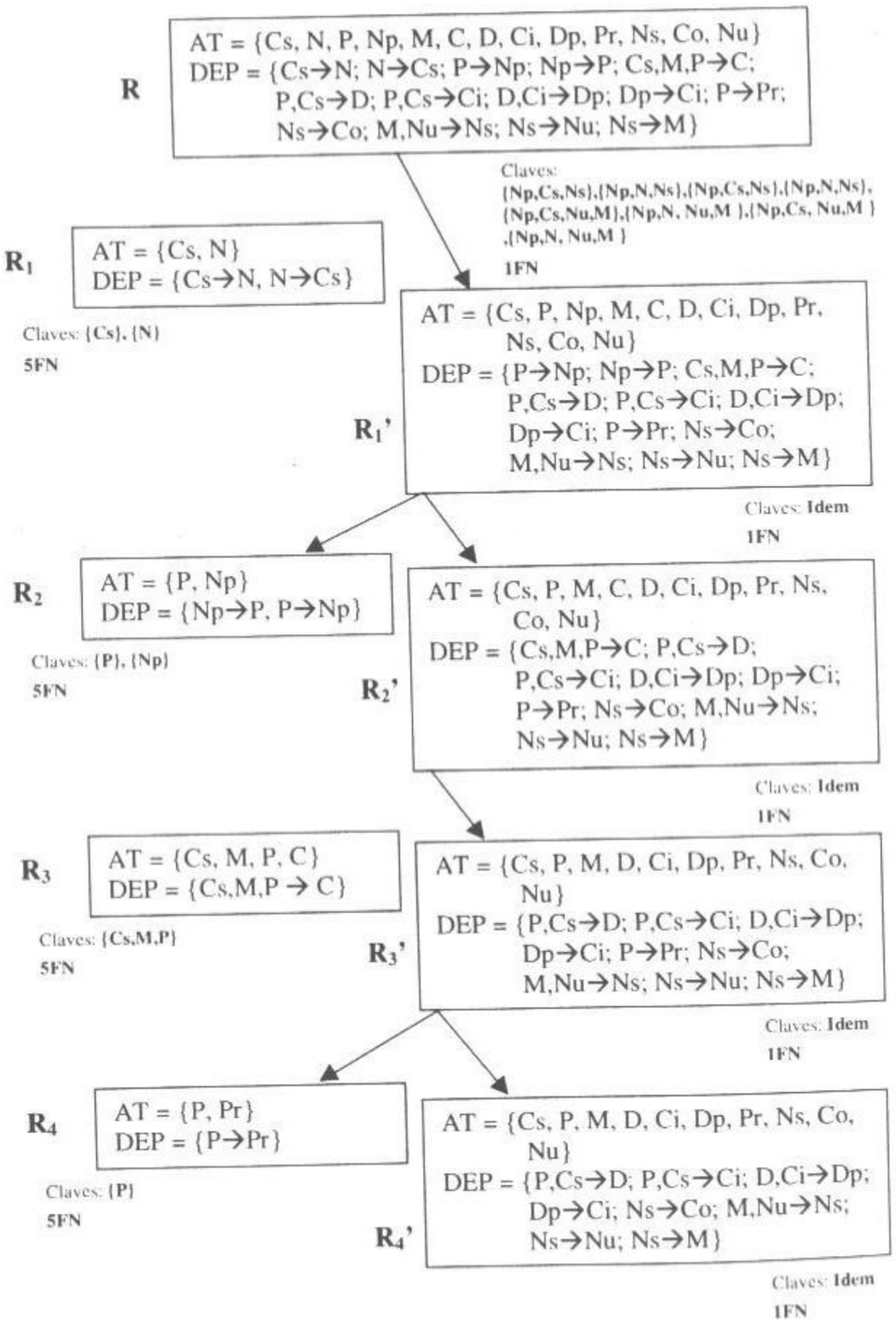


*Figura 3.29. Árbol de descomposición **hasta R_4'***

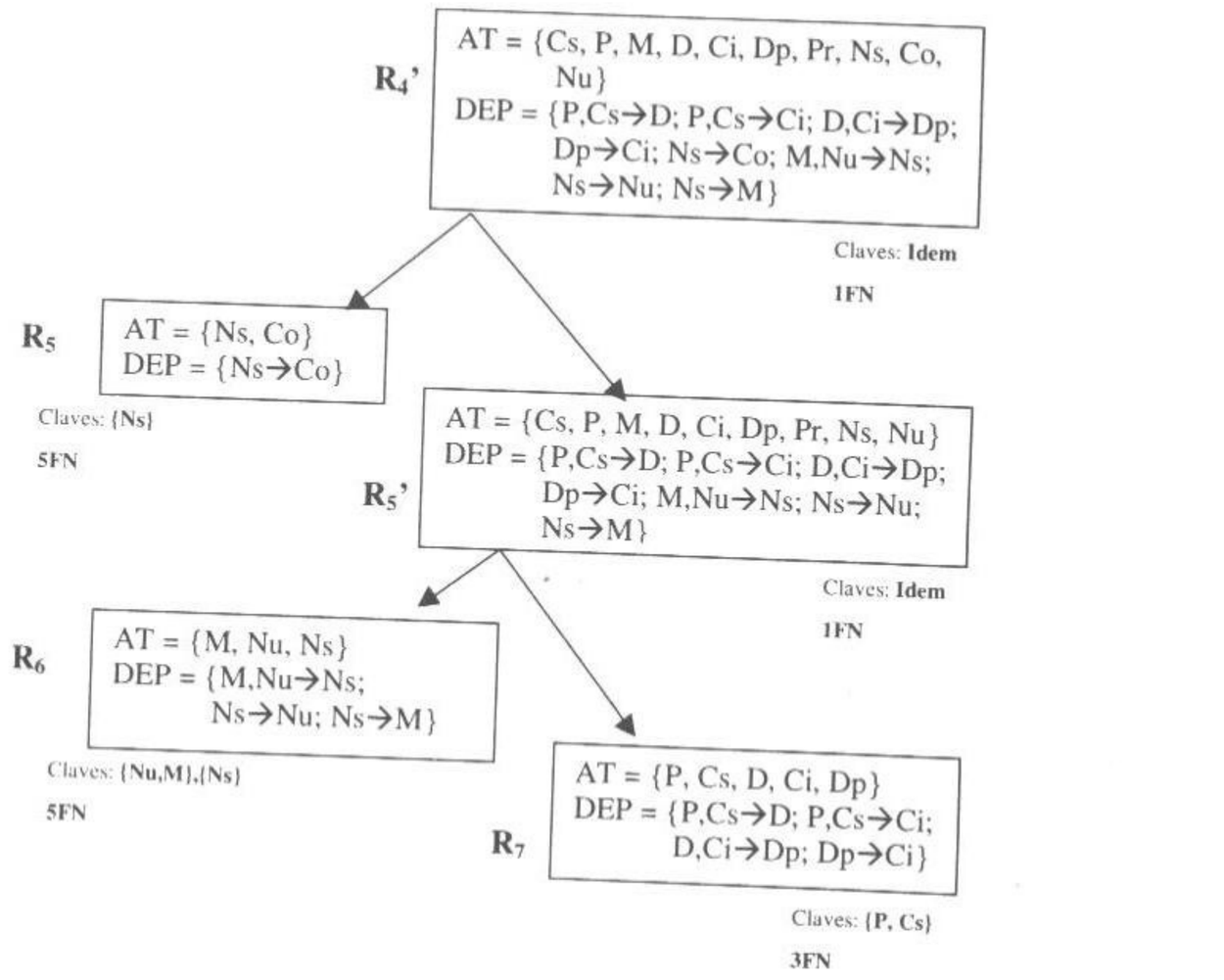


*Figura 3.30 Árbol de descomposición **desde R_4'***

PROBLEMA 9

Enunciado

Una cadena de agencias inmobiliarias desea crear una base de datos relacional aplicando la teoría de la normalización.

El Departamento de Informática de la cadena tiene unos estándares de nomenclatura para sus diseños, en lo que atañe a los nombres resumidos, que también se aplica en la base de datos; consisten (entre otras normas que no nos afectan en este caso) en dar como nombre resumido (siempre que no se especifique lo contrario) de cada entidad o de cada atributo la inicial del correspondiente nombre, y cuando coinciden las iniciales de dos atributos distintos se añade la letra de la entidad a la que pertenecen; para referirnos a la clave de una entidad se pone la letra inicial de dicha

entidad, y si ésta coincide con el nombre de algún atributo se pone una C delante de la misma.

Las especificaciones que nos dan son las siguientes:

a) Cada agencia, identificada por un código de agencia (CA), tiene un nombre distinto (NA). Las agencias tienen un único titular que se identifica por su DNI (DT) o por un código (CT). Además, una persona sólo puede ser titular de una agencia.

b) Las agencias disponen de muchos inmuebles (I) para su venta o alquiler, pero un inmueble sólo puede ser ofrecido en venta o alquiler por una única persona ofertante (O). El ofertante de un inmueble se lo entrega a una única agencia de la cadena (o a su titular) durante un período de tiempo (dado por una fecha inicial (FI) por una fecha final (FF)); durante este período de tiempo no puede variar el precio mínimo (P) solicitado por el ofertante para dicho inmueble (a un ofertante, un inmueble y una fecha inicial sólo le corresponde una fecha final).

c) Del titular de una agencia dependen empleados (E), los cuales no pueden depender nada más que de un titular.

d) Las agencias tienen demandantes de inmuebles (D); cuando un demandante cambia de agencia, se le considera por la cadena como un demandante distinto.

e) Los empleados tienen inmuebles a su cargo y durante el período (fecha inicial y fecha final) que la agencia está encargada del inmueble puede cambiar el empleado encargado del mismo.

f) Cada empleado tiene que registrar el número total de horas que ha dedicado a cada ofertante y a cada demandante de un inmueble (téngase en cuenta que es preciso distinguir entre horas dedicadas a ofertantes (HO) y horas dedicadas a demandantes (HD))

Se pide:

1) Especificar las dependencias funcionales que existen entre los atributos de la base de datos (si del enunciado se deduce la inexistencia de dependencia entre un implicante y un implicado incluirla)

2) Forma normal en la que se encuentra la relación, explicando todos los pasos del proceso que le ha permitido llegar a la correspondiente conclusión.

3) Aplicar el proceso de análisis (árbol de descomposición) para obtener un esquema relacional en FNBC haciendo las observaciones que estime pertinentes.

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

En primer lugar, se realiza una lista con todos los atributos que se incluyen en la base de datos, para no perder el significado del nombre abreviado de la misma:

Nombre completo	Nombre abreviado
Código de agencia	CA
Nombre de agencia	NA
DNI del titular	DT
Código del titular	CT
Inmueble	I
Ofertante	O
Fecha inicial	FI
Fecha final	FF
Precio mínimo	P
Empleado	E
Demandante	D
Horas dedicadas a ofertantes	HO
Horas dedicadas a demandantes	HD

El análisis de los supuestos semánticos del enunciado se detallan a continuación, realizando las referencias al enunciado que se estiman oportunas:

> *Cada agencia, identificada por un código de agencia (CA), tiene un nombre distinto (NA). Las agencias tienen un único titular que se identifica por su DNI (DT) o por un código (CT). Además una persona sólo puede ser titular de una agencia.*

Este párrafo del enunciado indica que una agencia se identifica por su código, luego, al tratarse de su clave principal, implica funcionalmente al resto de los atributos de la agencia ($CA \rightarrow NA$). También se sabe que todas las agencias tienen nombres distintos, luego el nombre también podría actuar como clave primaria en la relación (clave candidata), y por lo tanto también implica funcionalmente a todos los atributos de la agencia ($NA \rightarrow CA$).

Por otro lado se sabe que cada agencia tiene un único titular, por lo que conociendo la agencia, se sabe quién es su titular, por ejemplo, tómese como atributo representativo de la agencia su código y como atributo representativo del titular su DNI ($CA \rightarrow DT$). Además un titular sólo puede serlo en una agencia determinada ($DT \rightarrow CA$). También se sabe que el titular se identifica por su DNI o por su código (relación similar a la de la agencia explicada anteriormente), luego se tienen dos dependencias funcionales más ($DT \rightarrow CT$) y ($CT \rightarrow DT$). Resumiendo todas estas dependencias se tiene:

- $CA \rightarrow NA$
- $NA \rightarrow CA$
- $CT \rightarrow DT$
- $DT \rightarrow CT$
- $CA \rightarrow DT$
- $DT \rightarrow CA$

A partir de todas estas dependencias se deduce que *CA, NA, CT Y DT* son atributos equivalentes, ya que *CA* y *NA* son equivalentes al implicarse mutuamente, lo mismo ocurre con *DT* y *CT* y además también se tiene que *CA* y *DT* se implican mutuamente.

> *Las agencias disponen de muchos inmuebles (I) para su venta o alquiler, pero un inmueble sólo puede ser ofrecido en venta o alquiler por una única persona ofertante (O). El ofertante de un inmueble se lo entrega a una única agencia de la cadena (o a su titular) durante un período de tiempo (dado por una fecha inicial (FI) por una fecha final (FF)); durante este período de tiempo no puede variar el precio mínimo (P) solicitado por el ofertante para dicho inmueble (a un ofertante, un inmueble y una fecha inicial sólo le corresponde una fecha final).*

Por otro lado, se indica que las agencias disponen de varios muebles para su venta o alquiler, luego su identificador no determina un inmueble determinado ($CA \text{-/->} I$). Sin embargo, un inmueble sólo puede ofrecerlo un ofertante, luego ($I \rightarrow O$). Además, el ofertante de un inmueble se lo entrega a una única agencia durante un período de tiempo, luego, conociendo el ofertante, el inmueble y el período de tiempo, se conoce a qué agencia ha sido entregado ($O,I,FI,FF \rightarrow CA$). Además, para el mismo determinante se tiene que el precio mínimo solicitado ha de mantenerse, luego ($O,I,FI,FF \rightarrow P$). También se sabe que para un ofertante, un inmueble y una fecha inicial sólo existe una posible fecha final, luego también se tiene la dependencia funcional ($O,I,FI \rightarrow FF$). Resumiendo todas estas dependencias, y obviando las que no lo son, se tiene:

- $I \rightarrow O$
- $O,I,FI,FF \rightarrow CA$

- $O,I,FI,FF \rightarrow P$
- $O,I,FI \rightarrow FF$

> *Del titular de una agencia dependen empleados (E), los cuales no pueden depender nada más que de un titular.*

Este párrafo indica que de un titular dependen varios empleados (no uno sólo), luego el código del titular no determina funcionalmente al empleado ($CT \text{-/->} E$), aunque un empleado sólo depende de un determinado titular, luego el empleado implica funcionalmente al titular de la agencia que su responsable ($E \rightarrow CT$):

- $E \rightarrow CT$

> *Las agencias tienen demandantes de inmuebles (D); cuando un demandante cambia de agencia, se le considera por la cadena como un demandante distinto.*

Cada agencia tiene varios demandantes de inmuebles, luego ($CA \text{-/->} D$), pero cuando un demandante cambia de agencia se le considera con un código distinto, luego cada código de demandante implica funcionalmente a una única agencia ($D \rightarrow CA$):

- $D \rightarrow CA$

> *Los empleados tienen inmuebles a su cargo y durante el período (fecha inicial y fecha final) que la agencia está encargada del inmueble puede cambiar el empleado encargado del mismo.*

Esto indica que los empleados tienen varios inmuebles a su cargo ($E \text{-/->} I$). Además, en un período determinado de tiempo, la agencia encargada del inmueble puede cambiar de empleado, luego no lo determina funcionalmente ($FI,FF,I,CA \text{ -/->} E$). Por lo tanto no se tiene ninguna información de dependencias en este párrafo del enunciado.

> *Cada empleado tiene que registrar el número total de horas que ha dedicado a cada ofertante y a cada demandante de un inmueble (téngase en cuenta que es preciso distinguir entre horas dedicadas a ofertantes (HO) y horas dedicadas a demandantes (HD))*

Por último se sabe que un empleado registra un número total de horas dedicado a cada ofertante y a cada demandante de un inmueble, luego existen dos dependencias funcionales más:

- $E,O,I \rightarrow HO$
- $E,D,I \rightarrow HD$

Para finalizar este apartado, se pueden resumir todos los cálculos anteriores mediante la relación universal (R) de la base de datos de las agencias inmobiliarias. R está compuesta por un conjunto de atributos (AT) y un conjunto de dependencias (DEP) que se detallan a continuación:

R (AT, DEP), donde:

AT = {*CA, NA, DT, CT, I, O, FI, FF, P, E, D, HO, HD*}

DEP = {

(1) $CA \rightarrow NA$

(2) $NA \rightarrow CA$

(3) $CT \rightarrow DT$

(4) $DT \rightarrow CT$

(5) $CA \rightarrow DT$

(6) $DT \rightarrow CA$

(7) $I \rightarrow O$

(8) $O,I,FI,FF \rightarrow CA$

(9) $O,I,FI,FF \rightarrow P$

(10) $O,I,FI \rightarrow FF$

(11) $E \rightarrow CT$

(12) $D \rightarrow CA$

(13) $E,O,I \rightarrow HO$

(14) $E,D,I \rightarrow HD$

}

FORMA NORMAL DE LA RELACIÓN

El primer paso para hallar la forma normal en la que se encuentra una relación es el cálculo de su recubrimiento irredundante, posteriormente se hallarán las claves de la misma, sus atributos principales y no principales. A partir de los atributos principales y no principales ya se puede determinar la forma normal en la que se encuentra.

Recubrimiento minimal

Se comprueba que el conjunto de las dependencias halladas en el apartado anterior (DEP) es irredundante. Para ello ha de cumplir que todas sus dependencias

sean elementales, que ninguna tenga atributos extraños y no exista en DEP ninguna dependencia redundante. Cada una de estas condiciones se detallan a continuación.

Dependencias elementales

Se han definido todas las dependencias para que sean elementales (un único implicado, plenas y no triviales), luego se cumple esta primera condición.

Atributos extraños

Existen atributos extraños en tres de las dependencias de DEP, se detalla el cálculo de estos atributos extraños en las dependencias implicadas:

En primer lugar, en la dependencia (8) $\{O,I,FI,FF \rightarrow CA\}$ el atributo *O* es un atributo extraño, ya que se puede deducir *CA* a partir del resto del determinante:

- $(I,FI,FF)^+ = \{ I, FI, FF, O, CA \ldots\}$
  - (7): $I \rightarrow O$, luego $(I,FI,FF)^+ = \{I, FI, FF, O\}$
  - (8 sin modificar): $O,I,FI,FF \rightarrow CA$, luego $(I,FI,FF)^+ = \{I, FI, FF, O, CA\}$
  - ...

- Por la misma razón, en la dependencia (9) $\{O,I,FI,FF \rightarrow P\}$, en la dependencia (10) $\{O,I,FI \rightarrow FF\}$ y en la dependencia (13) $\{E,O,I \rightarrow HO\}$ el atributo *O* también es un atributo extraño.
- Igual ocurre con el atributo *FF*, que es extraño en las dependencias (8) y (9), ya que en la dependencia (10) $\{O,I,FI \rightarrow FF\}$ se indica que conociendo los valores de *O, I* y *FI*, se conoce el de *FF*.

Por lo tanto, y para concluir con este apartado se realiza un resumen de las dependencias de R tras la detección de estos atributos extraños:

DEP1 = {

(a) $CA \rightarrow NA$

(b) $NA \rightarrow CA$

(c) $CT \rightarrow DT$

(d) $DT \rightarrow CT$

(e) $CA \rightarrow DT$

(f) $DT \rightarrow CA$

(g) $I \rightarrow O$

(*h*) $I,FI \rightarrow CA$,ya que ($I,FI \rightarrow FF$ y $I \rightarrow O$)

(*i*) $I,FI \rightarrow P$,ya que ($I,FI \rightarrow FF$ y $I \rightarrow O$)

(*j*) $I,FI \rightarrow FF$,ya que ($I \rightarrow O$)

(*k*) $E \rightarrow CT$

(*l*) $D \rightarrow CA$

(*m*) $E,I \rightarrow HO$,ya que ($I \rightarrow O$)

(*n*) $E,D,I \rightarrow HD$

}

Dependencias redundantes

En DEP1 no existen dependencias redundantes, ya que ninguna de ellas se puede deducir a partir del resto de las dependencias del mismo conjunto. Se deja su comprobación al lector. Por lo tanto, el conjunto de dependencias mínimo para R es:

$$DEP^m = DEP1$$

Claves de la relación y atributos principales y no principales

El cálculo de la forma normal de una relación se basa en los atributos principales y no principales de la misma, por lo que, el primer paso es realizar el cálculo de la clave de la relación. Este cálculo se basa en el cierre transitivo de sus descriptores, que se detalla a continuación:

- $CA^+ = \{CA, NA, DT, CT\}$
- $NA^+ = \{CA, NA, DT, CT\}$, equivalente al anterior
- $DT^+ = \{CA, NA, DT, CT\}$, equivalente a los anteriores
- $CT^+ = \{CA, NA, DT, CT\}$, equivalente a los anteriores
- $E^+ = \{E, CA, NA, DT, CT\}$
- $D^+ = \{D, CA, NA, DT, CT\}$
- $I^+ = \{I, O\}$
- $O^+ = \{O\}$
- $FI^+ = \{FI\}$
- $FF^+ = \{FF\}$
- $HO^+ = \{HO\}$
- $HD^+ = \{HD\}$

Se eligen los atributos cuyos cierres tienen una cardinalidad mayor, *E* y *D*, que además han de formar parte de todas las claves, ya que no son implicados por ningún conjunto de atributos. Hasta el momento se tiene que $\{E,D\}^+ = \{E,D,CA,NA,DT,CT\}$, por lo que faltan aún atributos de AT no implicados por el conjunto $\{E,D\}$. Luego $\{E,D\}$ no es clave por sí sola. Se han de añadir nuevos atributos a la clave.

Por otro lado, si se añade el atributo *I* a la clave, con las dependencias (g), (m) y (n) se obtiene que $\{E,D,I\}^+ = \{E,D,CA,NA,DT,CT,O,HO,HD\}$. Todavía este conjunto no es clave, ya que no implica funcionalmente a todos los atributos de AT (aún faltan *FI, P* y *FF*), por lo tanto, se añaden nuevos atributos al conjunto.

Por las dependencias (i) y (j) se sabe que si se añade el atributo *FI* al conjunto anterior se implican funcionalmente a *P* y *FF*, luego el conjunto $\{E,D,I,FI\}$ sí es clave candidata para la relación R, ya que además ningún subconjunto determina a todo AT. Se concluye, por último que ésta es la única clave para R. Se deja para el lector la comprobación de que no existe ninguna clave candidata más.

Resumiendo:

- Claves candidatas: $\{E,D,I,FI\}$
- Atributos principales: $\{E,D,I,FI\}$
- Atributos no principales: {CA, NA, DT, CT, O, FF, P, HO, HD}

El nivel de normalización de la relación R es 1FN. No se encuentra en 2FN, ya que existen atributos no principales que dependen funcionalmente de parte de alguna clave. Por ejemplo, el atributo *O* depende de *I*, pero no de la clave completa.

NORMALIZACIÓN MEDIANTE ANÁLISIS

Se aplica el método de análisis o descomposición para dividir la relación inicial, R, en varias con un nivel de normalización mayor.

Paso 1: Recubrimiento minimal y claves

El primer paso del método es hallar el recubrimiento minimal de la relación y la forma normal en la que se encuentra. Estos puntos ya se han resuelto en apartados anteriores.

Paso 2: Descomposición de R

El siguiente paso es descomponer la relación inicial, R, ya que no se encuentra en el nivel de normalización deseado. Para ello se agrupan los atributos equivalentes y las dependencias con el mismo implicante. Se elige uno de estos grupos, teniendo en

cuenta que los implicados de las dependencias elegidas para la descomposición se pueden perder, y por lo tanto pueden perderse dependencias funcionales.

Se agrupan todos los atributos equivalentes de R (*CA, NA, DT, CT*) creando una nueva relación (R_1) con ellos. Se crea al mismo tiempo una nueva relación temporal (R_1') que se sigue descomponiendo hasta llegar a la forma normal deseada. Gráficamente estas dos relaciones se muestran en la figura 3.31.

R_1

AT = {CA, NA, CT, DT}
DEP = {CA→NA, NA→CA,
DT→CT, CT→DT,
CA→DT, DT→CA}

R_1'

AT = {CA,NA,CT,DT,I,O,FI,FF,P,
E,D,HO,HD}
DEP = {E→CT; D→CA; I→O;
I,FI→CA,P,FF; E,I→HO;
I,D,E→HD }

Figura 3.31. Descomposición de R

Las claves candidatas de la relación R_1 son {*CA*}, {*NA*}, {*DT*} y {*CT*}. Además, se sabe que R_1 se encuentra en 5FN, ya que se encuentra en 3FN por no tener atributos no principales, en FNBC, ya que todos los determinantes son clave, y en 4FN y 5FN al no poseer dependencias multivaluadas ni en combinación.

Por otro lado, en la relación R_1' se han agrupado las dependencias que tenían el mismo determinante {*I, FI*} y se han omitido todos los atributos que no participan en ninguna de las dependencias restantes. Se encuentra, por la misma razón que la relación R, en 1FN. Igualmente, la clave candidata de R se sigue manteniendo en R_1' por el mismo motivo.

Paso 3: Descomposición de R_1'

En este paso se elige otra de las dependencias de R_1' cuyo determinado no se encuentre en ninguna otra, por ejemplo, la dependencia {*I,D,E → HD*}. Se crea una nueva relación R_2 con esta dependencia y sus atributos y otra relación temporal R_2' con el resto de las dependencias de R_1' y los atributos restantes (se elimina el implicado *HD*). Gráficamente se muestra esta descomposición en la figura 3.32.

La relación R_2 se encuentra en 5FN y su clave es el conjunto {*D,E,I*}, mientras que la relación R_2' sigue estando en 1FN y su clave es igual a la de R y R_1' siguiendo el mismo razonamiento.

R_2

AT = {I,D,E,HD}
DEP = {I,D,E→HD}

R_2'

AT = {CA,NA,CT,DT,I,O,FI,FF,P,
E,D,HO}
DEP = {E→CT; D→CA; I→O;
I,FI→CA,P,FF; E,I→HO}

Figura 3.32. Descomposición de R_1'

Paso 4: Descomposición de R_2'

Se elige una nueva dependencia, donde el implicado no participa en ninguna otra dependencia de R_2', por ejemplo, la dependencia {*E,I → HO*}, creándose las relaciones que se muestran en la figura 3.33.

R_3

AT = {I,E,HO}
DEP = {I,E→HO}

R_3'

AT = {CA,NA,CT,DT,I,O, FI,
FF,P,E,D}
DEP = {E→CT; D→CA; I→O;
I,FI→CA,P,FF }

Figura 3.33. Descomposición de R_2'

La clave de la relación R_3 es el conjunto {*I,E*}, y por lo tanto se encuentra en 5F por el mismo razonamiento que el apartado anterior. Por otro lado, la relación R_3' se encuentra en 1FN, y sigue manteniendo las claves candidatas de R, R_1' y R_2'. Por lo tanto, se continúa descomponiendo en otras con mayor grado de normalización.

Paso 4: Descomposición de R_3'

Otra dependencia cuyo implicado no se encuentra en ninguna otra dependencia de R_3' es {*E→CT*}. Se opta por la elección de ésta y se crean las relaciones que se muestran en la figura 3.34.

En este caso, la relación R_4 se encuentra en 5FN y su única clave candidata es {*E*}, mientras que la relación R_4', cuya clave candidata es el conjunto {*I,FI,D*}, se encuentra aún en 1FN por la misma razón que R, R_1', R_2' y R_3', por lo tanto, se continúa su descomposición.

R_4 AT = {E, CT}
DEP = {E→CT}

R_4' AT = {CA,NA,DT,I,O,FI, FF,P,E,D}
DEP = {D→CA; I→O; I,FI→CA,P,FF }

Figura 3.34. Descomposición de R_3'

Paso 5: Descomposición de R_4'

La siguiente dependencia que se elige para la descomposición es $\{D \rightarrow CA\}$, donde su implicado no pertenece a ninguna otra dependencia. Se crean las dos relaciones que se muestran en la figura 3.35, donde R_5 se encuentra en 5FN y su clave es $\{D\}$ y la relación R_5' se encuentra en 1FN, al igual que las anteriores relaciones temporales.

R_5 AT = {D, CA}
DEP = {D→CA}

R_5' AT = {NA,DT,I,O,FI, FF,P,E,D}
DEP = {I→O; I,FI→CA,P,FF }

Figura 3.35. Descomposición de R_4'

Paso 6: Descomposición de R_5'

Se elige otra de las dependencias funcionales restantes, por ejemplo $\{I \rightarrow O\}$ y se crean dos nuevas dependencias, R_6 y R_7, que se muestran gráficamente en la figura 3.36. La relación R_6 se encuentra en 5FN y su clave candidata es $\{I\}$.

R_6 AT = {I,O}
DEP = {I→O}

R_7 AT = {NA, DT, I, FI, F, P, E, D}
DEP = {I,FI→NA,P,FF }

Figura 3.36. Descomposición de R_5'

La relación R_7 se encuentra igualmente en 5FN, ya que todo atributo no principal se encuentra directamente determinado por la clave (3FN), todo determinante es clave candidata (FNBC) y no existen atributos multivaluados ni en combinación. Su clave candidata es el conjunto de atributos $\{I,FI\}$.

Para finalizar, en las figuras 3.37 y 3.38 se muestra el árbol de descomposición, indicando en cada relación cuál es su clave y en qué forma normal se encuentra.

Se observa que en la descomposición de la relación universal R no se ha perdido ninguna dependencia funcional y que todas las relaciones se encuentran en 5FN, por lo que no se pueden descomponer más.

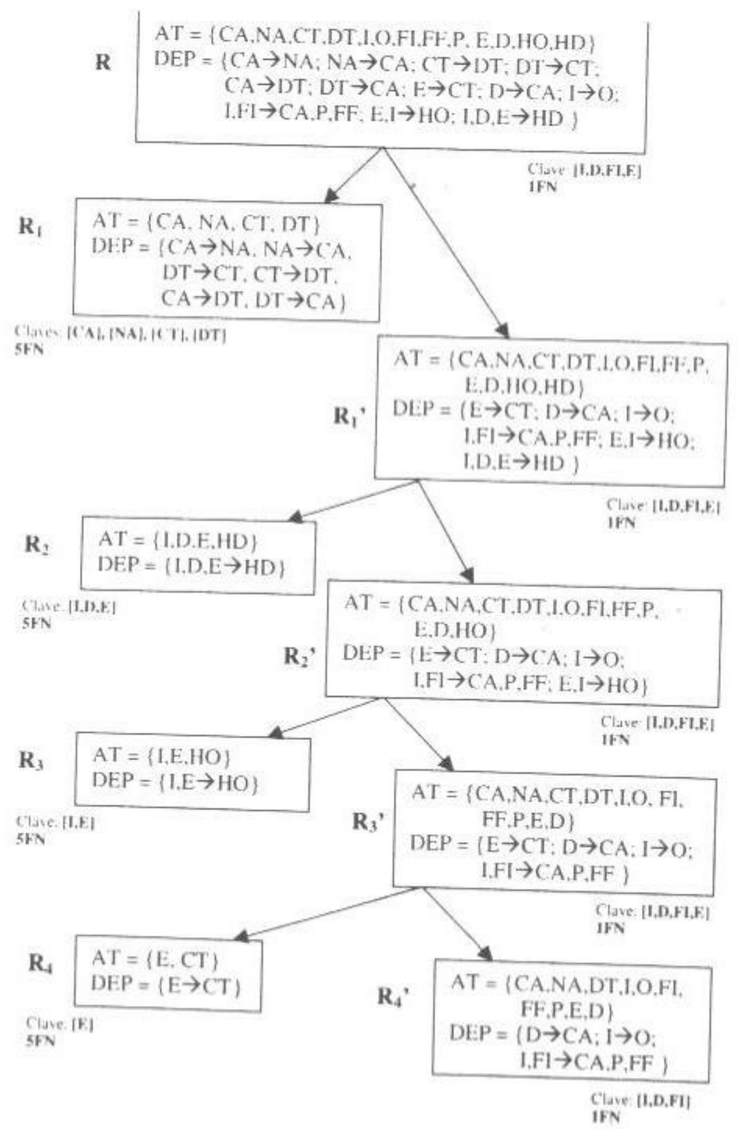


*Figura 3.37. Árbol de descomposición **hasta R_4'***

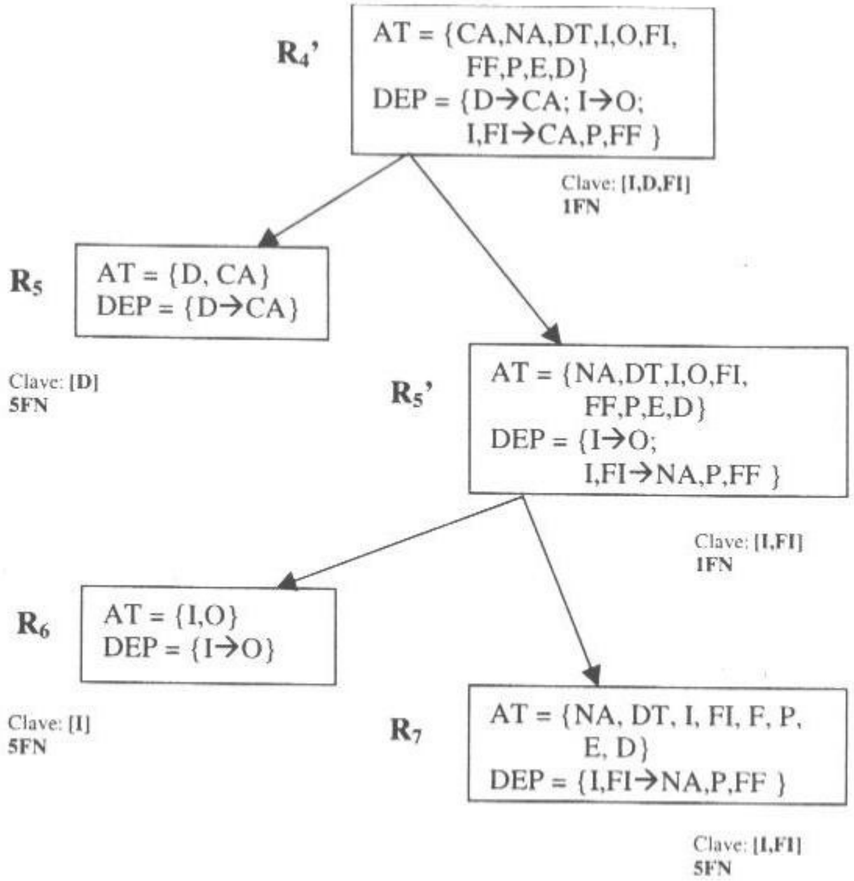


Figura 3.38. Árbol de descomposición ***desde*** R_4'

PROBLEMA 10

Enunciado

Se desea diseñar una base de datos relativa a los seminarios que se imparten en una determinada Universidad. El usuario nos da los siguientes supuestos semánticos:

a) Todo Seminario (**S**) es de un único Tipo (**T**)

b) Cada Día (**D**) sólo se puede impartir un Seminario de cada Tipo

c) Un Estudiante (**E**) permanece en la misma Aula (**A**) durante todo el Día

d) A un Estudiante se le da una única Nota (**N**) como resultado de un Seminario

e) Cada Día en un Aula sólo se puede impartir un Seminario

f) Un Seminario no puede tener dos Profesores (**P**) el mismo Día

g) Cada Tipo de seminario está siempre asociado al mismo conjunto de Libros de referencia (**L**) con independencia del resto de los atributos

h) Cada Seminario se imparte siempre para el mismo conjunto de valores de la pareja Día y Aula con independencia del resto de los atributos

Se pide:

1) Determinar las dependencias entre atributos.

2) Hallar un recubrimiento minimal válido.

3) Forma normal en la que se encuentra la relación, indicando qué dependencias impiden una forma normal más avanzada.

4) Esquema relacional en quinta forma normal y esquema definitivo con las observaciones que estime pertinentes.

Discusión del enunciado

DEPENDENCIAS ENTRE ATRIBUTOS

El análisis de los supuestos semánticos del enunciado se detallan a continuación:

*Todo Seminario (**S**) es de un único Tipo (**T**).*

Luego, conociendo el valor del atributo S se puede saber el valor del atributo T, y por lo tanto, S determina funcionalmente a T ($S \rightarrow T$):

- $S \rightarrow T$

*Cada Día (**D**) sólo se puede impartir un Seminario de cada Tipo.*

Esto indica que para un día concreto y un tipo concreto de seminario se sabe qué seminario se ha impartido, y por lo tanto se da la dependencia funcional siguiente:

- $D,T \rightarrow S$

*Un Estudiante (**E**) permanece en la misma Aula (**A**) durante todo el Día.*

Se sabe que un determinado estudiante en un día concreto siempre se encuentra en la misma aula, y por lo tanto los atributos D y E implican el aula:

- $D,E \rightarrow A$

*A un Estudiante se le da una única Nota (**N**) como resultado de un Seminario.*

También se sabe que un estudiante en un determinado seminario sólo puede tener una calificación, y por lo tanto, se da la siguiente dependencia funcional:

- $S,E \rightarrow N$

*Cada Día en un Aula (**A**) sólo se puede impartir un Seminario.*

Por otro lado se sabe que un día concreto en un aula sólo se puede impartir un seminario, y por lo tanto:

- $D,A \rightarrow S$

*Un Seminario no puede tener dos Profesores (**P**) el mismo Día.*

Este supuesto semántico indica que un seminario un día sólo puede tener un profesor, y por lo tanto lo determinan funcionalmente:

- $S,D \rightarrow P$

*Cada Tipo de seminario está siempre asociado al mismo conjunto de Libros de referencia (**L**) con independencia del resto de los atributos.*

Por otro lado se tiene que un tipo de seminario se encuentra asociado a un conjunto bien definido de libros de referencia, con independencia del resto de los atributos. Por lo tanto, en este caso se da una dependencia multivaluada del atributo *L* con respecto al atributo *T*:

- $T \rightarrow\rightarrow L$

Cada Seminario se imparte siempre para el mismo conjunto de valores de la pareja Día y Aula con independencia del resto de los atributos.

Por último se indica que cada seminario siempre se imparte en el mismo conjunto de valores (bien conocidos) para los atributos *D* y *A* con independencia del resto de los atributos, y por lo tanto se da la siguiente dependencia multivaluada:

- $S \rightarrow\rightarrow D, A$

Para finalizar este apartado, se resumen todos los cálculos anteriores mediante la relación universal, R, de la base de datos de la universidad. R está compuesta por un conjunto de atributos *AT* y un conjunto de dependencias que se detallan a continuación:

R (AT, DEP), donde:

AT = {*S, T, D, E, A, N, P, L*}

DEP = {

(15) $S \rightarrow T$

(16) $T, D \rightarrow S$

(17) $E, D \rightarrow A$

(18) $E, S \rightarrow N$

(19) $D, A \rightarrow S$

(20) $D, S \rightarrow P$

(21) $T \rightarrow\rightarrow L$

(22) $S \rightarrow\rightarrow D, A$

}

RECUBRIMIENTO IRREDUNDANTE

Para que un conjunto de dependencias sea irredundante o minimal han de cumplir tres características:

- *Dependencias elementales:* Todas las dependencias del DEP son elementales, ya que son plenas, no son triviales y tienen un único atributo implicado.
- *No existen atributos extraños:* En ningunas de las dependencias existen atributos extraños. No se realiza la comprobación detallada para ninguna de ellas, ya que el cálculo es similar al realizado en problemas anteriores.
- *No existen dependencias redundantes:* Ninguna de las dependencias de DEP es redundante, ya que no se pueden deducir a partir de las demás. Tampoco se detalla el cálculo de las dependencias redundantes.

Por lo tanto, se concluye que el conjunto de dependencias mínimo o irredundante de R es DEP, y por lo tanto:

$DEP^m = DEP$

FORMA NORMAL DE LA RELACIÓN

El primer paso para hallar la forma normal de la relación es hallar sus claves candidatas, sus atributos principales y no principales, y por último la forma normal en la que se encuentra.

Claves de la relación y atributos principales y no principales

El cálculo de la forma normal de una relación se basa en los atributos principales y no principales de la misma, por lo que, el primer paso es realizar el cálculo de la clave de la relación. Este cálculo se basa en el cierre transitivo de sus descriptores, y se detalla a continuación.

El cierre transitivo de *S* es el conjunto formado por {*S, T*}, mientras que el resto de los atributos de R sólo se determinan funcionalmente a sí mismos de forma trivial.

Por otro lado los atributos multivaluados *L* y *D* no son implicados por ningún otro, por lo que formarán parte de todas las claves candidatas. El cierre transitivo de estos dos elementos es {L, D}$^+$= {*L, D*}. Por lo tanto no implica a todos los atributos de R (AT).

Si, por ejemplo, se añadiera el atributo *E* a este conjunto, su cierre transitivo sería el siguiente:

{*E, D, L*}$^+$= AT

- (3) *E, D* → *A*, luego {*E, D, L*}$^+$= {*E, D, L, A*}
- (5) *D, A* → *S*, luego {*E, D, L*}$^+$= {*E, D, L, A, S*}
- (1) *S* → *T*, luego {*E, D, L*}$^+$= {*E, D, L, A, S, T*}
- (4) *E, S* → *N*, luego {*E, D, L*}$^+$= {*E, D, L, A, S, T, N*}
- (1) *S, D* → *P*, luego {*E, D, L*}$^+$= {*E, D, L, A, S, T, N, P*}

Por último se concluye, aunque no se explica de forma detallada, dejándose como ejercicio al lector, que el conjunto {*E, D, L*} implica a todos los atributos de R (AT), que no existe ningún subconjunto que implica a todo AT, y que no existen más claves candidatas. Resumiendo:

- Claves candidatas: {*E, D, L*}
- Atributos principales: {*E, D, L*}
- Atributos no principales: {*A, S, T, N, P*}

El nivel de normalización de la relación R es 2FN, ya que todos los atributos no principales dependen plenamente de la clave. No se encuentra en 3FN, ya que existen atributos no principales que dependen transitivamente de alguna clave. Por ejemplo, el atributo *S* depende del conjunto {*D, A*}.

NORMALIZACIÓN MEDIANTE ANÁLISIS

Se aplica el método de análisis o descomposición para dividir la relación inicial, R, en varias con un nivel de normalización mayor.

Paso 1: Recubrimiento minimal y claves

El primer paso del método es hallar el recubrimiento minimal de la relación y la forma normal en la que se encuentra (por si no es necesaria la descomposición). Estos puntos ya se han resuelto en apartados anteriores.

Paso 2: Descomposición de R

El siguiente paso es realizar la descomposición de la relación inicial, R. Para ello se agrupan los atributos equivalentes y las dependencias con el mismo implicante. En este caso no se realiza ninguna agrupación.

Se elige una de las dependencias funcionales cuyos implicados no participan en ninguna otra dependencia de DEP^m, por ejemplo, la dependencia *E,S*→*N*. Se crean dos nuevas relaciones, R_1, con esta dependencia y los atributos de la misma, y R_1', con carácter temporal, con todos los atributos de R menos lo no principales de R_1 y el resto de las dependencias. Gráficamente estas dos relaciones se muestran en la figura 3.39.

R_1

AT = {E,S,N}
DEP = {E,S → N}

R_1'

AT = {E,S,A,D,P,T,L}
DEP = {S→T; T,D→S; E,D→A;
D,A→S; S,D→P; T→→L;
S→→D,A }

Figura 3.39. Descomposición de R

La clave candidata de la relación R_1 es el conjunto {*E,S*}. Además, se sabe que R_1 se encuentra en 5FN, ya que se encuentra en 3FN debido a que todo atributo no principal se encuentra determinado directamente por la clave, en FNBC, ya que todos los determinantes son clave, y en 4FN y 5FN al no poseer dependencias multivaluadas ni en combinación.

Por otro lado, la relación R_1' se encuentra en 2FN y tiene la misma clave que R por las mismas razones.

Paso 3: Descomposición de R_1'

En este paso se elige otra de las dependencias de R_1' cuyo determinado no se encuentra en ninguna otra, por ejemplo, la dependencia {*S,D*→*P*}. Se crea una nueva relación R_2 con esta dependencia y sus atributos y otra relación temporal R_2' con el

resto de las dependencias de R_1' y los atributos restantes. Gráficamente se muestra esta descomposición en la figura 3.40.

R_2

AT = {S,D,P}
DEP = {D,S → P}

R_2'

AT = {E,S,A,D,T,L}
DEP = {S→T; T,D→S; E,D→A;
D,A→S; T→→L;
S→→D,A }

Figura 3.40. Descomposición de R_1'

La relación R_2 se encuentra en 5FN y su clave es el conjunto $\{D,S\}$, mientras que la relación R_2' sigue estando en 2FN y su clave es igual a la de R y R_1' siguiendo el mismo razonamiento.

Paso 4: Descomposición de R_2'

A continuación se elige otra dependencia de R_2' cuyo determinante no participe en ninguna otra dependencia de la relación. En esta ocasión no es tan fácil la elección de la dependencia para crear una nueva relación. Esto se debe a que no existe ninguna dependencia funcional cuyo implicado no pertenezca a ninguna otra:

- $S \rightarrow T$, donde T participa en la dependencia $T,D \rightarrow S$
- $T,D \rightarrow S$ y $D,A \rightarrow S$ donde S participa en $S \rightarrow T$, $D,A \rightarrow S$ y $S \rightarrow\rightarrow D,A$.
- $E,D \rightarrow A$, donde A participa en $D,A \rightarrow S$ y $S \rightarrow\rightarrow D,A$.

De esta forma, si se eligiera cualquiera de las dependencias anteriores para la descomposición de R_2' se producirían pérdidas de dependencias.

Por lo tanto, suponiendo que no se desea en ningún momento que existan pérdidas, se procede a la descomposición de la relación a partir de sus dependencias multivaluadas.

Se comienza la descomposición mediante la elección de la dependencia $T \rightarrow\rightarrow L$, se crea una nueva relación con esta dependencia y sus atributos (R_3) y una relación adicional con carácter temporal para continuar la descomposición (R_3'). Gráficamente esta descomposición se muestra en la figura 3.41.

La clave de la relación R_3 es el conjunto $\{T,L\}$ y se encuentra en 5FN, ya que se encuentra en FNBC y todos los atributos de R_3 forman parte de la clave.

R_3

AT = {T, L}
DEP = {T →→ L}

R_3'

AT = {E,S,A,D,T}
DEP = {S→T; T,D→S; E,D→A;
D,A→S; S→→D,A }

Figura 3.41. Descomposición de R_2'

Por otro lado, la relación R_3', cuya clave candidata es el conjunto formado por los atributos $\{E, D\}$, se encuentra en 2FN, pero no en 3FN, ya que existen dependencias transitivas. Por ejemplo, el atributo no principal S está determinado por el conjunto $\{T,D\}$, no directamente por la clave.

Paso 5: Descomposición de R_3'

A continuación se continúa la descomposición de las dependencias multivaluadas. En este caso, se divide la dependencia $S \rightarrow\rightarrow D,A$ en dos dependencias ($S \rightarrow\rightarrow D$ y $S \rightarrow\rightarrow A$) sin perder el significado de la dependencia.

Se elige cualquiera de las dos dependencias multivaluadas nuevas, por ejemplo $S \rightarrow\rightarrow D$ y se crea una nueva relación (R_4). Se crea otra relación temporal con el resto de las dependencias (R_4'). Gráficamente se muestra en la figura 3.42.

R_4

AT = {S,D}
DEP = {S →→ D}

R_4'

AT = {E,S,A,D,T}
DEP = {S→T; T,D→S; E,D→A;
D,A→S; S→→A}

Figura 3.42. Descomposición de R_3'

En este caso, la relación R_4 se encuentra en 5FN por el mismo motivo que R_3, siendo su clave candidata el conjunto $\{S,D\}$.

Por otro lado, la relación R_4', al igual que ocurría con la relación R_3', se encuentra en 2FN.

Paso 6: Descomposición de R_4'

Para terminar con las dependencias multivaluadas, se descompone la relación R_4' en dos relaciones: la relación R_5, escogiendo la dependencia multivaluada de R_4' ($S \rightarrow\rightarrow A$), y la relación temporal R_5'. Ambas relaciones se pueden ver gráficamente en la figura 3.43.

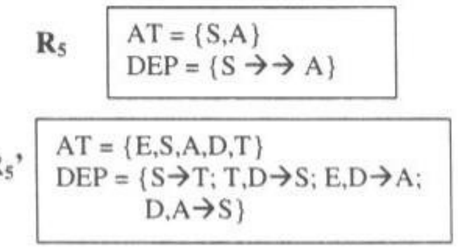


Figura 3.43. Descomposición de R_4'

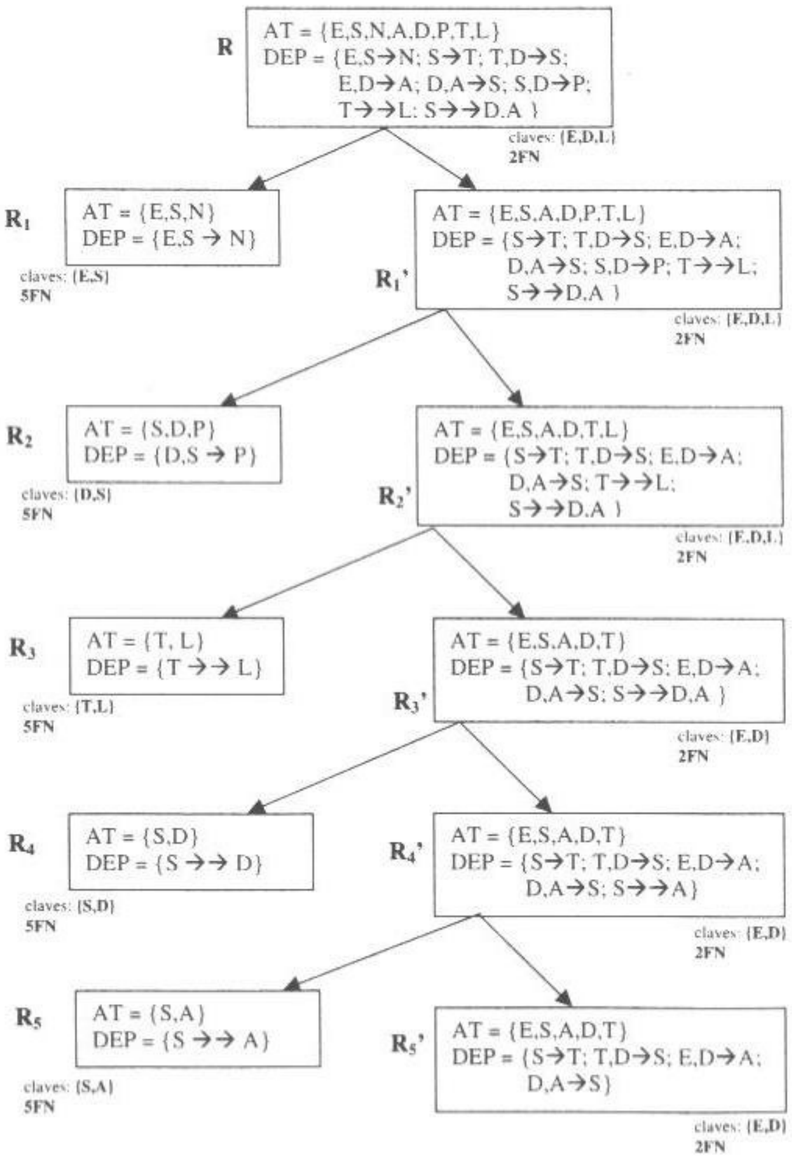


Figura 3.44. Árbol de descomposición sin pérdidas de dependencias

La clave de R_5 es el conjunto *{S,A}* y, al igual que ocurre con R_3 y R_4, se encuentra en 5FN.

Por otro lado, la relación R_5', al igual que ocurría con las anteriores relaciones temporales, sigue estando en 2FN.

Llegados a este punto, la relación R_5' no se podría descomponer más debido a que no existe ninguna dependencia funcional cuyo implicado no participe en otra dependencia de R_5'. Si fuera requisito indispensable el mantenimiento de las dependencias, no se podría descomponer más. De esta forma, el árbol de descomposición sin pérdidas de dependencias funcionales se muestra en la figura 3.44.

Paso 7: Descomposición de R_5'

En el caso de que no importe que se pierdan dependencias funcionales, la relación R_5' se puede seguir descomponiendo en otras relaciones con un nivel de normalización mayor.

Para ello se realizan las descomposiciones que aparecen en el árbol de descomposición de la figura 3.45, donde se ha perdido una dependencia funcional, en concreto la dependencia *{T,D→S}* al realizar la descomposición de R_5' eligiendo la dependencia *{S→T}*.

Se concluye que en este caso, todas las relaciones obtenidas se encuentran en 5FN, aunque se pierde una dependencia funcional, *T,D→S*.

En el caso de que se hubiera elegido cualquier otra dependencia para la descomposición, se hubiera perdido alguna otra dependencia. Se deja como ejercicio al lector su comprobación.

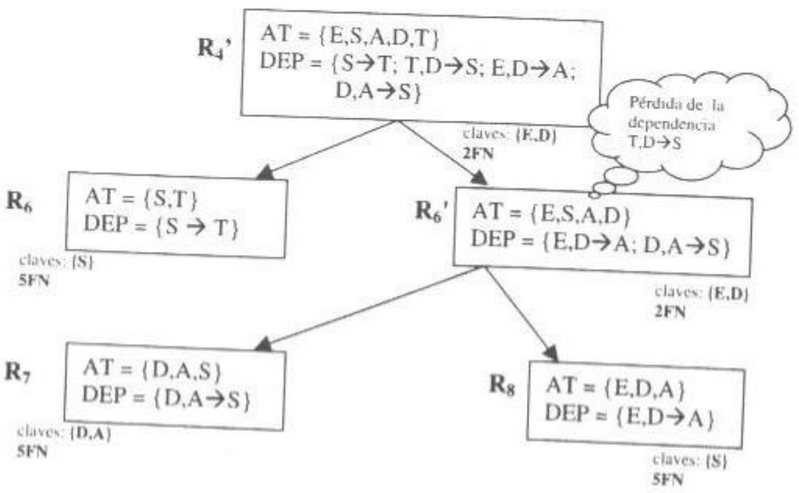


Figura 3.45. Árbol de descomposición parcial con pérdidas de dependencias

CAPÍTULO 4

DISEÑO DE BASES DE DATOS RELACIONALES DISTRIBUIDAS

BASES DE DATOS DISTRIBUIDAS

Introducción

Las propiedades y características de las Bases de Datos Distribuidas (BDD) requieren la adaptación de los diseños de las BD centralizadas a las mismas y éstos, como veremos, pueden abordarse de diferentes maneras. Para entender el diseño adaptado a una BDD debemos comprender su definición, propiedades que las caracterizan y arquitecturas que poseen. Posteriormente analizaremos las metodologías que existen para el diseño de BDD.

Una Base de Datos Distribuida es una colección de varias Bases de Datos interrelacionadas lógicamente y distribuidas sobre una red de ordenadores.

Son dos, pues, los rasgos que caracterizan una BDD. En primer lugar, la *integración* de los datos, es decir, los datos no son una colección de ficheros que están almacenados individualmente en los nodos de una red, sino que forman parte de una estructura global que los relaciona lógicamente. Y en segundo lugar, la *distribución* de

s datos en distintos nodos de una red que para los usuarios de la BDD aparecerán omo datos homogéneos y únicos.

En resumen podemos caracterizar las BDD de la siguiente forma:

Los datos deben estar físicamente en más de un ordenador, es decir, los datos se encuentran almacenados en distintas sedes.

Las sedes deben estar interconectadas mediante una red. Cada sede es un emplazamiento o nodo de la red. Para realizar el diseño no se tendrá en cuenta la topología, tipo y rendimiento de la red que une estas sedes, aunque estas propiedades son relevantes para obtener buenos rendimientos y poder evitar los atascos y retrasos en el tráfico de datos por la red. Consideramos que es un aspecto muy concreto y específico de cada red y que cada caso particular necesita su propio estudio.

Los datos han de estar lógicamente integrados en una única estructura lógica o esquema global común.

Los usuarios han de tener acceso –recuperación y actualización– a los datos pertenecientes a la BDD, ya se encuentren éstos en la misma sede (acceso local) o en otra (acceso remoto).

- Cada emplazamiento o nodo proporciona un entorno para la ejecución de transacciones tanto locales como globales.
- En una única operación, tanto de recuperación como de actualización, se puede acceder a datos que se encuentran en más de una sede sin que el usuario sepa la distribución de los mismos en las distintas sedes (propiedad de transparencia de red).

El Sistema de Gestión de la BDD (SGBDD) es un producto software que permite el manejo de bases de datos y hace la distribución transparente a los usuarios, y cumple con respecto a las BDD el mismo papel que cumple el Sistema Gestor de Base de Datos (SGBD) respecto a las BD centralizadas. Se compone principalmente de cuatro componentes:

- **Procesador de datos locales** que se encarga de la gestión local de los datos de forma parecida al software de un SGBD centralizado, por lo que además de ejecutar transacciones locales se encarga de la concurrencia y la recuperación ante fallos a nivel local.
- **Diccionario o directorio global** donde se guardará información acerca de dónde y cómo se almacenan los datos, el modo de acceso y otras características físicas. En resumen, contiene las especificaciones necesarias para pasar de la representación externa o esquema externo de los datos a la representación interna de los mismos.

- **Procesador de aplicaciones distribuidas** que es el responsable de las funciones distribuidas. Accede a la información sobre la ubicación de los datos, que se encuentra en el diccionario, y se ocupa de procesar todas las peticiones que involucran más de una sede para generar un plan de ejecución distribuido. Es el elemento diferenciador en los sistemas distribuidos, dada una operación se encargará de repartir el trabajo a los distintos procesadores locales que intervienen en dicha operación.
- **Software y red de comunicaciones.** No forma parte estrictamente del SGBDD sino que provee al procesador de aplicaciones distribuidas de primitivas y servicios de comunicaciones para que éste lleve a cabo su labor.

Clases y arquitecturas de los SGBDD

Dependiendo de la homogeneidad de los SGBD locales, de la distribución de los datos y de su autonomía tendremos distintos tipos de SGBDD.

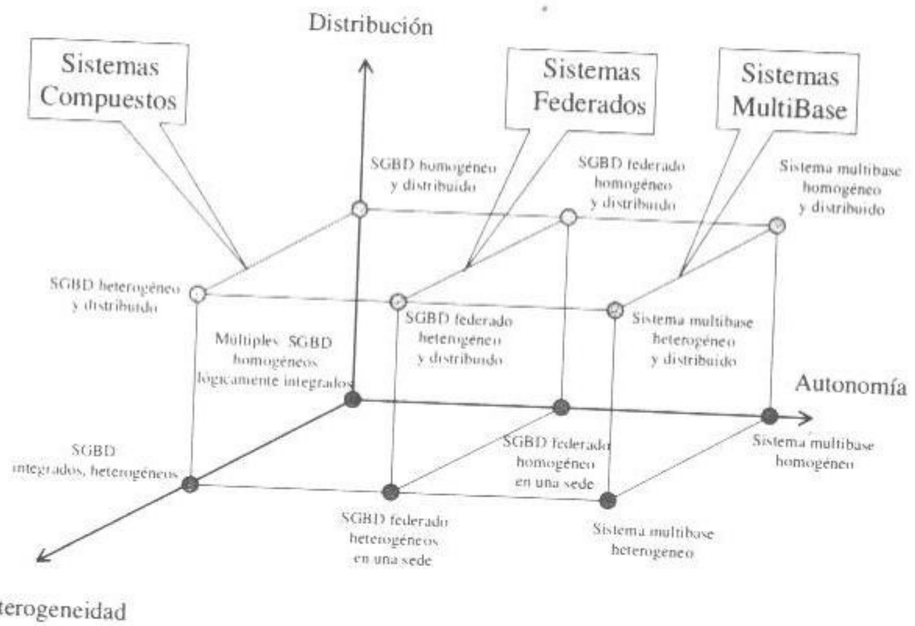


Figura 4.1. Clases de SGBDD

Empecemos por la *homogeneidad*. Si todos los SGBD son iguales tenemos un **SGBDD homogéneo** donde se tiene un único producto, y lenguaje de consultas por lo que suelen ser sistemas muy integrados. Si los SGBD son distintos tenemos **SGBDD heterogéneos** donde, aunque puedan utilizar el mismo modelo de datos, se tienen distintos productos y lenguajes de consultas que requieren su integración.

La *distribución* determina si los datos están distribuidos físicamente sobre múltiples sitios que se comunican entre sí o si se mantienen en un único lugar, datos centralizados.

Y por último la *autonomía*, que se refiere a la distribución del control. Se tendrán **stemas Estrechamente Integrados** (Sistemas compuestos) si todo el acceso a los atos se realiza siempre a través del procesador de datos distribuidos con las sedes ocales totalmente dependientes de éste, él gestiona todos los accesos y funciones de dministración. Los **Sistemas Semiautonomos** (Sistemas Federados) sin embargo, ejan que cada procesador local actúe de forma autónoma e independiente, de hecho, ada uno de ellos tiene sus usuarios, administradores, transacciones y aplicaciones xclusivamente locales, además de poseer porciones específicas de la BDD. Por ltimo, si los procesadores locales no conocen la existencia de otros SGBD ni cómo omunicarse con ellos tendremos **Sistemas con total autonomía** (Sistemas Multi-ase de Datos) donde al no haber un control global el acceso a los datos de distintos odos es especialmente complicado.

Podemos resumir las distintas clases de SGBDD representando cada uno de los actores estudiados anteriormente en un eje de coordenadas, de esta manera podremos bservar que distribución, autonomía y heterogeneidad son aspectos ortogonales.

La arquitectura de estos sistemas puede resumirse en dos tipos: la arquitectura ANSI/X3/SPARC para SGBD distribuidos y la arquitectura de los SGBD Multibase.

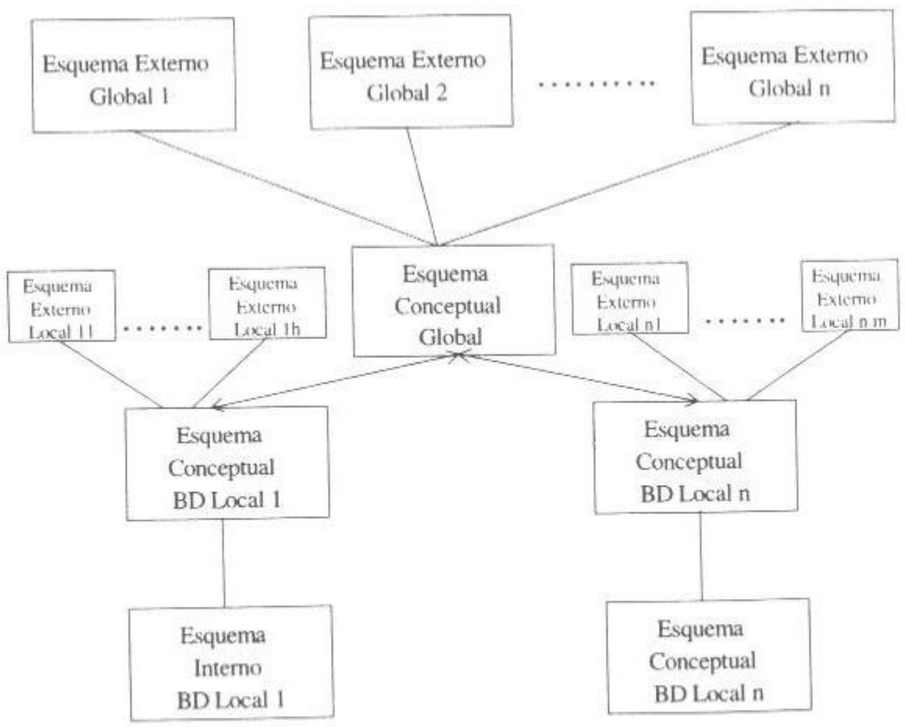


Figura 4.2. Arquitectura ANSI/X3/SPARC para SGBDD

La arquitectura ANSI/XE/SPARC (figura 4.2.) se corresponde con los sistemas federados, es una extensión de la dada para los sistemas centralizados donde si se omiten los esquemas externos locales tendríamos un sistema integrado.

En los sistemas Multibase no tenemos Esquema Conceptual Global, por lo que la autonomía local es completa.

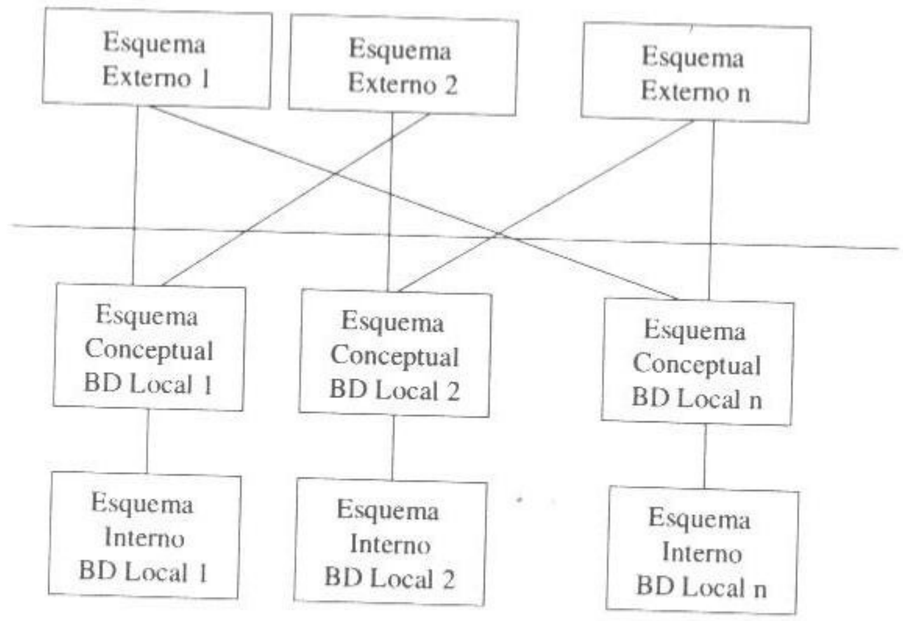


Figura 4.3. Arquitectura de los SGBD Multibase

Diseño de BDD

En cualquier proceso de diseño hay dos aproximaciones básicas: la ascendente o Top-Down y la descendente o Bottom-Up. Veamos estas aproximaciones en el caso del diseño de BDD.

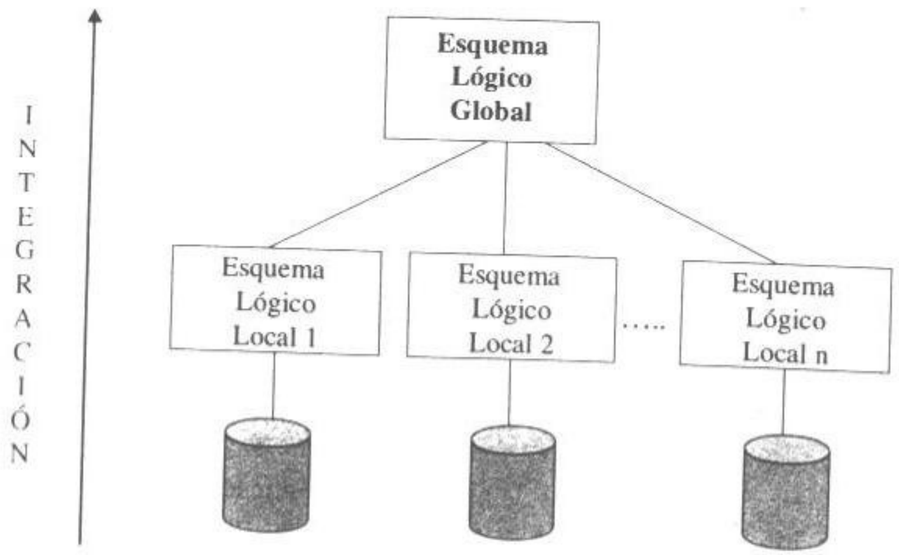


Figura 4.4. Metodología Ascendente de Diseño de BDD

La **metodología ascendente** se utiliza, en general, cuando existen varias BD Locales y queremos construir una BDD. Por lo que partimos de distintos esquemas lógicos locales (ELL) que se corresponden a Bases de Datos ubicadas en diferentes nodos de una red y se integran, parte de ellos o todo, en un único esquema lógico global (ELG).

La **metodología descendente** parte de un esquema lógico global (ELG) que mediante la fragmentación, replicación y asignación de los objetos pertenecientes a éste, construyen los esquemas lógicos locales.

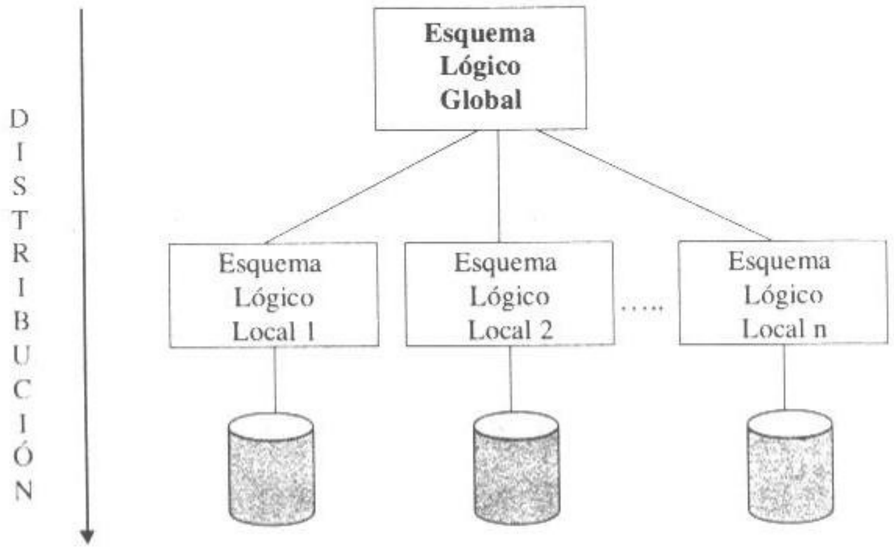


Figura 4.5. Metodología Descendente de Diseño de BDD

En la resolución de los problemas planteados en este capítulo, problemas que se corresponden con el diseño de BDD relacionales, vamos a utilizar una **metodología descendente**, es decir, partimos de un esquema lógico global (esquema o grafo relacional en nuestro caso) de la Base de Datos que queremos distribuir y a partir de éste construimos los esquemas de fragmentación, de asignación y replicación de los datos para distribuirlos en los diferentes nodos de la red.

En general, los esquemas de fragmentación se basan en el análisis de los datos utilizados por las distintas aplicaciones que acceden a la BD para crear relaciones más pequeñas y más adaptadas a las operaciones de recuperación y actualización que las aplicaciones utilizan, es decir, tener los datos divididos según la utilización que de ellos se hace.

Sin embargo, en los esquemas de asignación y replicación se fija desde qué nodo se demandan los datos y el tipo de operación que se realiza (si es de consulta o actualización), para que estas operaciones se puedan llevar a cabo de forma local y minimizar de esta forma el tráfico por la red que las ralentiza.

La replicación o duplicación se puede realizar cuando desde distintos nodos se requiere la misma información, si además las operaciones son de consulta no existe ningún problema en duplicar los datos. Ahora bien, si lo que hacemos es actualizarlos el SGBD debe asegurar que todas las copias son consistentes, es decir, que todas mantienen la misma información, por ejemplo, con mecanismos de propagación que actualicen todas las copias de los datos modificados, con el alto coste que esto supone. Por tanto, debemos llegar a una solución de compromiso y analizar ventajas y desventajas de replicar los datos.

Esquema de fragmentación

El esquema de fragmentación se compone de las distintas relaciones en las que hemos divido una relación proveniente del Esquema Lógico Global junto con la condición empleada para esta división, expresado todo ello en álgebra relacional.

Para asegurar que la BD no sufrirá cambios semánticos durante la fragmentación de los datos, se definen las siguientes tres normas que determinan la calidad de la fragmentación de una relación:

Completitud. Todos los datos han de encontrarse en, al menos, un fragmento.

Disyunción. Los fragmentos deberán ser disjuntos.

Reconstrucción. Siempre se ha de poder reconstruir la BD global a partir de los fragmentos.

Existen distintas formas de fragmentar una relación, con fragmentación horizontal, vertical o mixta.

Fragmentación vertical

La fragmentación vertical divide la relación R en conjuntos de columnas, así cada fragmento mantiene ciertos atributos de la relación original. La fragmentación vertical se realiza mediante el operador algebraico de proyección y su notación es la siguiente:

$R_i = \Pi_{L_i}(R)$ donde i = 1..n y R_i es el conjunto de fragmentos en que se divide la relación original R

L_i es la condición por la que fragmentamos, en este caso al ser una proyección se trata de un subconjunto de atributos de R. Para que se cumplan las condiciones anteriormente descritas sobre una correcta fragmentación los subconjuntos de atributos L_i han de cumplir:

- La unión de todos los L_i nos da todos los atributos de R.

- La intersección de todos los Li nos da la clave primaria de la relación R. Esto supone que todos los Li tienen en común la clave primaria para de esta forma poder reconstruir a partir de los fragmentos, R_i, la relación inicial, R, mediante la operación de combinación.

Veamos con un ejemplo la aplicación de este tipo de fragmentación. Supongamos que tenemos una base de datos centralizada de unos grandes almacenes y que una de sus relaciones es:

EMPLEADOS.(apellidos, nombre, dirección, teléfono, función, localización, extensión)

Se quieren distribuir los datos de EMPLEADOS teniendo en cuenta que se tienen dos sedes o nodos enlazados por una red de área local y que en el NODO1 se encuentra el departamento de contabilidad y desde aquí se manejan los datos personales del empleado (nombre, apellidos, dirección, teléfono y función), mientras que en el NODO2 se encuentra centralita y en este nodo se consultan los datos correspondientes a la localización del empleado (localización, y extensión).

Esquema de fragmentación:

Fragmentaremos verticalmente la tabla EMPLEADOS creando dos nuevas relaciones:

EMPLEADO_CONTABILIDAD =

$\Pi_{\text{apellidos, nombre, dirección, teléfono, función}}$(EMPLEADOS)

EMPLEADO_CENTRALITA = $\Pi_{\text{apellidos, nombre, localización, extensión}}$ (EMPLEADOS)

La combinación de ambas relaciones proporcionará la relación original:

EMPLEADOS =

EMPLEADO_CONTABILIDAD $*_{\text{apellidos, nombre}}$ EMPLEADO_CENTRALITA

Fragmentación horizontal

Este tipo de fragmentación divide una relación en subconjuntos de tuplas, cada uno de ellos con un significado lógico.

Existen dos tipos de fragmentación horizontal: primaria o derivada.

- La fragmentación horizontal primaria se define como una selección de la relación R y su notación es la siguiente: $R_i = \sigma_{P_i}(R)$ donde P_i es un predicado sobre uno o más atributos de R y es la condición empleada para seleccionar el contenido de los fragmentos.

- La fragmentación horizontal derivada se realiza en función de predicados definidos sobre atributos de otras relaciones o fragmentos; esto se debe a que la relación R a fragmentar, depende de la relación Q, sobre cuyos atributos está definido el predicado de la fragmentación. Se define empleando el operador relacional de semicombinación, que actúa de la misma forma que el operador de combinación, pero la relación resultante sólo tiene los atributos del primer operando.

$R_i = R \ltimes Q_i$ donde Q_i corresponde al conjunto de fragmentos en los que se ha divido la relación Q. La semicombinación se realiza por el atributo o atributos que relacionan estas dos tablas.

Para que la fragmentación sea correcta se tiene que cumplir que la unión de todos los R_i sea la relación R y que su intersección sea vacía.

Veamos una aplicación de estas fragmentaciones.

Supongamos que al ejemplo anterior le añadimos la siguiente especificación:

Cada producto, comercializado por los grandes almacenes, se encuentra en un almacén. No hay productos que se encuentren en más de un almacén. Se sabe que los almacenes están divididos en áreas (norte, sur, este y oeste) y que además de los nodos de centralita y contabilidad hay otros cuatro nodos más, cada uno de ellos se encuentra en una de estas áreas, y en cada una de estas sedes sólo se maneja la información referente a los productos almacenados en dichas áreas.

Por lo que además de la tabla EMPLEADO tendremos las siguientes relaciones:

PRODUCTOS (cód_pród, nombre, descripción, *almacén*)

ALMACÉN (cód_almacén, dirección, teléfono, área)

Dado que en cada almacén sólo se requiere información de los productos almacenados en el área en que se encuentra, podemos realizar una fragmentación para repartir la información que se encuentra en la relación PRODUCTOS. Para ello haremos una fragmentación horizontal primaria en la relación ALMACÉN:

$\text{ALMACÉN}_i = \sigma_{\text{area} = i}$ (ALMACÉN) donde i = {norte, sur, este, oeste}

Cada uno de estos fragmentos contiene información de los almacenes ubicados en cada una de las áreas.

Si realizamos una fragmentación horizontal derivada en la relación PRODUCTOS:

$\text{PRODUCTOS}_i = \text{PRODUCTOS} \ltimes_{\text{almacen=cod_almacen}} \text{ALMACÉN}_i$

Conseguiremos la información de los productos almacenados en cada área.

Las relaciones originales se obtendrán mediante la unión de los fragmentos.

ALMACÉN= $ALMACÉN_{norte} \cup ALMACÉN_{sur} \cup ALMACÉN_{este} \cup ALMACÉN_{oeste}$

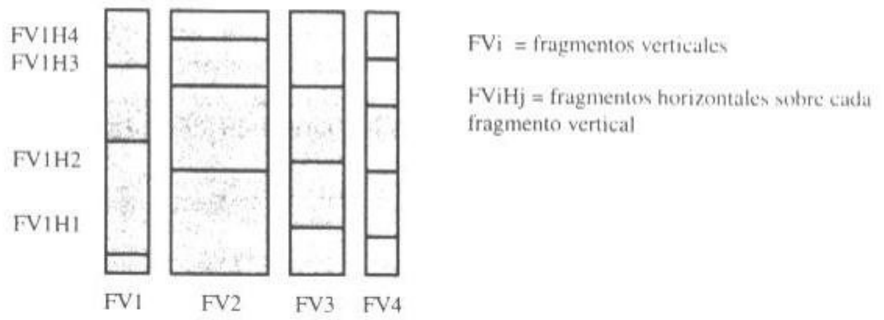
PRODUCTOS=

$PRODUCTOS_{norte} \cup PRODUCTOS_{sur} \cup PRODUCTOS_{este} \cup PRODUCTOS_{oeste}$

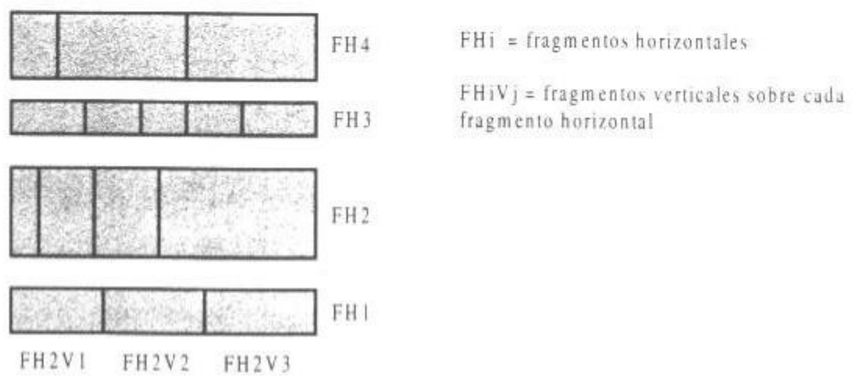
Fragmentación mixta

En algunos casos la fragmentación horizontal o vertical del esquema de la base de datos no es suficiente para satisfacer los requisitos de las aplicaciones de usuario, por lo que puede necesitarse volver a fragmentar los fragmentos obtenidos. Esto es lo que se denomina fragmentación mixta y se clasifica de la siguiente forma:

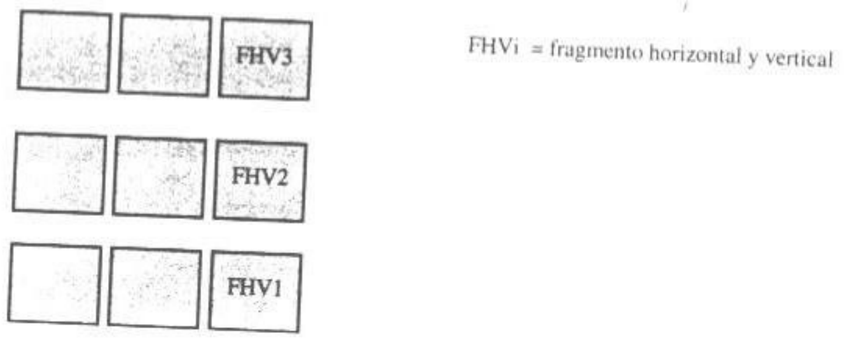
Fragmentación VH. Una fragmentación vertical seguida de una fragmentación horizontal, sobre cada uno de los fragmentos verticales.



Fragmentación HV. Una fragmentación horizontal seguida de una fragmentación vertical, sobre cada uno de los fragmentos horizontales.



Red de celdas. Aplicamos simultáneamente una fragmentación vertical y horizontal.



Esquema de asignación y replicación

El esquema de asignación y replicación consiste en la realización de la correspondencia entre los fragmentos y los nodos que constituyen la red de comunicaciones de la BDD. Dicha correspondencia debe hacerse de la manera más óptima, teniendo en cuenta los dos parámetros siguientes:

Mínimo coste: minimiza el coste de almacenamiento de cada fragmento en el nodo correspondiente, el coste de modificar un fragmento que está replicado en distintos nodos y el coste de la transferencia de datos por la red.

Rendimiento: minimiza los tiempos de respuesta y maximiza la capacidad de procesamiento del sistema en cada nodo.

Aunque existen algoritmos para realizar la asignación teniendo en cuenta estos parámetros, resultan complejos y no proporcionan la solución óptima sino simplemente una solución.

Por tanto, en la resolución de los problemas de este capítulo se ha tomado la determinación de realizar la asignación buscando siempre el procesamiento local de los datos y sólo utilizar replicación cuando la operación que se realice desde el nodo que solicita los datos sea siempre de consulta.

El esquema de asignación y replicación del ejemplo anterior quedaría de la siguiente forma:

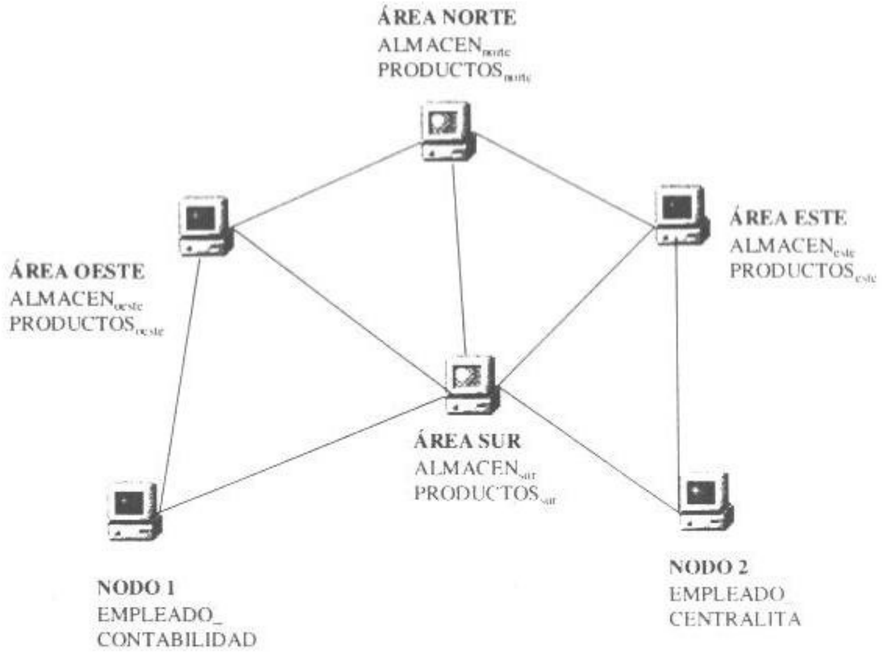


Figura 4.6. Esquema de Asignación y Replicación del Ejemplo

PROBLEMA 1: BDD SOCIEDADES MÉDICAS

Enunciado

Una de las sociedades médicas más importantes del país, con oficina central en Madrid, dispone de tres grandes centros médicos en esta misma ciudad. En cada uno de ellos se atienden determinadas especialidades médicas por el personal contratado por dicha sociedad, de manera que una especialidad puede darse en un centro o en varios y todos los centros tienen al menos una especialidad.

En la actualidad tienen una base de datos relacional centralizada, a la cual acceden todos los centros, con el esquema de la página siguiente.

Realice el diseño de los esquemas de fragmentación y asignación de una Base de Datos Distribuida para conseguir la mayor autonomía local, tanto en los centros médicos como en la sede central, sabiendo que las operaciones realizadas en cada uno de ellos son:

En la sede central (Cód_Centro = 01) se elaboran las nóminas del personal contratado por la Sociedad Médica y de aquí se envían a los Centros para que sean repartidas a los trabajadores.

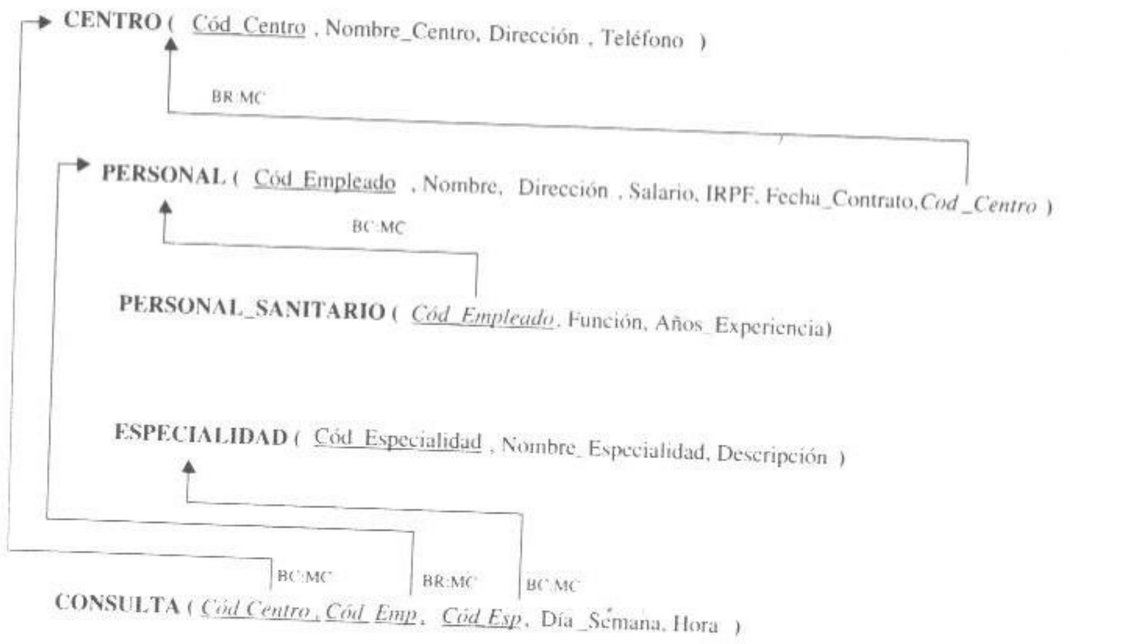


Figura 4.7. Esquema relacional correspondiente a una BD centralizada de Sociedades Médicas

Cada Centro médico (Cód_Centro = {02, 03, 04}) debe tener la relación de personal sanitario que trabaja en el mismo, así como el horario de la consulta y la especialidad que desarrolla.

Discusión del enunciado

Esquema de fragmentación

> *"En la sede central (Cód_Centro = 01) se elaboran las nóminas del personal contratado por la Sociedad Médica y de aquí se envían a los Centros para que sean repartidas a los trabajadores."*

Suponiendo que la información necesaria para realizar las nóminas son los atributos "Salario", "IRPF" y "Fecha_Contrato" se realizará una fragmentación vertical de la relación PERSONAL para separar esta información y posteriormente poder asignarla a la sede central. Las relaciones resultantes contendrán en cualquier caso la clave primaria de la relación de la que proceden.

INF_PERSONAL: $\Pi_{Cód_Empleado,\ Nombre,Dirección\ ,Cód_Centro}$ (PERSONAL)

NÓMINA_PERSONAL: $\Pi_{Cód_Empleado,\ Salario,\ IRPF,\ Fecha_Contrato}$ (PERSONAL)

El fragmento INF_PERSONAL nos interesará que se fragmente de nuevo para tener en cada centro la información del personal que está a su cargo. Para ello tendremos que aplicar una fragmentación horizontal:

PERSONAL_i: $\sigma_{Cód_centros='i'}$ (INF_PERSONAL) donde i = {01,02,03,04}.

> *"Cada Centro médico (Cód_Centro = {02, 03, 04}) debe tener la relación de personal sanitario que trabaja en el mismo, así como el horario de la consulta y la especialidad que desarrolla."*

Para tener la relación del personal sanitario en cada sede debemos partir de la tabla PERSONAL_SANITARIO y mediante los fragmentos PERSONAL_i realizar la fragmentación horizontal derivada.

PERSONAL_SANITARIO_i:

PERSONAL_SANITARIO $\ltimes_{Cód_Empleado}$ PERSONAL_i donde i = {01,02,03,04}

Hay que destacar que aunque en la sede central no haya personal sanitario, se ha considerado que para mayor completitud en la fragmentación mantendremos este fragmento vacío.

Como además se quiere la información acerca de las consultas de cada centro. cuál es su horario y qué especialidad se trata se tendrá que fragmentar la relación CONSULTAS verticalmente.

CONSULTAS_i: $\sigma_{Cód_centros='i'}$ (CONSULTAS) donde i = {01,02,03,04}

Una vez aplicada esta fragmentación sólo nos falta obtener las especialidades desarrolladas en cada una de las sedes de la sociedad médicas para ello fragmentaremos la tabla ESPECIALIDADES mediante un fragmentación horizontal derivada.

ESPECIALIDAD_i: ESPECIALIDADES $\ltimes_{Cód_Esp=Cód_Especialidad}$ CONSULTAS_i

Donde i = {01,02,03,04}

Esquema de asignación:

La asignación se especificará en una tabla indicando en la primera fila las sedes o nodos de la red que forma parte del sistema distribuido y en la primera columna la tabla base de la que se ha obtenido un determinado fragmento. Debemos aclarar que como la relación CENTRO es poco dinámica, es decir, no sufre apenas actualizaciones, se duplicará por todas los centros.

	Centro01	Centro02	Centro03	Centro04
CENTROS	CENTROS	CENTROS	CENTROS	CENTROS
PERSONAL	NÓMINA_ PERSONAL			
INF_PERSONAL	PERSONAL_01	PERSONAL_02	PERSONAL_03	PERSONAL_04
PERSONAL_ SANITARIO	PERSONAL_ SANITARIO_01	PERSONAL_ SANITARIO_02	PERSONAL_ SANITARIO_03	PERSONAL_ SANITARIO_04
CONSULTAS	CONSULTAS_01	CONSULTAS_02	CONSULTAS_03	CONSULTAS_04
ESPECIALIDADES	ESPECIALIDAD_01	ESPECIALIDAD_02	ESPECIALIDAD_03	ESPECIALIDAD_04

PROBLEMA 2: BDD UNIVERSIDAD CARLOS III

Enunciado

La Universidad Carlos III tiene en la actualidad tres campus distribuidos en las siguientes poblaciones de la Comunidad de Madrid: Getafe, Leganés y Colmenarejo. Esta ampliación ha repercutido en los sistemas informáticos de esta Universidad llevándose a cabo el tendido de una red para unir los tres campus y ahora se baraja la posibilidad de diseñar una Base de Datos Distribuida, actualmente centralizada en Getafe, para manejar de forma autónoma información sobre las titulaciones ofertadas.

En cada uno de los campus se quiere crear un departamento para la gestión de las distintas titulaciones impartidas en ese campus, además de manejar los datos acerca de los cursos de los que consta cada una de ellas y los grupos que forman parte de estos cursos. En este departamento también se desea guardar información de los profesores (nombre, e-mail y número de despacho) que imparten clases en cada uno de los campus.

El departamento de gestión de nóminas y contratación de profesorado se mantendrá en Getafe.

El esquema lógico global de la Base de Datos es el siguiente:

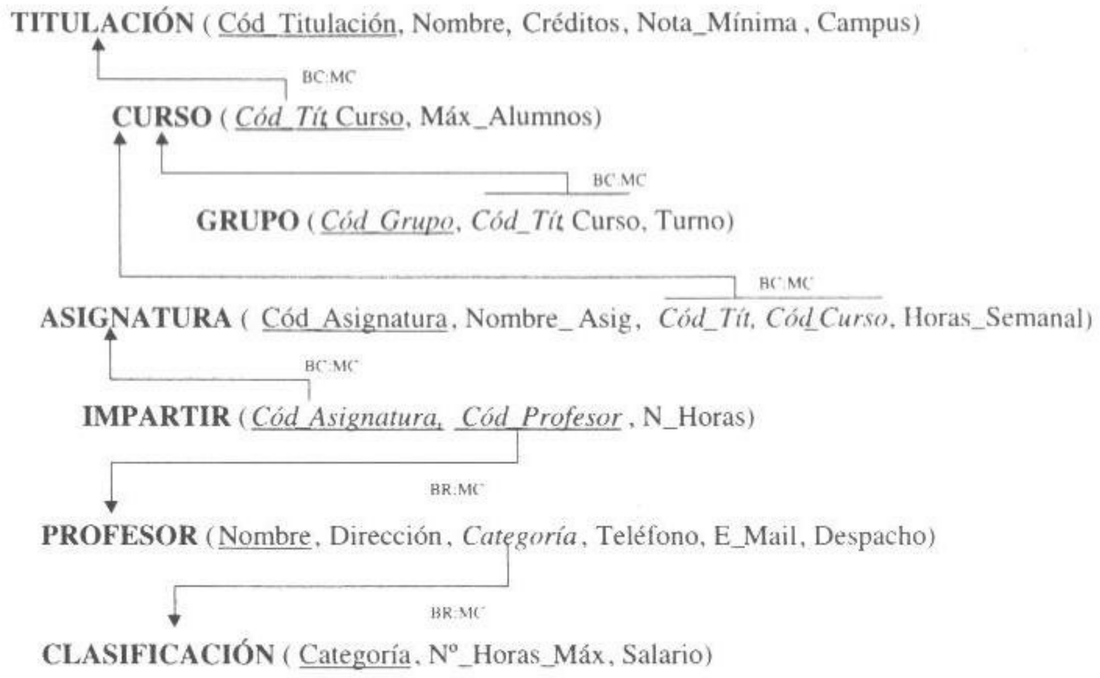


Figura 4.8. Esquema relacional correspondiente a una BD centralizada para la gestión de la Universidad Carlos III

Diseñar los esquemas de fragmentación y asignación de la Base de Datos Distribuida, explicando las consideraciones que se han tenido en cuenta para realizar este diseño.

En relación con el diseño realizado en el apartado anterior explicar qué información se guardaría en el diccionario global de la BDD para los esquemas de fragmentación y asignación.

Discusión del enunciado

Esquema de fragmentación:

> *"En cada uno de los campus se quiere crear un departamento para la gestión de las distintas titulaciones impartidas en ese campus, además de manejar los datos acerca de los cursos de los que consta cada una de ellas y los grupos que forman parte de estos cursos. En este departamento también se desea guardar información de los profesores (nombre, e-mail y número de despacho) que imparten clases en cada uno de los campus."*

Como se quiere que en cada campus se maneje la información de las titulaciones que allí se imparten tendremos que aplicar una fragmentación horizontal en la relación TITULACIÓN.

TITULACIÓN_i: $\sigma_{campus='i'}$(TITULACIÓN)

Donde i = {Getafe, Leganés, Colmenarejo}.

Para tener la información completa sobre las titulaciones de cada campus necesitamos conocer qué cursos se imparten y los grupos formados para cada curso, por lo que tendremos, tanto para la relación CURSO como para la relación GRUPO, que realizar una fragmentación horizontal derivada.

CURSO_i: CURSO $\ltimes_{Cód_Titulación}$ TITULACIÓN_i

GRUPO_i: GRUPO $\ltimes_{Cód_Titulación,\ Curso}$ CURSO_i

Donde i = {Getafe, Leganés, Colmenarejo}

De la misma forma asignamos las asignaturas impartidas en las distintas titulaciones a los campus correspondientes realizando una fragmentación horizontal derivada con los fragmentos de CURSO_i.

ASIGNATURA_i: ASIGNATURA $\ltimes_{Cód_Titulación,\ Curso}$ CURSO_i

Donde i = {Getafe, Leganés, Colmenarejo}

Por último, para saber los profesores que dan clases en los distintos campus realizamos otra fragmentación derivada con los fragmentos de las asignaturas.

IMPARTIR_i: IMPARTIR $\ltimes_{Cód_Asignatura}$ ASIGNATURA_i

Donde i = {Getafe, Leganés, Colmenarejo}

> *"El departamento de gestión de nóminas y contratación de profesorado se mantendrá en Getafe."*

Como sólo necesitamos algunos datos acerca de los profesores que imparten asignaturas en las titulaciones de cada campus, primero realizamos una fragmentación vertical para quedarnos con la información que nos interesa.

INF_PROFESOR: $\Pi_{Nombre,\ E\text{-}Mail,Despacho}$(PROFESOR)

NOMINA_PROFESOR: $\Pi_{Nombre,\ Dirección,\ Teléfono,\ Categoría}$(PROFESOR)

Para luego realizar una fragmentación horizontal derivada y de esta forma obtener fragmentos con los atributos de los profesores necesarios en cada uno de los campus.

INF_PROF_i: INF_PROFESOR $\ltimes_{Cód_Asignatura}$ IMPARTIR_i

Donde i = {Getafe, Leganés, Colmenarejo}

Esquema de asignación:

A continuación presentamos una tabla con la asignación de cada uno de los fragmentos obtenidos en el apartado anterior. La relación CLASIFICACIÓN tendrá pocas actualizaciones, por lo que se puede duplicar en todas las sedes.

	Getafe	Colmenarejo	Leganés
TITULACIÓN	TITULACIÓN_Getafe	TITULACIÓN_ Colmenarejo	TITULACIÓN_Leganés
CURSO	CURSO_Getafe	CURSO_Colmenarejo	CURSO_Leganés
GRUPO	GRUPO_Getafe	GRUPO_Colmenarejo	GRUPO_Leganés
ASIGNATURA	ASIGNATURA_Getafe	ASIGNATURA_ Colmenarejo	ASIGNATURA_Leganés
IMPARTIR	IMPARTIR_Getafe	IMPARTIR_Colmenarejo	IMPARTIR_Leganés
PROFESOR	NÓMINA_PROFESOR		
CLASIFICACIÓN	CLASIFICACIÓN	CLASIFICACIÓN	CLASIFICACIÓN
INF_PROFESOR	INF_PROF_Getafe	INF_PROF_Colmenarejo	INF_PROF_Leganés

Contenido del diccionario:

El diccionario global de la base de datos distribuida anteriormente diseñada ha de contener información sobre el esquema de fragmentación, de asignación, si existe replicación y además el esquema global de la base de datos.

El esquema global de la BD contiene la misma información que si se tratara de una BD centralizada, por lo tanto vamos a describir sólo aquella información que caracteriza a una BDD relacional.

Al ser la estructura del diccionario relacional, la información sobre los fragmentos se guardará en una relación que contiene como mínimo los siguientes campos: "Nombre_Tabla_Base", es decir, nombre de la relación a la que se aplica un determinado tipo de fragmentación. "Nombre_Fragmento", "Sede", "Tipo", que tomará los valores H, HD o V dependiendo de si la fragmentación es horizontal, horizontal derivada o vertical, "Condición" que nos indicará qué condiciones hemos

utilizado en los operadores del álgebra relacional para generar el fragmento y "Replicado" que tomará los valores S o N dependiendo de si el fragmento, está replicado o no.

Nombre_Tabla_ Base	Nombre_ Fragmento	Sede	Tipo	Condición	Replicado
TITULACIÓN	TITULACIÓN _Getafe	Getafe	H	Campus= 'Getafe'	NO
CURSO	CURSO _Getafe	Getafe	HD	Cód_Titulación = TITULACIÓN_ Getafe.Cód_Tit	NO
PROFESOR	INF_ PROFESOR		V	Nombre, E-Mail,Despacho	NO
INF_PROFESOR	INF_PROF_ GETAFE	Getafe	HD	IMPARTIR_ GETAFE. Cód_Profesor = PROFESOR_ INF. Nombre	NO
PROFESOR	NÓMINA_ PROFESOR	Getafe	V	Nombre, Dirección, Categoría, Teléfono	NO

Para los demás fragmentos, la información guardada en el diccionario global sería análoga.

Si los fragmentos estuvieran duplicados se debería guardar la ubicación de las replicaciones, es decir, los nodos o sedes donde se encuentra cada uno de los fragmentos duplicados.

PROBLEMA 3: BDD SERVICIOS INFORMÁTICOS

Enunciado

La empresa de servicios informáticos INFOSYSTEM está ubicada en cuatro edificios cuyos códigos de identificación son '00', '01', '02' y '03', conectados entre sí por una red. En cada uno de ellos se encuentra una o varias unidades de desarrollo, por ejemplo, en el edificio cuyo código es '00' se halla la unidad de Recursos Humanos.

Los empleados de la empresa y los proyectos que se realizan en ésta, están asignados a una sola unidad de desarrollo, aunque un empleado puede trabajar en un proyecto que no pertenezca a su unidad.

Los proyectos se realizan para una determinada empresa de la cual nos interesa saber el nombre y el teléfono para que dicha información este disponible en cada uno de los edificios donde se desarrollan proyectos para estas empresas, y los demás datos se los quedará Recursos Humanos para tramitar las facturas correspondientes.

Pueden existir proyectos internos de los cuales no se tramita factura, por lo que no es necesario que Recursos Humanos posea información de ellos. Estos proyectos tienen como código de empresa cliente el '00000'.

Suponiendo que el esquema de la base de datos relacional es el de la figura 3.3:

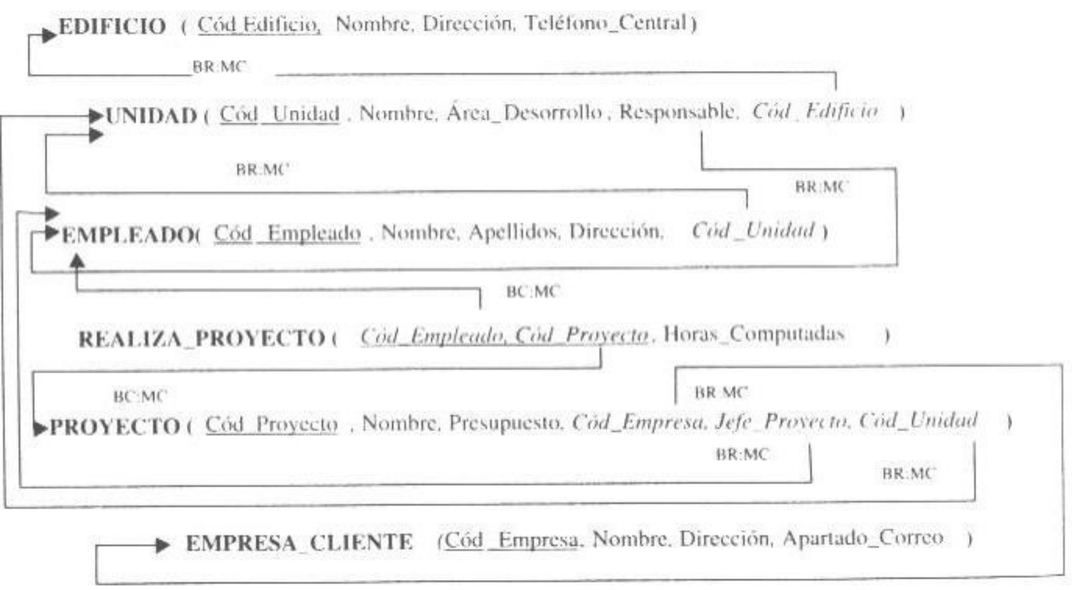


Figura 4.9. Esquema relacional correspondiente a una BD centralizada para la gestión de una empresa de Servicios Informáticos.

1. Diseñar los esquemas de fragmentación y asignación de la Base de Datos Distribuida.
   Tener en cuenta que cada unidad de desarrollo gestiona las nóminas de sus empleados y necesita la información de los proyectos en los que trabaja para poder pagarles conforme a las horas que han computado en cada uno de ellos.

2. Conforme al diseño realizado en el apartado anterior, modificar si se cree necesario los esquemas de fragmentación y de asignación para conseguir que la siguiente consulta se realice de manera local: "código de los proyectos que desarrollan las unidades de un determinado edificio".

Discusión del enunciado

APARTADO 1

Esquema de fragmentación:

> *"La empresa de servicios informáticos INFOSYSTEM está ubicada en cuatro edificios cuyos códigos de identificación son '00', '01', '02' y '03', conectados entre sí por una red. En cada uno de ellos se encuentra una o varias unidades de desarrollo, por ejemplo, en el edificio cuyo código es '00' se halla la unidad de Recursos Humanos."*

> *"Los empleados de la empresa y los proyectos que se realizan en ésta, están asignados a una sola unidad de desarrollo, aunque un empleado puede trabajar en un proyecto que no pertenezca a su unidad."*

Como se puede observar en el esquema relacional, en un edificio existe una o varias unidades de desarrollo, pero una determinada unidad sólo se encuentra en un edificio en concreto. Por tanto, nos interesaría tener en cada uno de los edificios tanto el personal como los proyectos que se llevan a cabo en él.

La relación UNIDAD la fragmentamos horizontalmente para saber las unidades asignadas a cada edificio.

UNIDAD_i: $\sigma_{Cód_Edificio='i'}$(UNIDAD) donde i = 0..3

Para saber los empleados designados a las distintas unidades realizamos una fragmentación horizontal derivada.

EMPLEADO_i: EMPLEADO $\ltimes_{Cód_Unidad}$ UNIDAD_i

donde i = {00, 01,02,03}

De igual forma, obtenemos los fragmentos con los proyectos que se desarrollan en cada unidad.

PROYECTO_i: PROYECTO $\ltimes_{Cód_Unidad}$ UNIDAD_i

donde i = {00, 01,02,03}

> *"Los proyectos se realizan para una determinada empresa de la cual nos interesa saber el nombre y el teléfono para que dicha información este disponible en cada uno de los edificios donde se desarrollan proyectos para estas empresas, y los demás datos se los quedará Recursos Humanos para tramitar las facturas correspondientes."*

Para que Recursos Humanos tenga la información necesaria de la empresa cliente fragmentamos verticalmente la tabla EMPRESA_CLIENTE.

INF_EMPRESA: $\Pi_{Cód_Empresa,\ Nombre}$ (EMPRESA_CLIENTE)

FACT_EMPRESA:

$\Pi_{Cód_Empresa,Dirección,Apartado_Correo}$(EMPRESA_CLIENTE)

Los proyectos internos que tienen como Cod_Empresa = '00000' no se mandan a facturar, pero esta información es muy pequeña (una tupla) y podemos dejarla en la tabla FACT_EMPRESA y en INF_EMPRESA.

La información de las empresas que participan en proyectos asociados a cada una de las unidades se obtiene:

INF_EMPRESA_i: INF_EMPRESA $\bowtie_{Cód_Empresa}$ PROYECTO_i

donde i = {00, 01,02,03}

"Tener en cuenta que cada unidad de desarrollo gestiona las nóminas de sus empleados y necesita la información de los proyectos en los que trabaja para poder pagarles conforme a las horas que han computado en cada uno de ellos."

Para que cada edificio tenga el código de los proyectos en los que trabaja el personal de las distintas unidades asignadas a éste realizamos la siguiente fragmentación horizontal derivada.

REALIZA_PROYECTO_i:

REALIZA_PROYECTO$_{Cód_Proyecto}$ $\bowtie$ PROYECTO_i

donde i = {00, 01,02,03}

Esquema de asignación:

A continuación presentamos una tabla con la asignación de cada uno de los fragmentos obtenidos en el apartado anterior. La relación EDIFICIO, que aparece en el esquema relacional, es una relación que suponemos estática, es decir, en ella no se van a realizar operaciones de actualización solo de consulta, por lo que decidimos duplicarla en cada uno de las sedes.

	Edf.00	Edf.01	Edf.02	Edf.03
UNIDAD	UNIDAD_00	UNIDAD_01	UNIDAD_02	UNIDAD_03
EMPLEADO	EMPLEADO_00	EMPLEADO_01	EMPLEADO_02	EMPLEADO_03
PROYECTO	PROYECTO_00	PROYECTO_01	PROYECTO_02	PROYECTO_03
EMPRESA_ CLIENTE	FACT_EMPRESA			
INF_EMPRESA	INF_ EMPRESA_00	INF_ EMPRESA_01	INF_ EMPRESA_02	INF_ EMPRESA_03
REALIZA_ PROYECTO	REALIZA_ PROYECTO_00	REALIZA_ PROYECTO_01	REALIZA_ PROYECTO_02	REALIZA_ PROYECTO_03
EDIFICIO	EDIFICIO	EDIFICIO	EDIFICIO	EDIFICIO

APARTADO 2

Esquema de fragmentación:

Para poder mantener el esquema de fragmentación anterior y además poder dar la información referente al código de los empleados que trabajan en los proyectos desarrollados por las unidades asignadas a los distintos edificios, realizaremos la siguiente fragmentación horizontal derivada.

EMP_i_j : REALIZA_PROYECTO_i $\bowtie_{Cod_Emp}$ EMPLEADO_ j

donde i,j = {00,01,02,03}

Por ejemplo, la información que contiene el fragmento EMP_00_01 sería los empleados asignados al edificio 01 que participan en los proyectos llevados a cabo por las unidades asociadas al edificio 00.

Esquema de asignación:

La tabla descriptora de la asignación de los fragmentos contendrá los nuevos fragmentos hallados en el esquema de fragmentación anterior y que se asignarán de la forma siguiente:

En el EDIFICIO K: EMP_K_j donde j, K = {00, 01, 02, 03} que serán los empleados de todos los edificios que trabajan en proyectos del edificio K. Y además, EMP_i_K donde i = {00, 01, 02, 03}- {K} y K = {00, 01, 02, 03} que serán los

mpleados del edificio K que trabajan en proyectos de otros edificios. En este caso el ndice i no recorre todos los números de edificio para no repetir el fragmento EMP_K_K.

	Edf.00	Edf.01	Edf.02	Edf.03
UNIDAD	UNIDAD_00	UNIDAD_01	UNIDAD_02	UNIDAD_03
EMPLEADO	EMPLEADO_00	EMPLEADO_01	EMPLEADO_02	EMPLEADO_03
PROYECTO	PROYECTO_00	PROYECTO_01	PROYECTO_02	PROYECTO_03
EMPRESA_ CLIENTE	FACT_EMPRESA			
INF_EMPRESA	INF_ EMPRESA_00	INF_ EMPRESA_01	INF_ EMPRESA_02	INF_ EMPRESA_03
REALIZA_ PROYECTO_00	EMP_00_00 EMP_00_01 EMP_00_02 EMP_00_03	EMP_00_01	EMP_00_02	EMP_00_03
REALIZA_ PROYECTO_01	EMP_01_00	EMP_01_00 EMP_01_01 EMP_01_02 EMP_01_03	EMP_01_02	EMP_01_03
REALIZA_ PROYECTO_02	EMP_02_00	EMP_02_01	EMP_02_00 EMP_02_01 EMP_02_02 EMP_02_03	EMP_02_03
REALIZA_ PROYECTO_03	EMP_03_00	EMP_03_01	EMP_03_02	EMP_03_00 EMP_03_01 EMP_03_02 EMP_03_03
EDIFICIO	EDIFICIO	EDIFICIO	EDIFICIO	EDIFICIO

Tendríamos duplicación en algunos de los fragmentos pero conseguiríamos que la consulta se realizara de manera local.

PROBLEMA 4: BDD OFICINAS DEL INEM

Enunciado

Las oficinas del INEM de la Comunidad de Madrid se encuentran agrupadas en cinco zonas: Norte, Sur, Este, Oeste y Centro. Cada una de ellas posee un ordenador en el cual se almacenan los datos de las oficinas asignadas a la zona.

En cada una de las oficinas se imparten cursos para los desempleados. A cada oficina se le asignan, de forma exclusiva, distintos cursos, es decir, cada oficina es especialista en la docencia de unos determinados cursos, los cuales no se pueden impartir en ninguna otra oficina.

Los desempleados están adscritos en una única oficina, aunque pueden realizar cursos en cualquier oficina de la Comunidad de Madrid.

Suponiendo que todos los ordenadores de las cinco zonas están conectados entre sí y que la base de datos relacional es la que figura a continuación:

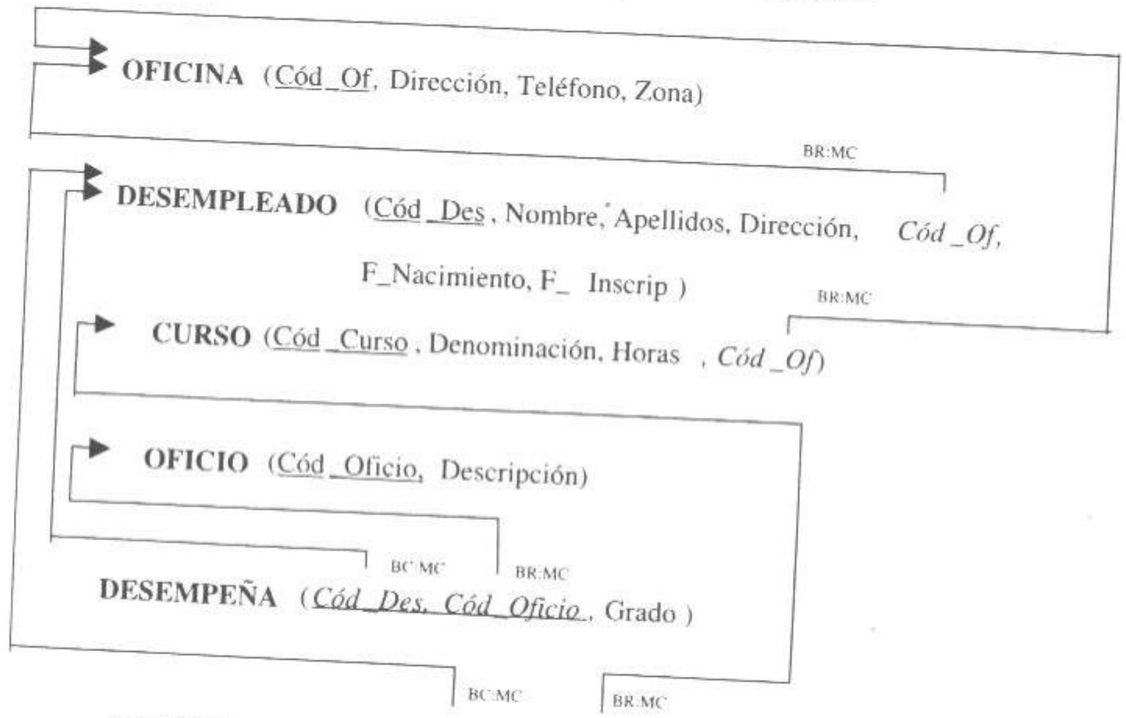


Figura 4.10. Esquema relacional correspondiente a una BD centralizada para la gestión de cursos del INEM

1. Diseñar los esquemas de fragmentación y asignación de la Base de Datos Distribuida teniendo en cuenta las siguientes especificaciones:

   - En cada una de las zonas se precisan los siguientes datos de los desempleados: *Cód_Des, Nombre, Apellidos, Dirección, Cód_Of*
   - La zona centro es la encargada de realizar las estadísticas de los desempleados de la Comunidad, para lo que necesita su fecha de nacimiento y la fecha en la que se inscribió en el paro.

- Cada zona precisa saber, con total autonomía local, los desempleados que han realizado los cursos asignados en sus oficinas.

Además de las especificaciones anteriores, cada zona desea saber los cursos que han realizado los desempleados de la misma. Modificar, si se considera conveniente, los esquemas anteriores sabiendo que la red que une las distintas zonas es muy lenta y no es conveniente el tráfico de los datos por la misma, ya que no es una línea dedicada.

cusión del enunciado

ARTADO 1

uema de fragmentación:

Vamos a ir construyendo los fragmentos según las especificaciones del problema.

> *"En cada una de las zonas se precisan los siguientes datos de los desempleados: Cód_Des, Nombre, Apellidos, Dirección, Cód_Of."*

> *"La zona centro es la encargada de realizar las estadísticas de los desempleados de la Comunidad, para lo que necesita su fecha de nacimiento y la fecha en la que se inscribió en el paro."*

Para quedarnos con aquellos atributos que se necesitan en cada una de las sedes remos que realizar una fragmentación vertical en la relación DESEMPLEADO.

DESEMP1: $\Pi_{\text{Cód_Des, F_Nacimiento,F_Inscrip}}$(DESEMPLEADO)

DESEMP2: $\Pi_{\text{Cód_Des, Nombre, Apellidos, Dirección, Cod_Of}}$(DESEMPLEADO)

De esta fragmentación obtenemos dos fragmentos denominados, DESEMP1 y SEMP2, que cumplen que son disjuntos y cuya unión es la relación SEMPLEADO, asegurando de esta forma que la fragmentación sea correcta.

La relación DESEMP1 servirá para que la zona centro obtenga sus estadísticas y la otra relación, DESEMP2, tenemos los datos personales de los desempleados de omunidad de Madrid, pero no están por zonas. Para conseguir los datos personales los desempleados de cada zona necesitaremos saber cúales son las oficinas espondientes a cada zona.

En un primer lugar la relación OFICINA sufrirá una fragmentación horizontal a condición será que el atributo "zona" tome el valor de cada zona.

OF_i: $\sigma_{\text{Zona}='i'}$(OFICINA) donde i = {Norte, Sur, Este, Oeste, Centro}

Esta fragmentación nos da como resultado cinco relaciones con los mismos atributos que la relación OFICINA, pero con la información de las oficinas asignadas a cada zona.

Finalmente, para conseguir los fragmentos con la información referente a los desempleados de cada zona se realizará fragmentación horizontal derivada.

DESEMP_ZONA_i: DESEMP2 $\underset{Cód_Of}{\ltimes}$ OF_i

donde i = {Norte, Sur, Este, Oeste, Centro}

Se podría pensar en otro tipo de fragmentación para conseguir los desempleados asignados a cada zona. Primero realizando una fragmentación derivada de la relación DESEMPLEADO con las oficinas de cada zona.

DESEMP_i: DESEMPLEADO $\underset{Cód_Of}{\ltimes}$ OF_i

donde i = {Norte, Sur, Este, Oeste, Centro}

Estos cinco fragmentos tendrían todos los atributos de la relación DESEMPLEADO, por lo que para que la zona centro pudiera realizar sus estadísticas con total autonomía se debería realizar una nueva fragmentación, pero esta vez vertical en cada uno de ellos.

DESEMP1_i: $\Pi_{\text{Cód_Des, F_Nacimiento,F_Inscrip}}$(DESEMP_i)

donde i = {Norte, Sur, Este, Oeste, Centro}

De esta forma se aumentaría el número de fragmentos, lo que supone pérdida de eficiencia a la hora de realizar cualquier operación, tanto de actualización como de consulta, sobre la unión de fragmentos que se encuentran en la misma sede.

> *"Cada zona precisa saber, con total autonomía local, los desempleados que han realizado los cursos asignados en sus oficinas."*

Se realiza una fragmentación horizontal derivada para saber los cursos que se han impartido en cada zona.

CURSO_i: CURSO $\underset{Cód\ Curso}{\ltimes}$ OF_i donde i = {Norte, Sur, Este, Oeste, Centro}

Como cada zona precisa saber los desempleados que han realizado los cursos asignados a sus oficinas, se realiza una fragmentación horizontal derivada:

REALIZA_CURSO_i: REALIZA_CURSO $\underset{Cód\ Curso}{\ltimes}$ CURSO_i

donde i = {Norte, Sur, Este, Oeste, Centro}

De esta fragmentación obtenemos el código del desempleado y el código de los ursos que se han realizado en cada zona o sede.

Una vez cubierta las especificaciones pedidas por el enunciado tenemos que omprobar si las relaciones que nos quedan en el enunciado deben ser también agmentadas para conseguir más autonomía local.

La relación DESEMPEÑA se puede fragmentar para que en cada zona se ncuentren los datos de los trabajos desempeñados por los desempleados inscritos en misma. Para ello utilizaremos una fragmentación horizontal derivada.

DESEMPEÑA_i: DESEMPEÑA $\ltimes_{Cód_Des}$ DESEMP_ZONA_i

donde i = {Norte, Sur, Este, Oeste, Centro}

Por último nos queda la relación OFICIO, que si suponemos que es una tabla con ocas operaciones de actualización y que sólo se realizan consultas, duplicaremos en odas las sedes.

l esquema de asignación:

Describimos a continuación, mediante una tabla, la asignación de cada fragmento relación a las sedes correspondientes, indicando en la primera columna la relación ase de la que procede cada fragmento.

	Norte	Sur	Centro	Este	Oeste
OFICINA	OF_Norte	OF_Sur	OF_Centro	OF_Este	OF_Oeste
DESEM-PLEADO			DESEMP1		
DESEMP2	DESEMP_ ZONA_Norte	DESEMP_ ZONA_Sur	DESEMP_ ZONA_Centro	DESEMP_ ZONA_Este	DESEMP_ ZONA_Oeste
CURSO	CURSO_Norte	CURSO_Sur	CURSO_Centro	CURSO_Este	CURSO_Oeste
DESEMPEÑA	DESEMPEÑA_ Norte	DESEMPEÑA_ Sur	DESEMPEÑA_ Centro	DESEMPEÑA_ Este	DESEMPEÑA_ Oeste
REALIZA_ CURSO	REALIZA_ CURSO_Norte	REALIZA_ CURSO_Sur	REALIZA_ CURSO_Centro	REALIZA_ CURSO_Este	REALIZA_ CURSO_Oeste
OFICIO	OFICIO	OFICIO	OFICIO	OFICIO	OFICIO

APARTADO 2

Para saber en cada zona o sede los cursos que han realizado los desempleados de a misma, se pueden considerar dos soluciones:

La primera de ellas sería volver a fragmentar cada uno de los fragmentos de REALIZA_CURSO. Por ejemplo:

G_Norte_j : REALIZA_CURSO_Norte $\ltimes_{Cód_Des}$ DESEMP_ZONA_j

donde j = {Norte, Sur, Este, Oeste, Centro}

La información que obtendríamos con cada uno de estos fragmentos sería los desempleados de la zona j que han realizado algún curso de los ofrecidos en la zona Norte.

G_i_j : REALIZA_CURSO_i $\ltimes_{Cód_Des}$ DESEMP_ZONA_j

donde i, j = {Norte, Sur, Este, Oeste, Centro}

Y por lo tanto, el fragmento G_i_j contendrá los desempleados de la zona j que han realizado algunos de los cursos ofertados en la zona i.

La asignación de estos fragmentos se realiza de la siguiente forma:

En la zona k tendríamos los fragmentos G_K_j y los G_i_K donde i = {Norte, Sur, Este, Oeste, Centro}- {K} y donde j, K = {Norte, Sur, Este, Oeste, Centro}

	Norte	Sur	Centro	Este	Oeste
OFICINA	OF_Norte	OF_Sur	OF_Centro	OF_Este	OF_Oeste
DESEM-PLEADO			DESEMP1		
DESEMP2	DESEMP_ ZONA_Norte	DESEMP_ ZONA_Sur	DESEMP_ ZONA_Centro	DESEMP_ ZONA_Este	DESEMP_ ZONA_Oeste
CURSO	CURSO_Norte	CURSO_Sur	CURSO_Centro	CURSO_Este	CURSO_Oeste
DESEMPEÑA	DESEMPEÑA_ Norte	DESEMPEÑA_ Sur	DESEMPEÑA_ Centro	DESEMPEÑA_ Este	DESEMPEÑA_ Oeste
REALIZA_ CURSO_Norte	G_Norte_Norte G_Norte_Sur G_Norte_Centro G_Norte_Este G_Norte_Oeste	G_Norte_Sur	G_Norte_Centro	G_Norte_Este	G_Norte_Oeste
REALIZA_ CURSO_Sur	G_Sur_Norte	G_Sur_Norte G_Sur_Sur G_Sur_Centro G_Sur_Este G_Sur_Oeste	G_Sur_Centro	G_Sur_Este	G_Sur_Oeste

	Norte	Sur	Centro	Este	Oeste
REALIZA_CURSO_Centro	G_Centro_Norte	G_Centro_Sur	G_Centro_Norte G_Centro_Sur G_Centro_Centro G_Centro_Este G_Centro_Oeste	G_Centro_Este	G_Centro_Oeste
REALIZA_CURSO_Este	G_Este_Norte	G_Este_Sur	G_Este_Centro	G_Este_Norte G_Este_Sur G_Este_Centro G_Este_Este G_Este_Oeste	G_Este_Oeste
REALIZA_CURSO_Oeste	G_Oeste_Norte	G_Oeste_Sur	G_Oeste_Centro	G_Oeste_Este	G_Oeste_Norte G_Oeste_Sur G_Oeste_Centro G_Oeste_Este G_Oeste_Oeste
OFICIO	OFICIO	OFICIO	OFICIO	OFICIO	OFICIO

Como se puede observar en la tabla tendremos duplicación de algunos fragmentos, pero aseguramos total autonomía en cada una de las sedes.

Otra solución posible sería la duplicación, en cada una de las zonas, de la tabla REALIZA_CURSO, ya que esta tabla no sufre actualizaciones continuas, y al tener pocos atributos su almacenamiento no sería muy costoso.

CAPÍTULO 5

DISEÑO CON HERRAMIENTAS CASE

INTRODUCCIÓN

En este capítulo se pretende mostrar cómo se pueden realizar las distintas fases del diseño de bases de datos mediante la tecnología CASE. Para ello, hemos planteado la solución de dos problemas. El primero se ha desarrollado con la herramienta CASE ERwin y el segundo con la herramienta CASE proporcionada por ORACLE, Designer2000.

En cada problema se parte del esquema E/R teórico y se introduce en una herramienta CASE adaptándolo a la notación que utiliza, comentando en cada caso las diferencias que existen y las soluciones adoptadas para realizar la adaptación, evitando la pérdida de semántica. A continuación, se realiza el paso al relacional y la generación automática de código.

En ambas herramientas se ha utilizado la conexión al sistema gestor de bases de datos ORACLE (aunque las herramientas permiten conexiones a otros sistemas de bases de datos, como Informix, Access, etc.) y por tanto, todo el código SQL que se genera está especificado para ORACLE 8.0, a diferencia de los ejercicios resueltos en el capítulo 2.1. en los que aparecía el código en SQL 92.

Hay que aclarar, aunque se hablará de ello en los ejercicios, que no todas las fases del desarrollo de una base de datos están automatizadas en las herramientas CASE. Lo importante es tener los conocimientos teóricos acerca de cómo diseñar una base de datos para que luego la herramienta CASE ofrezca el soporte adecuado en la implementación de la base de datos y de esta manera presentar una ayuda al diseñador.

CASO PRÁCTICO 1: BASURA ESPACIAL

Enunciado

Desde los inicios de la era espacial hasta nuestros días, la basura espacial se ha convertido en un problema que dificulta la realización de misiones espaciales. Por este motivo, se crea una comisión internacional para el tratamiento del problema. Una de las primeras medidas de esta comisión es la de crear un sistema de información para el seguimiento de la misma. El análisis realizado es el siguiente:

Basura espacial es cualquier objeto desprendido de (o producido por) alguna nave que no sea de utilidad y cuyo diámetro sea superior a una micra o su peso superior a 1 miligramo. A cada uno de estos objetos se le asigna un identificador secuencial numérico y se pretende recoger tanto su tamaño, como su peso, velocidad, última órbita conocida (que puede no coincidir con aquella en la que se lanzó la nave que la produjo) y las distintas posiciones del mismo (coordenadas polares r y δ respecto al centro de la tierra) registradas en determinados momentos.

Cada nave es una clase de nave en concreto y se identifica por una matrícula que contiene el código de la clase de nave a la que pertenece. Las naves pueden tener tripulantes, recogiéndose sólo el nombre de cada tripulante, que se supone único. Toda nave tiene una misión (científica, militar o espía) y es propiedad de una agencia espacial. Una nave, a lo largo de su vida útil, puede cambiar de órbita. Se registrarán todas las órbitas de cada nave, así como las fechas (día y hora) inicial y final de permanencia en cada órbita.

Para cada clase de nave se registrarán todos los tipos de componentes que posee: a cada uno de ellos se le asigna un código y se recoge su diámetro y peso.

Las órbitas pueden ser circulares o elípticas y geoestacionarias o no; se identifican por su altura, su sentido (coordenadas polares r y δ) y su excentricidad (τ). Una órbita circular tiene excentricidad 0.

Se registran los lanzamientos que realizan las agencias espaciales de una nave y las órbitas iniciales en las que las posicionan (en caso de que la nave esté en órbita). Se desea almacenar la fechas en la que la agencia lanzó la nave.

Las agencias espaciales pueden ser de dos tipos: públicas o privadas; para las públicas se recoge el nombre de los Estados que las mantienen, mientras que para las

privadas se recoge las empresas (CIF, nombre, capital, etc.) que las financian. Ninguna agencia pública puede recibir financiación de empresas, ni ninguna agencia privada puede recibir aportación de ningún Estado. Se desea recoger la proporción (en tanto por ciento) de la participación de cada empresa en la contribución a la agencia. Las agencias espaciales tendrán un nombre y un número de personas trabajando en ellas. Toda agencia privada tiene que ser supervisada por una agencia pública.

Una agencia espacial es responsable de limpiar la basura que produce cualquiera de sus naves; para ello, una de las técnicas más empleadas es el uso de un rayo láser para la fragmentación del objeto (basura) hasta que deje de resultar peligroso. El problema de esta técnica es que, aunque reduce el tamaño de los objetos, aumenta su número, por lo que, cada vez que se fragmenta un objeto, se desea registrar el objeto del cual proceden los nuevos objetos producidos y la fecha en la que ha tenido lugar la fragmentación.

Esquema E/R

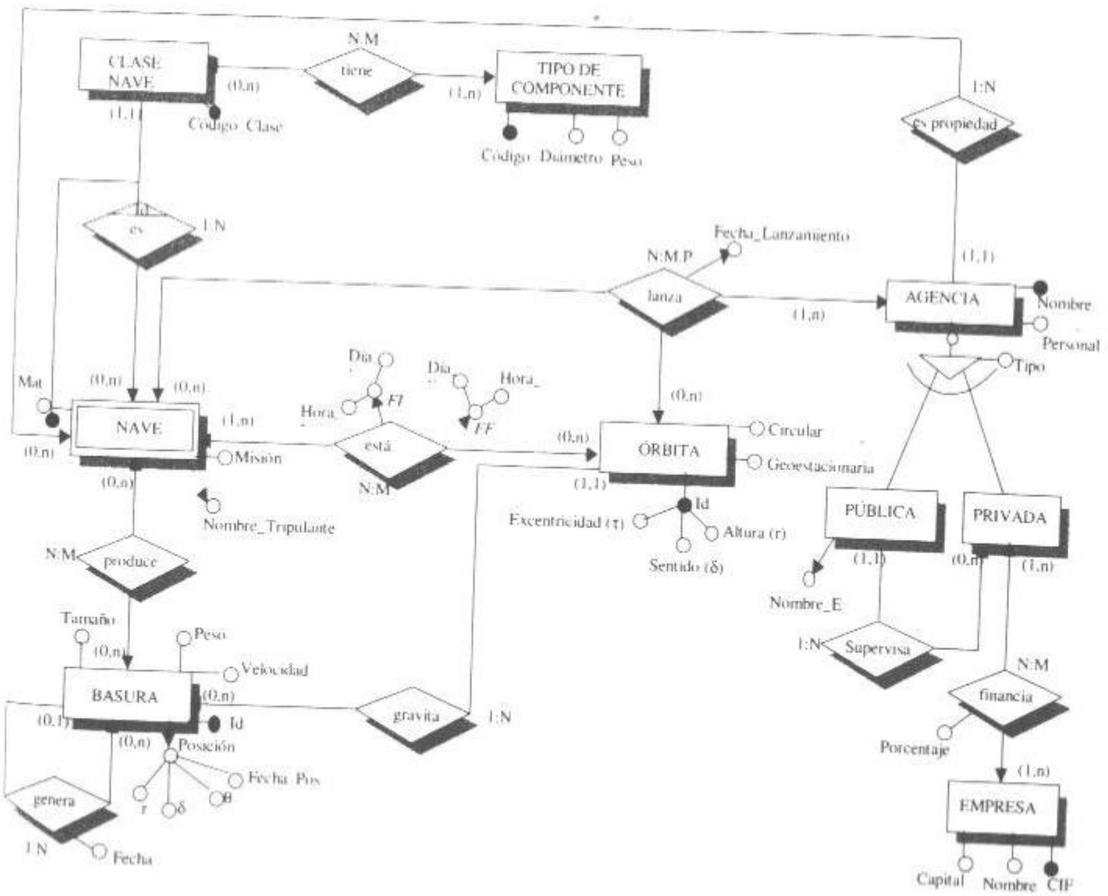


Figura 5.1. Esquema E/R

NSIDERACIONES SOBRE LA SOLUCIÓN PROPUESTA

Se define como basura espacial todo objeto procedente de una nave cuyo diá-ro sea superior a una micra o cuyo peso sea superior a un miligramo. Esta de-ción puede verse como una restricción sobre las ocurrencias de la entidad *BA-RA*. Esta restricción no se puede recoger en el modelo E/R, aunque podría ɔgerse en el modelo relacional mediante un CHECK entre los atributos "Tamaño" Peso". Por otro lado, el atributo "Posición" de *BASURA* es un atributo multivaluado , aparte de las coordenadas, incluye también la fecha en la que se tomó la posición.

La última órbita de la basura espacial debe pertenecer a una de las órbitas ibles, es decir, debe ser una ocurrencia de la entidad *ÓRBITA* y ser proporcionada diante la interrelación *GRAVITA*.

"Nombre_Tripulante" es un atributo multivaluado opcional de la entidad *NAVE*, e permite recoger los nombres de todos los tripulantes de cada nave. Se podría haber nsado en introducir una entidad *TRIPULANTE* que se interrelacionase con *NAVE*, ro no es necesario, pues de cada tripulante sólo se desea recoger su nombre. Por otro o el atributo "Misión" debe estar definido sobre el dominio {científica, militar, ɔía}.

Una *NAVE* podría estar en la misma *ÓRBITA* en distintas fechas Por tanto las chas inicial ("FI") y final ("FF") de permanencia en cada órbita son atributos ltivaluados de la interrelación **Está**.

El atributo "Circular" de la entidad *ÓRBITA* se podría haber considerado deri-do, dado que toda órbita circular tiene excentricidad 0.

La interrelación **Lanza** es ternaria y en ella se mantendrá información acerca de órbita del primer lanzamiento. Esta interrelación no es redundante, puesto que. mo en el enunciado no se establece ninguna restricción al respecto, una *NAVE* ɔdría ser lanzada por cualquier *AGENCIA* (no sólo por la agencia propietaria) y por puesto, podría lanzarse varias veces, por lo que el atributo "Fecha_Lanzamiento" es ultivaluado y provoca que la misma nave pueda tener varias órbitas iniciales.

No se puede recoger la exclusión en la financiación de los distintos tipos de agen-as.

TIPO DE COMPONENTE no es un componente específico de una NAVE: por lo, la cardinalidad es (0,N). No existe una dependencia en identificación entre *NAVE* *TIPO DE COMPONENTE*.

Transformación a Designer2000

Los pasos básicos que se deben seguir para la construcción de una base de datos ıediante la herramienta CASE Designer2000 son los siguientes:

1. Adaptación y Representación del esquema E/R mediante el módulo *Entity Relationship Diagrammer*.
2. Transformación y adaptación del diagrama E/R a un esquema relacional, mediante la utilidad *Database Design Transformer* del módulo *Design Editor*.
3. Generación y comprobación de código.

Diagrama entidad interrelación (*Entity Relationship Diagrammer*)

Los objetos permitidos en el módulo *Entity Relationship Diagrammer* son, aunque con las diferencias que comentaremos posteriormente, los mismos que el modelo teórico, es decir, entidades, interrelaciones y atributos.

Una **entidad** (tanto regular como débil) se representa mediante una caja en la que aparecen tanto el nombre de la entidad como los atributos que contiene y para cada atributo un indicador, del tipo de atributo (UID, atributo opcional u obligatorio, etc.), como se muestra en la figura 5.2.

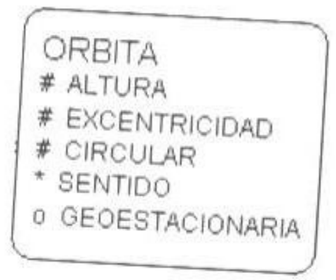


Figura 5.2. Representación de entidades

Mención especial debe hacerse a las jerarquías. El modelo E/R de Designer sólo permite las jerarquías totales y exclusivas y su representación es como se muestra en la figura 5.3.

Si nuestro esquema E/R tuviera jerarquías que no fueran de este tipo, deberían adaptarse, posiblemente considerando los subtipos de intersección elementales como entidades.

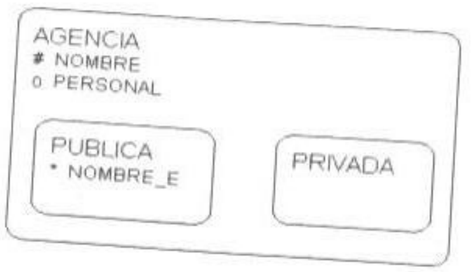


Figura 5.3. Representación de jerarquías

Para la representación de las **interrelaciones** en Designer la cardinalidad mínima mira en un extremo y la máxima en el opuesto. Además el verbo que representa la errelación en el modelo teórico, se divide en dos en la herramienta, uno en cada ntido. Así, por ejemplo, en la figura 5.4 se refleja la correspondencia entre la errelación **Lanza** del modelo teórico y la interrelación que genera Designer.

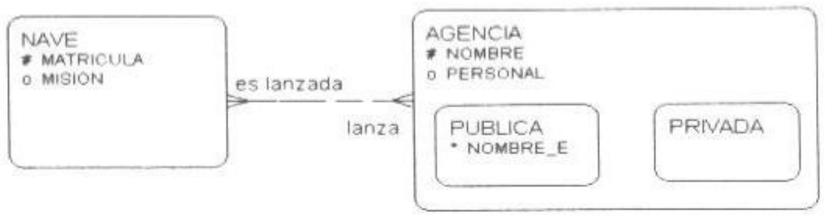


Figura 5.4. Representación de interrelaciones

Como puede apreciarse, aparecen dos verbos en el esquema "es lanzada" y "lan-'. Una *NAVE* "es lanzada" por, al menos, una agencia (línea continua en el lado de entidad *NAVE*) y como máximo "es lanzada" por *n* agencias (*pata de gallo* en la en-ad *AGENCIA*). Por otro lado, una *AGENCIA* "lanza", como mínimo, 0 naves (línea scontinua en el lado de la entidad *AGENCIA*) y como máximo de n (*pata de gallo* en entidad *NAVE*).

Para la representación de dependencias Designer utiliza como notación un rombo uado sobre la línea que une la entidad débil. Si la dependencia es una dependencia identificación, debemos transferir el UID de la entidad regular a la entidad débil y nsiderar éste como parte del UID de la entidad débil (guión vertical en la línea que e la entidad débil) como se muestra en la figura 5.5. Sin embargo, la herramienta no stingue mediante ningún tipo de notación las entidades débiles de las regulares.

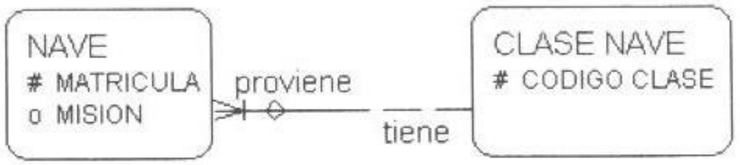


Figura 5.5. Representación de dependencias

Designer no admite la representación de interrelaciones de grado superior a dos. tampoco la representación de **atributos** multivaluados ni compuestos, tanto si estos ributos pertenecen a entidades como si pertenecen a interrelaciones.

Además, no se permite la representación de atributos en ningún tipo de interre-ción. Solamente los atributos multivaluados y los univaluados correspondientes a las terrelaciones se podrán representar posteriormente en el modelo relacional; pero si ueremos hacer constar estas características, podemos incluir un breve comentario

donde corresponda (en la entidad o en la interrelación), y así, lo podremos tener presente en los informes (*Reports*) que genera la herramienta.

Por último, mostramos el diagrama E/R completo producido para la herramienta:

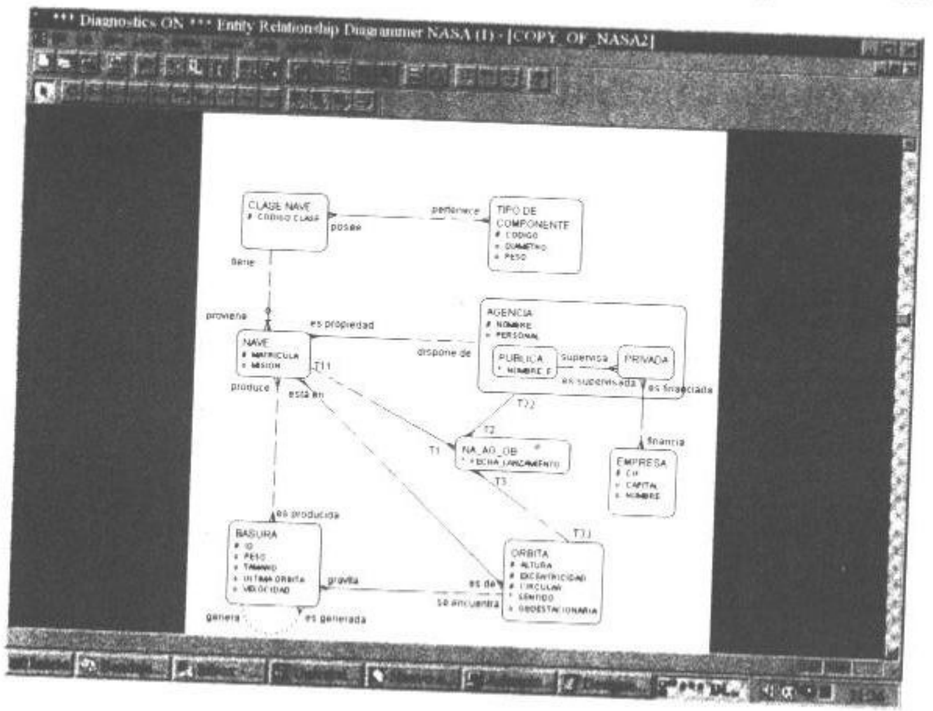


Figura 5.6. Diagrama E/R

Obsérvese cómo se ha tenido que adaptar el esquema E/R original a la notación admitida por la herramienta. Así, se ha tenido que transformar la interrelación ternaria (concepto de superior nivel de abstracción) a una entidad con tres interrelaciones (conceptos de menor nivel de abstracción).

Diseño de la base de datos (*Database Design*)

Tras realizar el diagrama entidad interrelación y transformarlo mediante la utilidad *Database Design Transformer*, podemos obtener el equivalente al modelo relacional de Codd. Las entidades se transforman en relaciones, las interrelaciones 1:N se transforman mediante la propagación de clave y las interrelaciones N:M dan como resultado una nueva relación cuya clave primaria resulta de la concatenación de las claves primarias de las relaciones que intervienen.

Como se comentó en el apartado anterior, el único tipo de jerarquía que se permite en esta herramienta CASE es la jerarquía total y exclusiva. La transformación de este tipo de jerarquías se hace, por defecto, considerando una única relación para el supertipo, y añadiendo a esta relación un atributo discriminante. Cuando se diseña una

base de datos, a menudo se necesita, como en el caso que nos ocupa, tener relaciones tanto para el supertipo como para los subtipos, pues éstos están relacionados con otras entidades. Si hacemos que Designer genere relaciones para todas las entidades que intervienen en la jerarquía, conseguiremos también tantas interrelaciones para los subtipos como interrelaciones tenga el supertipo, por lo que ya en el modelo relacional habrá que retocarlas manualmente. Además, no se creará el atributo discriminante en el supertipo y, en caso de necesitarlo, habrá que introducirlo posteriormente.

En el diagrama E/R teórico propuesto para este ejercicio, podemos apreciar que las interrelaciones **Genera** y **Financia** poseen cada una un atributo que no pudimos modelar en el módulo *Entity Relationship Diagrammer*. En el caso de la interrelación **Financia** y al haberse convertido en una relación (su cardinalidad era N:M), únicamente debemos crear el atributo "Porcentaje" dentro de la relación (Figura 5.7). La interrelación **Genera**, al tener cardinalidad 1:N y ser una interrelación reflexiva, se habría transformado mediante la propagación de la clave de la relación BASURA a sí misma; para introducir el atributo "Fecha" debemos construir una nueva relación GENERA_BASURA e introducir los atributos que forman la clave primaria, definiendo ésta, y el atributo "Fecha" como aparece en la figura 5.8.

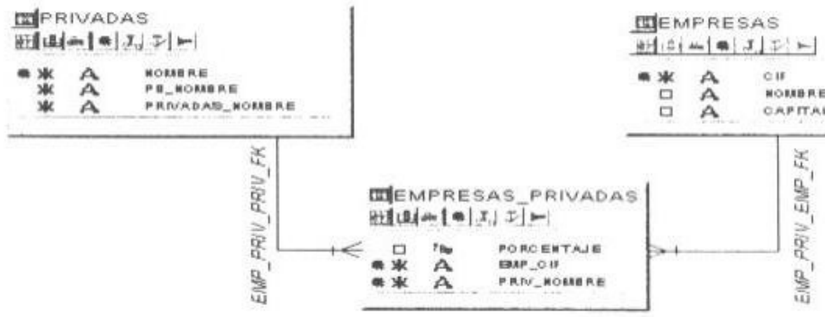


Figura 5.7. Inclusión del atributo "Porcentaje"

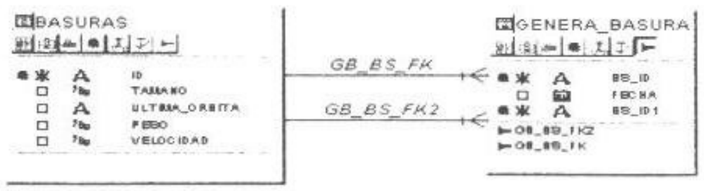


Figura 5.8. Inclusión del atributo "Fecha"

Alternativamente, se podría haber considerado la propagación de clave hacia la relación BASURA, lo que conllevaría la inclusión del atributo "Fecha" en la misma. Esta solución, aun siendo igualmente válida, provocaría la introducción de gran cantidad de valores nulos, pues la basura espacial se puede o no fragmentar.

Siguiendo con la transformación de los atributos, aquellos que son multivaluados se tratan igual a como se haría en el modelo relacional teórico. Así, por ejemplo, en el caso del atributo "Nombre_Tripulante" de la entidad *NAVE*, se construye una nueva relación TRIPULANTES, cuya clave primaria estará compuesta por la clave primaria de la relación NAVE y el nombre del tripulante (figura 5.9).

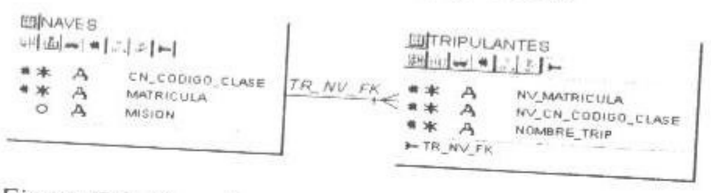


Figura 5.9. Transformación de atributos multivaluados

Para completar el modelo podemos definir las opciones de borrado y modificación que estimemos oportunas, así como los disparadores y *checks* necesarios para controlar la integridad de la base de datos.

La figura 5.10 muestra el esquema relacional completo que genera la herramienta.

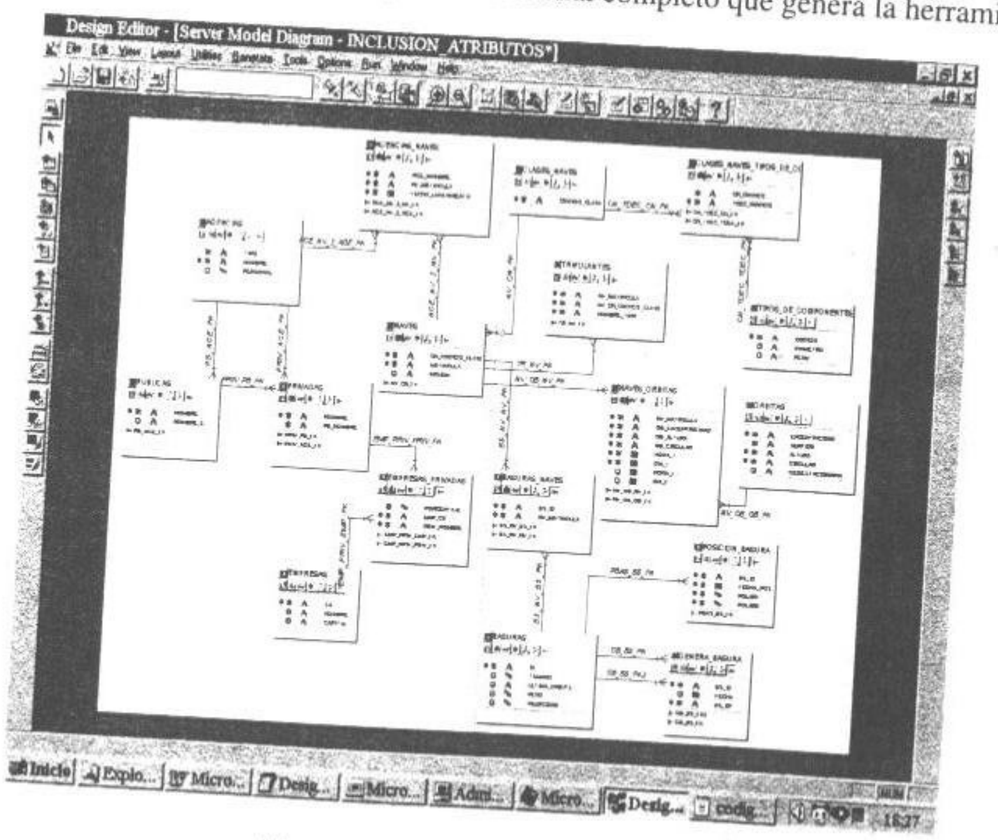


Figura 5.10. Esquema relacional

Generación y comprobación de código

Una vez se ha construido un diseño de base de datos ajustado a nuestras necesidades, se puede proceder a la generación automática de código, que puede ser generado directamente contra el SGBD o bien ser implementado mediante cuatro tipos de ficheros: el primero, *.tab, contiene el código para la creación de las tablas, en él se excluye la definición de las claves primarias y ajenas, como se muestra en la figura 5.11.

codigo.tab - Bloc de notas

```
-- d:\libro\casos completos\nasa\codigo.tab
--
-- Generated for Oracle 8 on Thu Jul 13  13:49:43 2000 by Server Generator 2.1.19.5.0


PROMPT Creating Table 'ORBITAS'
CREATE TABLE ORBITAS
 (EXCENTRICIDAD VARCHAR2(240) NOT NULL
 ,SENTIDO VARCHAR2(240) NOT NULL
 ,ALTURA VARCHAR2(240) NOT NULL
 ,CIRCULAR VARCHAR2(240) NOT NULL
 ,GEOESTACIONARIA VARCHAR2(240)
 )
/

PROMPT Creating Table 'CLASES_NAVES_TIPOS_DE_COMPONEN'
CREATE TABLE CLASES_NAVES_TIPOS_DE_COMPONEN
 (CN_CODIGO VARCHAR2(240) NOT NULL
 ,TDEC_CODIGO VARCHAR2(240) NOT NULL
 )
/

PROMPT Creating Table 'AGENCIAS'
CREATE TABLE AGENCIAS
 (TIPO VARCHAR2(240) NOT NULL
```

Figura 5.11. Generación de tablas

El segundo fichero, *.con contiene la definición de las restricciones (*constraints*). así como los disparadores, *checks*, etc., definidos en el diseño de la base de datos.

codigo.con - Bloc de notas

```
-- d:\libro\casos completos\nasa\codigo.con
--
-- Generated for Oracle 8 on Thu Jul 13  13:49:44 2000 by Server Generator 2.1.19.5.0

PROMPT Creating Primary Key on 'ORBITAS'
ALTER TABLE ORBITAS
 ADD CONSTRAINT OB_PK PRIMARY KEY
  (EXCENTRICIDAD
  ,ALTURA
  ,CIRCULAR)
/

PROMPT Creating Primary Key on 'CLASES_NAVES_TIPOS_DE_COMPONEN'
ALTER TABLE CLASES_NAVES_TIPOS_DE_COMPONEN
 ADD CONSTRAINT CN_TDEC_PK PRIMARY KEY
  (TDEC_CODIGO)
/

PROMPT Creating Primary Key on 'AGENCIAS'
ALTER TABLE AGENCIAS
```

Figura 5.12. Generación de restricciones

El tercer fichero, *.ind contiene el código necesario para generar los índices sobre determinadas columnas.

codigo.ind - Bloc de notas

```
-- d:\libro\casos completos\nasa\codigo.ind
--
-- Generated for Oracle 8 on Thu Jul 13  13:49:44 2000 by Server Generator 2.1.19.5.0


PROMPT Creating Index 'CN_TDEC_CN_FK_I'
CREATE INDEX CN_TDEC_CN_FK_I ON CLASES_NAVES_TIPOS_DE_COMPONEN
 (CN_CODIGO)
/

PROMPT Creating Index 'CN_TDEC_TDEC_FK_I'
CREATE INDEX CN_TDEC_TDEC_FK_I ON CLASES_NAVES_TIPOS_DE_COMPONEN
 (TDEC_CODIGO)
/

PROMPT Creating Index 'AGE_MV_2_AGE_FK_I'
CREATE INDEX AGE_MV_2_AGE_FK_I ON AGENCIAS_NAVES
 (AGE_NOMBRE)
/

PROMPT Creating Index 'AGE_MV_2_MV_FK_I'
CREATE INDEX AGE_MV_2_MV_FK_I ON AGENCIAS_NAVES
 (MV_MATRICULA)
```

Figura 5.13. Generación de índices

Por último, el fichero *.sql es el que realiza el *spool* de los ficheros anteriores para crear los objetos en la base de datos (Figura 5.14), y estructura la ejecución de los otros tres.

codigo.sql - Bloc de notas

```
-- d:\libro\casos completos\nasa\codigo.sql
--
-- Generated for Oracle 8 on Thu Jul 13  13:49:45 2000 by Server Generator 2.1.19.5.0


SPOOL codigo.lst

@@ codigo.tab
@@ codigo.ind
@@ codigo.con

SPOOL OFF
```

Figura 5.14. Generación del fichero de spool

CASO PRÁCTICO 2: SEGUROS DE AUTOMÓVIL

Enunciado

La entrada en vigor a escala europea en el sector de seguros del nuevo modelo de documento "Declaración amistosa de accidente de automóvil" para la resolución simplificada de trámites entre compañías de seguros, hace que una compañía

eguradora de vehículos se plantee la necesidad de un nuevo sistema de información ue le soporte. Como primera medida, se ha realizado un estudio de este documento ue se presenta en la figura 5.15[30]) y una serie de entrevistas a usuarios. Las con-usiones obtenidas y necesarias aclaraciones que se utilizarán como guía para la aboración del esquema de datos, son las siguientes:

Este documento lo recibe la compañía aseguradora sólo en aquellos accidentes de áfico que afectan directamente a dos vehículos[31], denominados genéricamente A y B n los cuales al menos uno de ellos está asegurado por la compañía. Se entiende que el ocumento lo recibe de cada vehículo, con independencia de que lo presente el onductor, el asegurado o cualquier otra persona. Así, para el mismo accidente, la ompañía aseguradora podrá recibir, en el caso de que ambos vehículos estén segurados por ella, un máximo de dos de estos documentos (uno por el vehículo A y ro por el vehículo B).

Aunque los dos documentos correspondientes al mismo accidente serán xactamente iguales (ya que proceden de papel autocopiativo), por motivos de eguridad cada uno de ellos se digitalizará, almacenándose en un fichero de imagen. omo el mismo vehículo, podrá, a lo largo del tiempo, presentar distintos documentos para distintos accidentes) se registrará la fecha de presentación. Cada uno de estos ocumentos se referirá exactamente a dos vehículos (el A y el B), y para cada uno de los, refiere las circunstancias que concurren en el accidente, los daños apreciados y bservaciones.

Las circunstancias tendrán un código y una descripción. Compruebe cómo el dato el número de casillas marcadas, se emplea únicamente con propósitos de seguridad a n de evitar manipulaciones posteriores del documento.

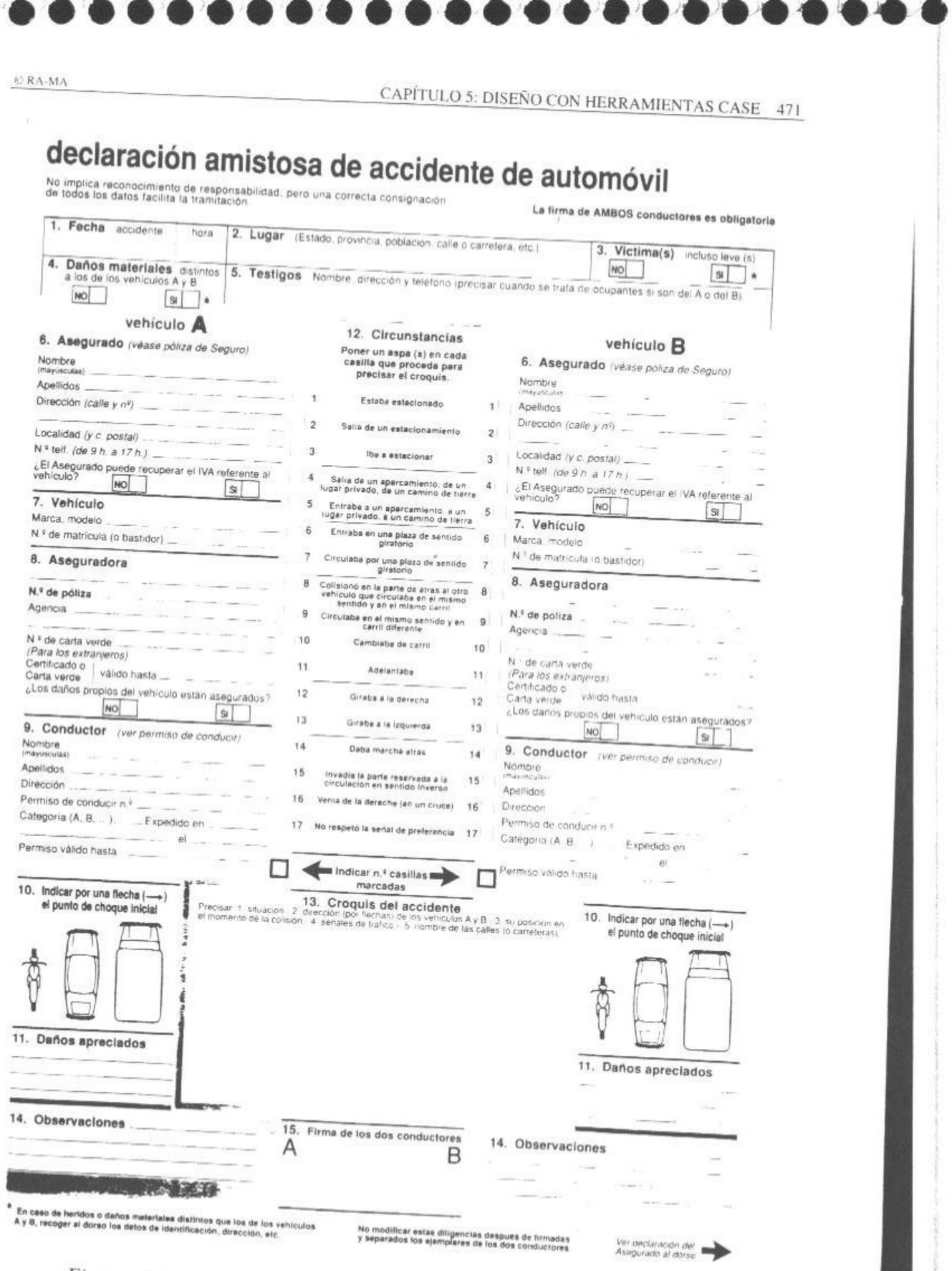
Como se puede observar en el documento, cada vehículo se identificará por el úmero de matrícula o en su defecto por el número de bastidor, y tendrá como datos sociados la marca y el modelo.

Cada accidente se identifica mediante un número que se asignará al tiempo que se raban los documentos[32]. En el anexo presentado pueden verse los datos del ormulario de "Declaración amistosa de accidente de automóvil". De entre estos datos, l croquis, las firmas de los conductores y los puntos de choque inicial se recogerán omo imágenes, pues está previsto que en caso necesario su consulta se hará irectamente sobre la imagen del documento. Además, para el punto de choque inicial ambién se guardará una descripción (por ejemplo, golpe en puerta derecha delantera, otura de parabrisas, etc.)

[30] Reproducido con el permiso del CEA *(Comité Européen des Assurances)*.

[31] Para el tratamiento de otros tipos de accidente hay otros tipos de documento cuyo tratamiento ueda fuera del presente sistema.

[32] A fin de evitar duplicidades en la asignación de estos números, en la grabación de los documentos e requiere un proceso de confrontación de los documentos recibidos relativos al mismo accidente.



declaración amistosa de accidente de automóvil

No implica reconocimiento de responsabilidad, pero una correcta consignación de todos los datos facilita la tramitación

La firma de AMBOS conductores es obligatoria

1. Fecha accidente hora
2. Lugar (Estado, provincia, población, calle o carretera, etc.)
3. Víctima(s) incluso leve (s) NO SI *
4. Daños materiales distintos a los de los vehículos A y B NO SI *
5. Testigos Nombre, dirección y teléfono (precisar cuando se trata de ocupantes si son del A o del B)

vehículo A

6. Asegurado (véase póliza de Seguro)
Nombre (mayúsculas)
Apellidos
Dirección (calle y nº)
Localidad (y c. postal)
Nº telf. (de 9 h. a 17 h.)
¿El Asegurado puede recuperar el IVA referente al vehículo? NO SI
7. Vehículo
Marca, modelo
Nº de matrícula (o bastidor)
8. Aseguradora
Nº de póliza
Agencia
Nº de carta verde (Para los extranjeros)
Certificado o Carta verde válido hasta
¿Los daños propios del vehículo están asegurados? NO SI
9. Conductor (ver permiso de conducir)
Nombre (mayúsculas)
Apellidos
Dirección
Permiso de conducir nº
Categoría (A, B, ...). Expedido en el
Permiso válido hasta
10. Indicar por una flecha (→) el punto de choque inicial
11. Daños apreciados
14. Observaciones

12. Circunstancias
Poner un aspa (x) en cada casilla que proceda para precisar el croquis.

1	Estaba estacionado	1	10	Cambiaba de carril	10
2	Salía de un estacionamiento	2	11	Adelantaba	11
3	Iba a estacionar	3	12	Giraba a la derecha	12
4	Salía de un aparcamiento, de un lugar privado, de un camino de tierra	4	13	Giraba a la izquierda	13
5	Entraba a un aparcamiento, a un lugar privado, a un camino de tierra	5	14	Daba marcha atrás	14
6	Entraba en una plaza de sentido giratorio	6	15	Invadía la parte reservada a la circulación en sentido inverso	15
7	Circulaba por una plaza de sentido giratorio	7	16	Venía de la derecha (en un cruce)	16
8	Colisionó en la parte de atrás al otro vehículo que circulaba en el mismo sentido y en el mismo carril	8	17	No respetó la señal de preferencia	17
9	Circulaba en el mismo sentido y en carril diferente	9			

Indicar nº casillas marcadas

13. Croquis del accidente
Precisar: 1. situación, 2. dirección (por flechas) de los vehículos A y B, 3. su posición en el momento de la colisión, 4. señales de tráfico, 5. nombre de las calles (o carreteras).

15. Firma de los dos conductores
A B

vehículo B

6. Asegurado (véase póliza de Seguro)
Nombre (mayúsculas)
Apellidos
Dirección (calle y nº)
Localidad (y c. postal)
Nº telf. (de 9 h. a 17 h.)
¿El Asegurado puede recuperar el IVA referente al vehículo? NO SI
7. Vehículo
Marca, modelo
Nº de matrícula (o bastidor)
8. Aseguradora
Nº de póliza
Agencia
Nº de carta verde (Para los extranjeros)
Certificado o Carta verde válido hasta
¿Los daños propios del vehículo están asegurados? NO SI
9. Conductor (ver permiso de conducir)
Nombre (mayúsculas)
Apellidos
Dirección
Permiso de conducir nº
Categoría (A, B, ...). Expedido en el
Permiso válido hasta
10. Indicar por una flecha (→) el punto de choque inicial
11. Daños apreciados
14. Observaciones

* En caso de heridos o daños materiales distintos que los de los vehículos A y B, recoger al dorso los datos de identificación, dirección, etc.

No modificar estas diligencias después de firmadas y separados los ejemplares de los dos conductores

Ver declaración del Asegurado al dorso

Figura. 5.15. Formulario de declaración amistosa de accidentes

A propósito de los datos del accidente, pueden hacerse algunas observaciones de erés, como el que un accidente implica a dos tipos de personas, cada uno de ellos alizando un papel diferente (asegurado o conductor)[33] aun cuando ambos papeles drían coincidir en la misma persona. Por este motivo, se considera conveniente stinguir entre éstos los papeles que una persona implicada en el accidente puede mar (suponemos que un nº de póliza identifica a un asegurado). Se registrará además vehículo que bien conducen o bien aseguran (hay que tener en cuenta que un mismo hículo puede cambiar tanto de conductor como de asegurado en los distintos cidentes que pueda tener).

Todo conductor debería tener un permiso de conducción que le faculta para nducir diferentes tipos de vehículos agrupados en categorías que se identifican rectamente mediante un código de categoría.

Aunque no aparezca en el formulario, de las aseguradoras de los vehículos se ardará el nombre, la sede social y un número de teléfono de contacto. Los datos erca del seguro pueden tomarse directamente del documento.

squema E/R

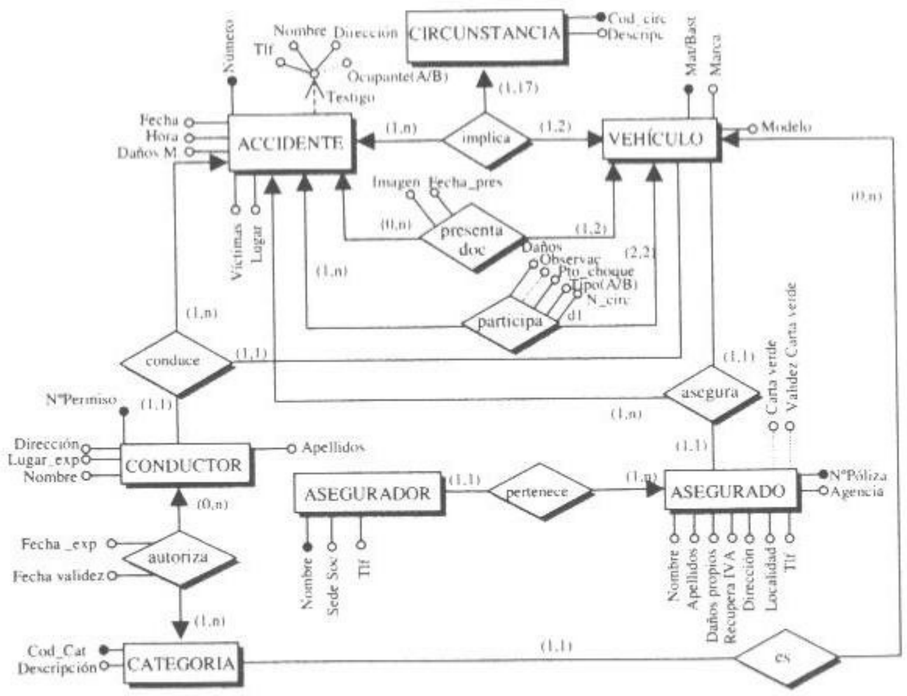


Figura 5.16. Esquema E/R

[33] Una persona implicada en un accidente podría serlo también en el papel de víctima, incluso sin star relacionada con ninguno de los dos vehículos (por ejemplo, se trataría de un peatón), aunque dado ue esta posibilidad se contempla tan sólo en el dorso del documento y que por simplicidad tan sólo se resenta el anverso de éste, no consideraremos dentro del sistema los datos de las víctimas.

Consideraciones sobre la solución propuesta

Aunque se dice en el enunciado, conviene subrayar que se está realizando el sistema de información de una única compañía aseguradora, por ello solamente se recogerán datos de aquellos accidentes de tráfico en que uno de los vehículos afectados está asegurado por esta compañía. Además, el tipo de accidentes que se considera es uno de los muchos tipos posibles, en el que sólo hay dos vehículos afectados y ambos conductores acuerdan la presentación de este documento.

Por simplificar, se ha eliminado el reverso del documento, que contiene una serie de datos que cada conductor debería rellenar antes de presentar su copia del documento. La presencia de estos datos invalidan cualquier intento de emplear la solución obtenida como base del sistema de información de una compañía aseguradora que maneje "declaraciones amistosas".

En la solución propuesta no aparece ninguna entidad *DOCUMENTO*. Aunque podría incluirse, se ha decidido no hacerlo debido a que el concepto de documento está íntimamente relacionado con el de accidente. De hecho, un documento es el reflejo de un accidente para la compañía aseguradora. Trabajando con el concepto de documento se primaría la recogida de información administrativa, mientras que con el concepto de accidente se da más importancia al hecho en sí que desencadena la actuación de la aseguradora. Podríamos, incluso, manteniendo ambos conceptos, abordar las dos perspectivas del problema. En la solución elegida, toda la información relativa al documento presentado se almacenará en la interrelación binaria **Presenta_Doc** entre vehículo y accidente.

Veamos ahora las cardinalidades de la interrelación **Presenta_Doc**. Cada ocurrencia de esta interrelación representa la presentación de un documento referido a un accidente. Puesto que para un accidente dado podrán presentarse uno o dos documentos (serán exactamente dos en el caso de que los dos vehículos estén asegurados por nuestra compañía), la cardinalidad será (1,2). Además, un vehículo podrá presentar a lo largo del tiempo tantos documentos como accidentes haya tenido, y como mínimo no habrá presentado ninguno, dado que puede tratarse de un vehículo que no esté asegurado por esta compañía pero que se haya visto afectado en un accidente con otro vehículo que sí sea de esta compañía, cardinalidad (0,N).

Como vemos en la interrelación **Participa**, en un accidente participan exactamente dos vehículos, cardinalidad (2,2), y el mismo vehículo resultará afectado tantas veces como accidentes haya tenido, y como mínimo habrá resultado afectado por un accidente, dado que en otro caso la información de este vehículo no se encontraría en la BD, cardinalidad (1,N).

La semántica reflejada en la interrelación ternaria **Implica** será que por cada vehículo afectado en un accidente concurren como mínimo una y como máximo diecisiete circunstancias. Además, cada circunstancia y vehículo pueden aparecer en

n accidente como mínimo y en *n* como máximo (en diferentes accidentes el mismo ehículo puede repetir las mismas circunstancias), por último, cada circunstancia roducida en un accidente concreto afecta a uno o a lo sumo dos vehículos (depen-iendo de que los dos vehículos compartan o no la misma circunstancia). No sería orrecto incluir en esta interrelación atributos relativos a un vehículo en un accidente, omo "Daños", "Observaciones", etc., ya que en ese caso se estarían repitiendo por ada circunstancia. Sólo se deberían recoger una vez por cada vehículo afectado por n accidente.

La semántica recogida por las dos interrelaciones binarias **Presenta_Doc** y **Par-cipa** podría recogerse en una única, que contuviera los atributos de ambas. En ese aso, la imagen del documento y la fecha de presentación del mismo tomarían valor ulo en el caso de los vehículos no asegurados por nuestra compañía. Se ha dividido n dos porque se ha considerado que el esquema queda más claro, aunque la solución lternativa sería igualmente válida (los controles a realizar serían distintos).

El atributo "Nº_Circ" de la interrelación **Participa** es un atributo derivado que ecoge el número de circunstancias para cada vehículo en cada accidente. Obsérvese ue, aunque pudiera pensarse, por hacer referencia a las circunstancias que concurren n un accidente, podría pertenecer a la interrelación **Implica**, si se hubiera hecho así, e repetiría para cada circunstancia. Lo mismo ocurriría para el resto de atributos de sta interrelación.

"Testigo" es un atributo compuesto, multivaluado y opcional que pertenece onceptualmente a la entidad *ACCIDENTE*. Uno de los componentes de este atributo ecoge de qué vehículo era ocupante el testigo. Esto podría hacernos pensar incluirlo n la interrelación **Participa**, que relaciona las entidades *ACCIDENTE* y *VEHÍCULO*. n este caso, habría que complicar el diseño decidiendo dónde incluir los testigos que o son ocupantes de ningún vehículo (una posibilidad sería situar los testigos ocu-antes de algún vehículo en **Participa** y el resto en *ACCIDENTE*).

La identificación de cada vehículo puede realizarse mediante el número de matrí-ula o el número de bastidor. Consideraremos que el dominio del atributo "Mat_Bast" e la entidad *VEHÍCULO* tiene valores de los dos tipos y que éstos no coinciden.

Los accidentes tienen un atributo propio, "Número", que les identifica. En el nunciado se nos informa del procedimiento que se ha de realizar para evitar que el ismo accidente aparezca por duplicado o que diferentes accidentes se confundan en no solo. Este procedimiento es un aspecto funcional que no influye en la obtención el esquema.

La entidad *CONDUCTOR* recoge los datos de los conductores, mientras que la ntidad *ASEGURADO* recogerá los de los asegurados. Podría pensarse además en una ntidad *IMPLICADO* que generalizara a las dos anteriores y que en la posterior trans-ormación al relacional evitaría la repetición de datos cuando coincidan el conductor y

el asegurado. Sin embargo, para que se tratara de una auténtica generalización, tanto el supertipo como los subtipos deberían identificarse mediante el mismo atributo identificador y no es el caso, por lo que habría de incorporarse un atributo que hiciera factible esta generalización. Además, desde un punto de vista conceptual esta generalización no es estrictamente necesaria.

Serán atributos de la entidad *CONDUCTOR*, el "Nombre", "Apellidos", "Dirección", "Número de Permiso" (atributo identificador principal, ya que, como se afirma en el enunciado, todo conductor debe tener un permiso de conducir, luego no es posible que haya conductores sin permiso de conducir, o al menos, conductores que hayan rellenado este tipo de documento) y "Lugar_Expedición" (lugar de expedición del único permiso de conducción de un conductor, aunque este permiso autorice varias categorías, cada una de ellas con su propio período de validez).

Entre las entidades *CONDUCTOR*, *ACCIDENTE* y *VEHÍCULO* se da una interrelación ternaria **Conduce**. Aunque en esta interrelación podrán figurar varios accidentes en los que un conductor conducía el mismo vehículo, no habrá más de dos ejemplares por cada accidente, uno correspondiente al vehículo A y su conductor y otro al vehículo B y su respectivo conductor; no es posible recoger esta última restricción dentro del esquema E/R con esta solución. Como veremos después, en el esquema ERwin sí será posible, ya que las cardinalidades de las interrelaciones ternarias recogen distinta semántica.

De forma similar, en el caso de los asegurados tenemos una interrelación ternaria **Asegura** entre las entidades *VEHÍCULO*, *ACCIDENTE* y *ASEGURADO*. El atributo "Número_Póliza", por ser único para cada asegurado, será identificador de la entidad. Aunque no incluiremos una entidad *PÓLIZA* en la solución, obsérvese que posiblemente la compañía aseguradora tendrá en algún lugar (una BD o un fichero) la información asociada a sus pólizas, por lo que, en el caso de que cualquiera de los vehículos esté asegurado por ésta, tendrá disponibles de manera inmediata todos los datos asociados a la póliza.

Entre las interrelaciones existirán algunas dependencias de inclusión. Por ejemplo, los pares "Accidente-Vehículo" de las interrelaciones **Conduce**, **Asegura**, **Presenta_Doc** e **Implica** estarán incluidos en la interrelación **Participa**.

Se tiene una entidad *CATEGORÍA* con la que la entidad *CONDUCTOR* mantiene una interrelación N:M **Autoriza**, y dado que cada categoría autorizada a un conductor en un permiso de conducción puede tener su propia fecha de expedición y fecha de validez, se tendrá que esta interrelación **Autoriza** tiene como atributos "Fecha_Expedición" y "Fecha_Validez".

Como hipótesis adicional al enunciado puede suponerse a efectos de control de la aseguradora una interrelación, **Es**, entre los vehículos y las categorías, de manera que para cada vehículo se disponga su categoría y poder así contrastar si un conductor

nducía un vehículo de una categoría para la cual no estuviera autorizado, aunque los mplares de esta interrelación provendrían de una fuente ajena al documento, es cir, la propia compañía aseguradora debería averiguar la categoría de cada vehículo.

La entidad *ASEGURADORA* mantiene una interrelación 1:N **Pertenece** con la tidad *ASEGURADO*. Esta entidad tendrá los atributos "Nombre" (atributo identifidor principal), "Daños_Propios", "Agencia", "Carta_Verde", y "Fecha_Validez". tos tres últimos atributos son opcionales.

ransformación al modelo lógico soportado en la herramienta Rwin

La herramienta ERwin permite trabajar con modelos lógicos y físicos para la presentación de los esquemas relacionales y físicos, respectivamente. Soporta la tación de las metodologías Information Engineering e IDEF1X (en el ejemplo que remos a continuación, se utilizará la notación de IDEF1X). Es importante destacar e ERwin no soporta la fase de diseño conceptual (no es posible representar esemas E/R); sin embargo, el modelo lógico es una combinación de constructores E/R oncretamente las interrelaciones N:M, dependencias en existencia jerarquías) y ementos del modelo relacional. Aquí se considerará que un esquema lógico es una presentación del grafo relacional. Veamos a continuación los elementos del modelo gico soportado en esta herramienta.

En el modelo lógico se distinguen dos tipos de entidades: dependientes e independientes, en función de cómo adquieren sus claves. Las entidades independientes no ependen de ninguna otra entidad para su identificación, y se representan por una caja esquinas cuadradas. Las entidades dependientes dependen de una o más entidades ra su identificación, y se representan por una caja de esquinas redondeadas. En anto a las interrelaciones, únicamente se admiten las interrelaciones binarias. xisten interrelaciones en identificación (*identifying relationship*), si necesitan incluir clave primaria de la entidad padre como parte de la clave primaria de la hija para su entificación, e interrelaciones en no identificación (*non identifying relationship*) que onectan una entidad padre con la entidad hija de talforma que los atributos de la clave ena de las relaciones están situadas en el área de datos y no en el área de las claves s decir, la clave ajena que se propaga no pasa a formar parte de la clave primaria de entidad hija).

También existen interrelaciones N:M (llamadas *many-to-many*) en el modelo gico que se transforman al modelo físico incluyendo una entidad de enlace que se laciona mediante dos interrelaciones en identificación con las entidades que forman interrelación origen. Sin embargo, debido a que puede ocurrir que las interlaciones N:M tengan atributos, optaremos por descomponer directamente en el odelo lógico la interrelación en dos interrelaciones 1:N empleando una entidad de

enlace (al igual que se ha visto para la transformación de este tipo de interrelaciones en la construcción del grafo relacional en el capítulo 2).

En principio, cada entidad la transformaremos en una entidad independiente en ERwin. Por tanto, como podemos observar en la figura 5.17, tendremos como entidades independientes Circunstancia, Vehículo, Accidente, Conductor, Categoría, Asegurado y Aseguradora.

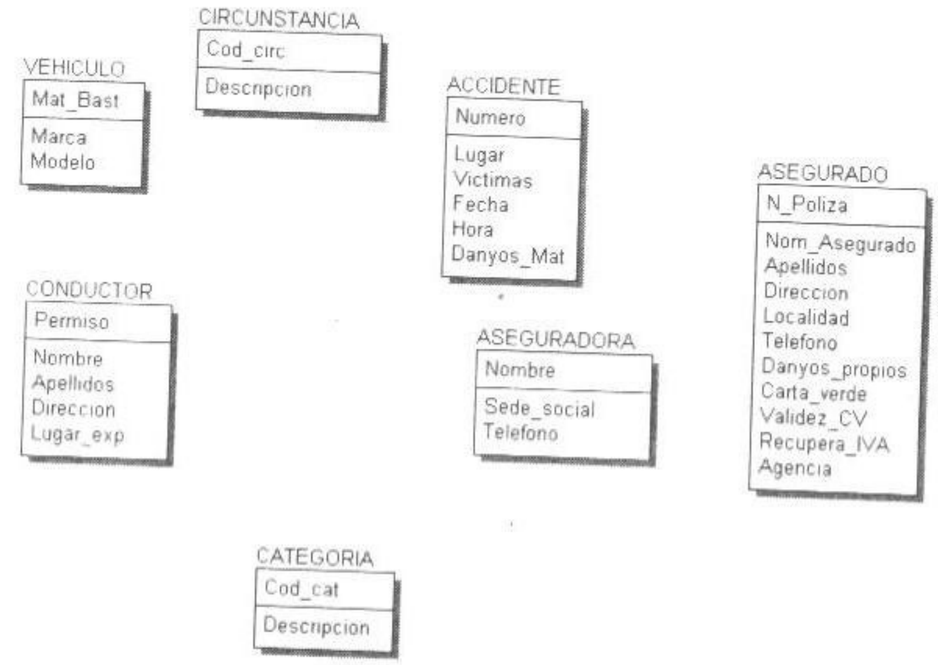


Figura 5.17. Entidades independientes

Las interrelaciones 1:N sin atributos propios las transformamos en ERwin mediante una relación en no identificación, que automáticamente propaga la clave correspondiente a la cardinalidad máxima 1 a la entidad con cardinalidad máxima N como clave ajena. Por ejemplo, la figura 5.18 muestra la transformación de la interrelación **Es** entre *CATEGORÍA* y *VEHÍCULO*. La línea punteada indica que es una relación en no identificación, y el código de categoría propagado como clave ajena se indica con las siglas FK (*Foreign Key*).

Figura 5.18. Transformación de la interrelación Es

De forma análoga al caso anterior, transformaríamos la interrelación **Pertenece** entre *ASEGURADORA* y *ASEGURADO* (figura 5.19).

Figura 5.19. Transformación de la interrelación Pertenece

Como se ha mencionado anteriormente, al no permitir ERwin la inclusión de atributos en las interrelaciones binarias las representaremos mediante una asociación de entidades, es decir, una entidad dependiente que contendrá como clave las claves ajenas de las entidades que relaciona.

Por ejemplo, la interrelación **Autoriza** entre *CONDUCTOR* y *CATEGORÍA* se transformaría en una entidad dependiente, y dos relaciones en identificación que propagarían las claves a la nueva entidad, a la que le añadiríamos los atributos propios de la interrelación. Aunque posteriormente hablaremos con más detalle de las cardinalidades en ERwin, la participación mínima *1* de *CONDUCTOR* en **Autoriza** se indica mediante una *P*, como puede verse en la figura 5.20.

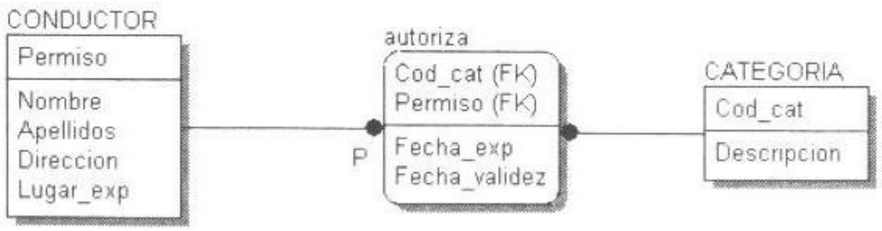


Figura 5.20. Transformación de interrelación binaria con atributos

De igual forma, las interrelaciones **Presenta_Doc** y **Participa**, ambas con atributos propios, se traducirían mediante dos entidades dependientes, como puede verse en la figura 5.21. La participación mínima *1* de *ACCIDENTE* en **Presenta_Doc** y *VEHÍCULO* en **Participa** se indica con una *P*. Nótese que mientras la participación

máxima y mínima 2 de *ACCIDENTE* en **Participa** se indica con este número en la relación correspondiente, su participación máxima (también 2) en **Presenta_Doc** no puede indicarse en el esquema, ya que las únicas cardinalidades permitidas son (0,n), (1,n), (0,1) y (k,k), para un *k* fijo.

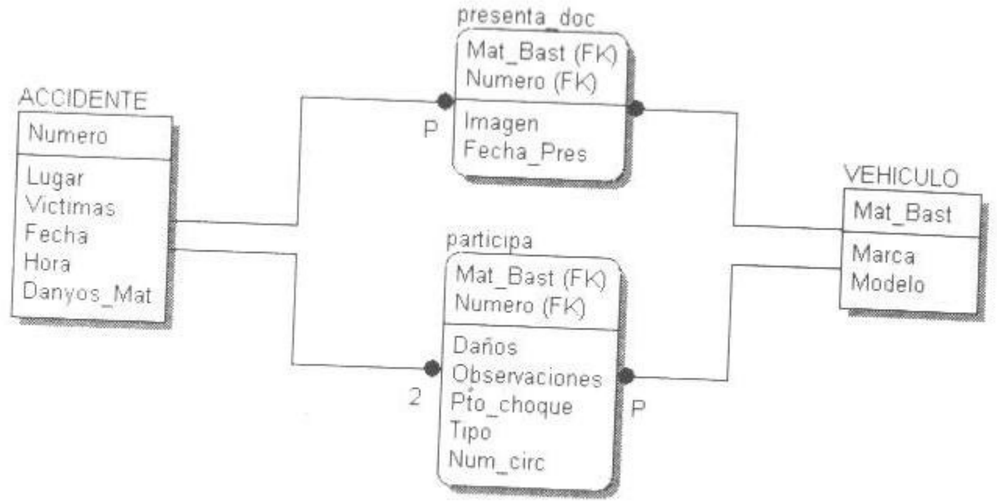


Figura 5.21. Transformación de cardinalidades

ERwin no admite atributos multivaluados, por lo que para transformar el atributo multivaluado "Testigo", crearemos una entidad que necesitará para su identificación de la clave de accidente, además del nombre del testigo (suponemos para ello que en un mismo accidente no habrá dos testigos con el mismo nombre). Como un accidente puede no tener testigos, no todas las ocurrencias de *ACCIDENTE* tendrán por qué participar en la relación, por lo que su cardinalidad mínima será 0, lo cual se refleja por la ausencia de indicador en el extremo de la relación correspondiente a *TESTIGO* (figura 5.22).

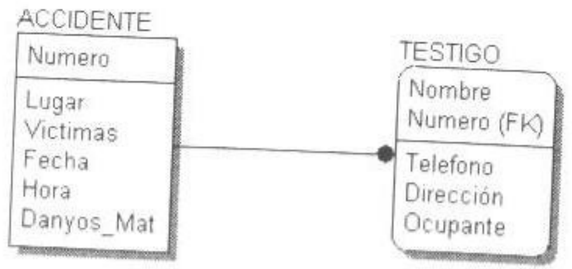


Figura 5.22. Transformación de un atributo multivaluado

Las interrelaciones ternarias las transformaremos de forma similar a como hacíamos con las binarias con atributos propios. Sin embargo, al no ser el significado de las cardinalidades de ERwin las mismas que las que utilizábamos en el modelo ER,

ara la transformación será necesario revisar la semántica del problema. Mientras que n el ER la cardinalidad reflejaba, dados un par de ejemplares de dos entidades articipantes en una interrelación con cuántos ejemplares de la entidad restante se elacionaban, en el caso de ERwin refleja el número de veces que un mismo ejemplar e una entidad participa en una relación. En el caso de las relaciones binarias el ignificado es equivalente (aunque se colocan en el lado opuesto de la relación), pero n las ternarias, como acabamos de decir, es necesario revisar la semántica del problema.

Por ejemplo, es claro que cada accidente debería participar exactamente dos eces en las interrelaciones ternarias **Conduce** y **Asegura** (una por cada vehículo). En l esquema ER mostrado en la figura 5.16 esta semántica no puede reflejarse debido al istinto significado de las cardinalidades. Sin embargo, en el caso de ERwin, sí se uede reflejar esta semántica, como puede observarse en la figura 5.23, que refleja la ransformación de las tres interrelaciones ternarias.

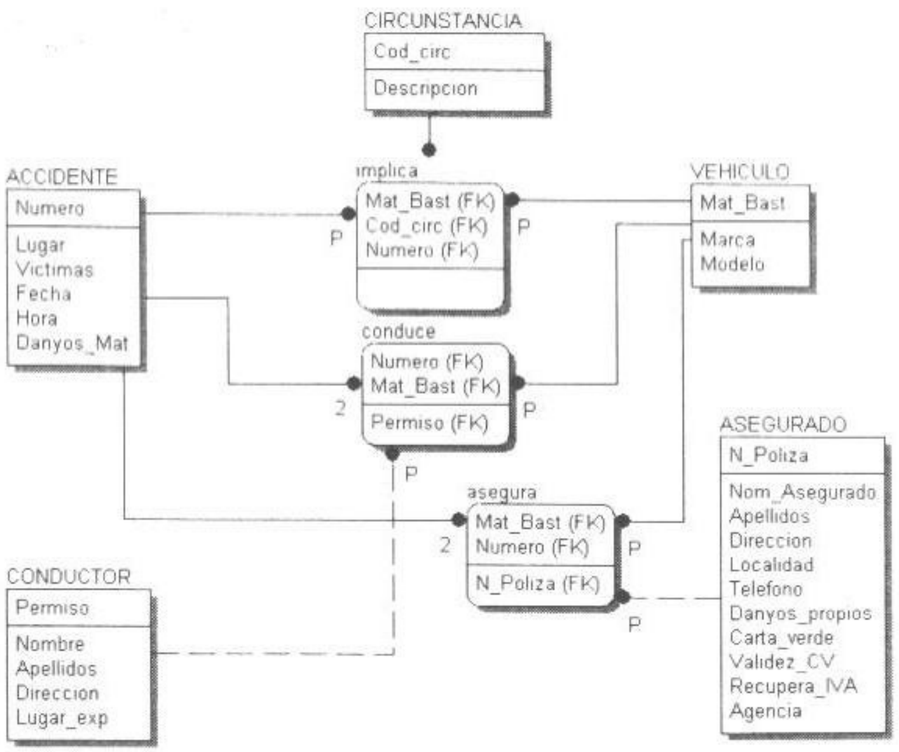


Figura 5.23. Transformación de interrelaciones ternarias

ERwin no sólo permite describir la participación de cada entidad en cada elación, sino también así como la definición de dominios, opciones de borrado, nserción y modificación, y otras muchas características en las que no profundizaremos. La figura 5.23 muestra el entorno de ERwin con el diseño lógico completo.

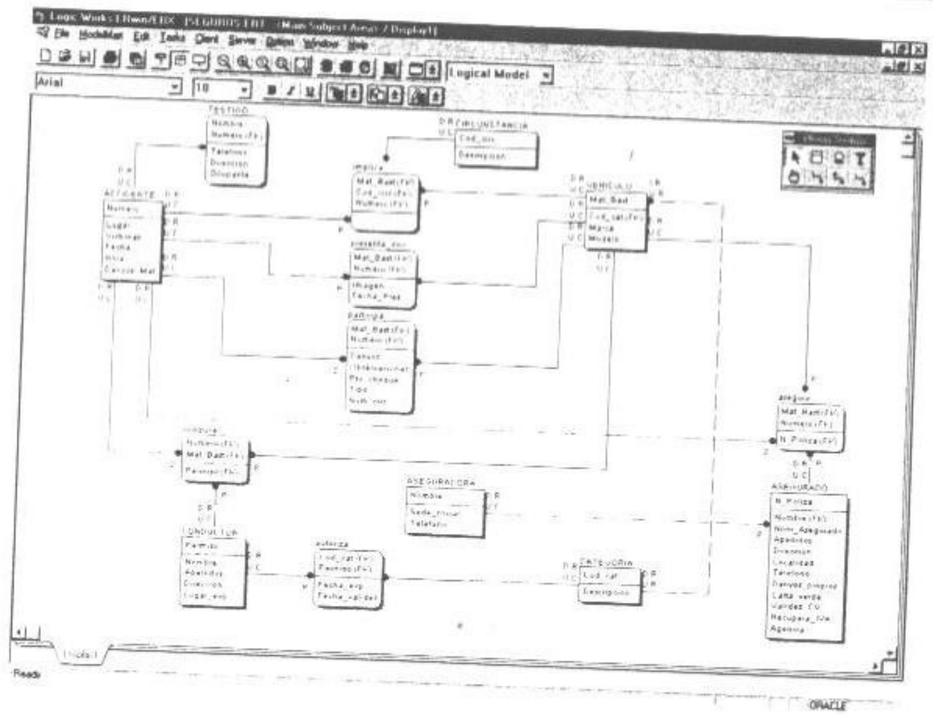


Figura 5.24. Modelo lógico completo

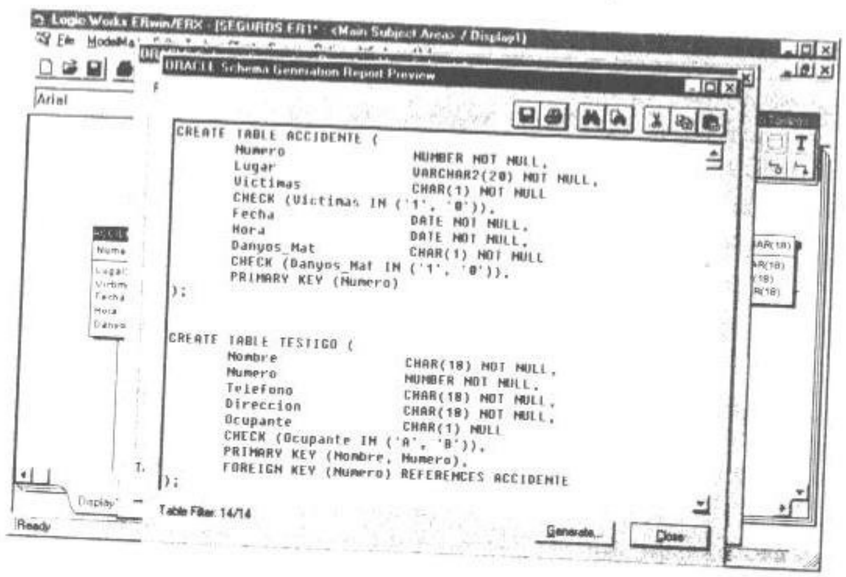


Figura 5.25. Ejemplo de código generado

Este diseño lógico se transformaría en lo que llama ERwin llama "modelo físico" (*Physical Model*) mediante la transformación de las interrelaciones N:M en entidades asociación y que se correspondería con el modelo relacional del sistema gestor elegido para su transformación. En él será dónde se definan aspectos concretos dependientes de cada sistema. En nuestro caso, el diseño físico sería muy similar al diseño lógico, pues no teníamos ninguna interrelación N:M, ya que las que aparecían en nuestro

esquema conceptual se tuvieron que transformar en entidades para poder recoger sus atributos.

Una vez obtenido el diseño completo, la herramienta puede generar el código relativo al modelo de implementación de un sistema concreto. El que elegiremos será la versión 7.0 de ORACLE. ERwin nos permite configurar determinados aspectos de la generación de las tablas, índices, disparadores, etc.

La figura 5.25 muestra parte del código generado para el ejemplo anterior (ver figura en la página anterior).

APÉNDICE

LISTA DE ACRÓNIMOS Y ABREVIATURAS

1FN	PRIMERA FORMA NORMAL
2FN	SEGUNDA FORMA NORMAL
3FN	TERCERA FORMA NORMAL
ANP	ATRIBUTOS NO PRINCIPALES
AP	ATRIBUTOS PRINCIPALES
AT	ATRIBUTOS
B:C	BORRADO EN CASCADA
B:D	BORRADO CON PUESTA A VALOR POR DEFECTO
B:N	BORRADO CON PUESTA A NULOS
B:R	BORRADO RESTRINGIDO

BD	BASE DE DATOS
BDD	BASE DE DATOS DISTRIBUIDA
CASE	COMPUTER AIDED SOFTWARE ENGINEERING
CC	CLAVES CANDIDATAS
DEP	DEPENDENCIAS
DEP^m	CONJUNTO MINIMAL DE DEPENDENCIAS
DF	DEPENDENCIA FUNCIONAL
E/R	ENTIDAD/INTERRELACIÓN
ECA	EVENTO-CONDICIÓN-ACCIÓN
FK	FOREIGN KEY (CLAVE AJENA)
IA	IDENTIFICADOR ALTERNATIVO
IP	IDENTIFICADOR PRINCIPAL
M:C	MODIFICACIÓN EN CASCADA
M:D	MODIFICACIÓN CON PUESTA A VALOR POR DEFECTO
M:N	MODIFICACIÓN CON PUESTA A NULOS
M:R	MODIFICACIÓN RESTRINGIDO
PK	PRIMARY KEY (CLAVE PRIMARIA)
SGBD	SISTEMA DE GESTIÓN DE BASES DE DATOS
SGBDD	SISTEMA DE GESTIÓN DE BASES DE DATOS DISTRIBUIDAS
SQL	STANDARD QUERY LANGUAGE

BIBLIOGRAFÍA

CHEN, P. (1976). *The Entity/Relationship Model: Toward a Unified View of Data*. En: *CACM*, 13, 6.

COOD (1970). *A Relational Model of Data for Large Shared Data Banks"*. En: *CACM*, 1, 1.

DE MIGUEL, A. y PIATTINI, M. (1999). *Fundamentos y Modelos de Bases de Datos*. Ed. Ra-Ma, Madrid.

DE MIGUEL, A., PIATTINI, M. y MARCOS, E.(1999). *Diseño de Bases de Datos Relacionales*. Ed. Ra-Ma, Madrid.

FAGIN, R. (1977). *Multivalued Dependencies and a New Normal Form for Relational Databases*. En: *ACM TODS*, 2, 3.

OZSU, M. T. y VALDURIEZ, P. (1999). *Principles of Distributed Database Systems*. 2ª Ed. Prentice-Hall.

ULLMAN Y WIDOM (1997). A first course in Database Systems. Ed. Prentice-Hall.

ÍNDICE ALFABÉTICO

A

Agrupación, 141
Álgebra Relacional, 138
Análisis, 321
Árbol de descomposición, 321
arquitectura ANSI/X3/SPARC, 432
arquitecturas de los SGBDD, 431
Aserciones, 129
ASSERTION, 129
Atributo, 132
atributos, 2
Atributos compuestos, 133
atributos compuestos/simples, 3
atributos derivados, 3, 133
atributos extraños, 317
Atributos Multivaluados, 132
Atributos no principales, 318
Atributos obligatorios, 133
atributos obligatorios/opcionales, 3
Atributos opcionales, 133
Atributos principales, 318
Atributos univaluados, 132
atributos univaluados/multivaluados, 3
axiomas de Armstrong, 317

B

Bases de Datos Distribuidas, 429
Borrado/modificación con puesta a nulos, 128
Borrado/modificación con puesta a un valor por defecto, 128
Borrado/modificación en cascada, 127
Borrado/modificación restringido, 128

C

Cálculo del recubrimiento minimal, 317
CASCADE, 127
CASE, 459
clave candidata, 318
claves candidatas, 314, 317
combinación, 138, 140
Cuarta forma normal, 320

D

Dependencia de Identificación, 3
dependencia redundante, 317
dependencias, 314

Dependencias en combinación, 316
Dependencias en Existencia, 3
Dependencias funcionales, 314
Dependencias jerárquicas, 316
Dependencias miltivaluadas, 315
Designer2000, 459
diferencia, 138, 140
Diseño conceptual, 1
Disparadores, 129
distribución, 431
división, 138, 141
Dominios, 133

E

Entidad, 132
entidades, 2
entidades débiles, 3
entidades fuertes, 3
ERwin, 459
esquema de asignación y replicación, 439

F

FOREIGN KEY, 126
forma normal, 314
Forma normal de Boyce-Codd, 320
fragmentación, 435
Fragmentación horizontal, 436
Fragmentación mixta, 438
Fragmentación vertical, 435

G

gemeralizaciones exclusivas, 5
generalización parcial, 5
generalización total, 5
generalizaciones, 4
generalizaciones con solapamiento, 5
grafo relacional, 129
GROUP BY, 141

I

Identificador alternativo, 133
Identificador principal, 133
Identificadores Alternativos, 2
Identificadores Principales, 2
integridad referencial, 126
interrelaciones, 3
Interrelaciones Exclusivas, 137
intersección, 138, 141

J

Join, 140

M

metodología ascendente, 434
metodología descendente, 434
métodos de síntesis o análisis, 314
Modelo E/R, 1
Modelo Relacional, 125

N

normalización, 318
NOT NULL, 126

O

opciones de borrado y modificación, 127

P

Pérdidas de semántica, 137
PRIMARY KEY, 126
Primera forma normal, 319
producto cartesiano, 138, 140
proyección, 138, 140

Q

Quinta forma normal, 321
recubrimiento minimal, 314

R

Relación, 125
restricción, 138, 139
restricción de cardinalidad, 3
restricción de clave ajena, 126
restricción de obligatoriedad, 126
Restricción de unicidad, 126
restricciones de clave primaria, 126
Restricciones de Verificación (CHECK), 128
restricciones inherentes, 126
restricciones semánticas, 126
RESTRICT, 128

S

script, 142
Segunda forma normal, 319
selección, 139
SET DEFAULT, 128
SET NULL, 128
SGBDD heterogéneos, 431
SGBDD homogéneo, 431
Síntesis, 321
Sistemas compuestos, 432
Sistemas federados, 432
Sistemas Multi Base de datos, 432

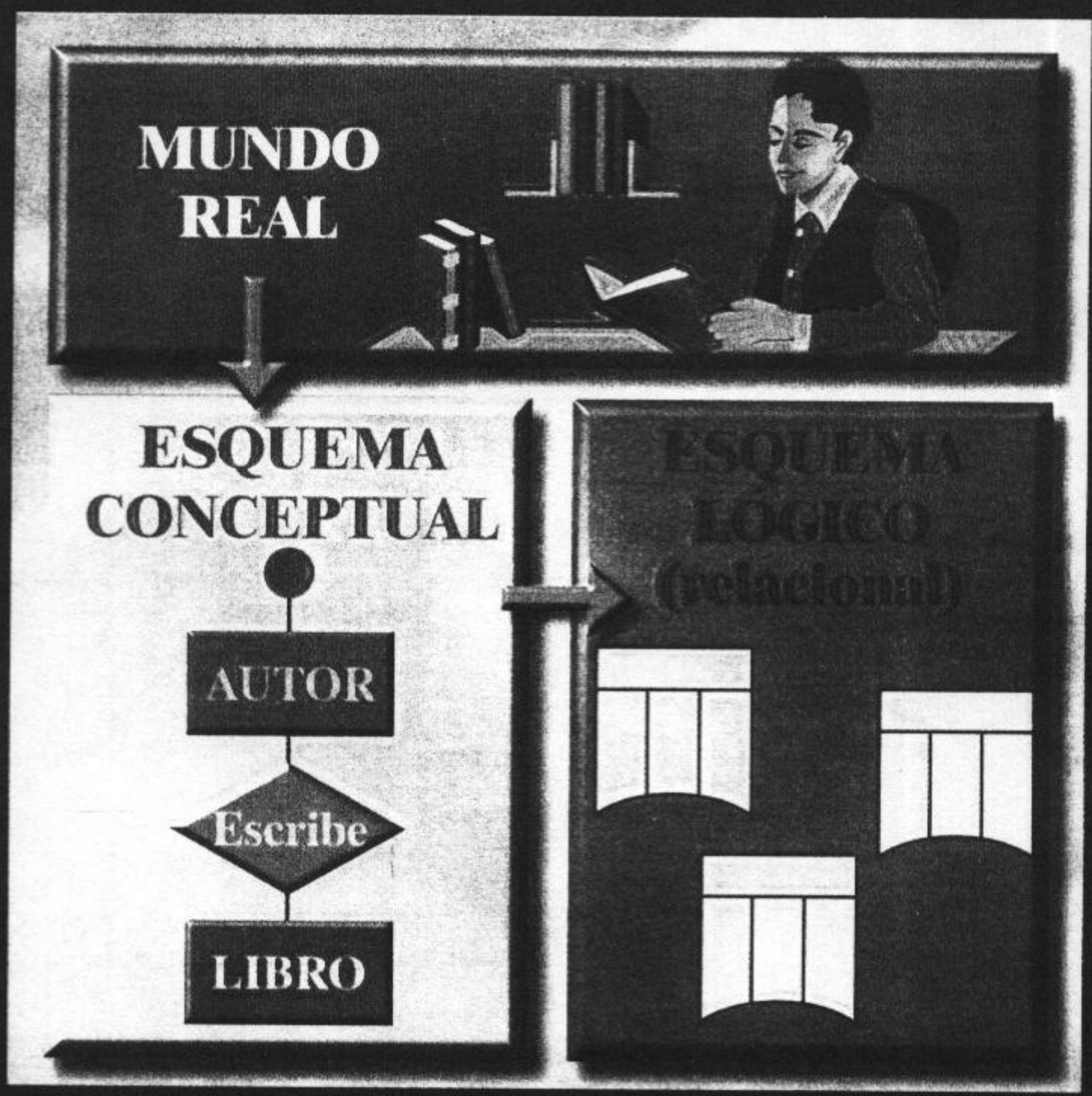
SQL 92, 459
SQL-92, 142
subtipos, 4
superclave, 317
supertipo, 4

T

Tercera forma normal, 320
tipo de correspondencia, 3
Transformación de dependencias en existencia y en identif., 135
Transformación de generalizaciones, 136
Transformación de interrelaciones de grado superior a dos, 136
transformación de un esquema E/R a un esquema relacional, 130
TRIGGER, 129

U

unión, 138, 140
UNIQUE, 126
Universo del Discurso, 1, 5

Diseño de
Bases de Datos
Problemas resueltos

Adoración de Miguel • Paloma Martínez • Elena Castro
José Mª Cavero • Dolores Cuadra • Ana Mª Iglesias • Carlos Nieto

Ra-Ma®