

Microsoft®  
**SQL Azure™**

Administración y desarrollo  
en la nube

MARÍA PÉREZ MARQUÉS

 **Alfaomega**



# **Microsoft SQL Azure**

**Administración y desarrollo en la  
nube**

# **Microsoft SQL Azure**

**Administración y desarrollo en la  
nube**

**María Pérez Marqués**





# ÍNDICE

<b>Introducción .....</b>	<b>XIII</b>
<b>Capítulo 1. Conceptos de base de datos de SQL Azure .....</b>	<b>1</b>
EL NUEVO PORTAL DE ADMINISTRACIÓN .....	1
INFORMACIÓN GENERAL DE SQL AZURE .....	2
SIMILITUDES Y DIFERENCIAS CON SQL SERVER .....	2
ACCESO A DATOS DE SQL AZURE .....	5
Hospedar sus datos en la nube .....	6
Hospedar su aplicación en la nube .....	6
ARQUITECTURA DE SQL AZURE .....	7
El nivel de cliente .....	7
El nivel de servicios .....	9
El nivel de plataforma .....	9
Nivel de infraestructura .....	9
MODELO DE APROVISIONAMIENTO DE SQL AZURE .....	9
Introducción .....	10
Información general sobre el modelo de aprovisionamiento de SQL Azure .....	11
CUENTAS Y FACTURACIÓN EN SQL AZURE .....	12
Ediciones de SQL Azure .....	12
Facturación .....	13
Suscripciones de SQL Azure .....	14
Ancho de banda en la plataforma Windows Azure .....	14
Limitar el tamaño de la base de datos con el valor MAXSIZE .....	15
Acceso a los datos de facturación y uso .....	15
<b>Capítulo 2. Administrar bases de datos e inicios de sesión en SQL Azure ....</b>	<b>17</b>
ADMINISTRACIÓN EN SQL AZURE .....	17
Administración de la seguridad de SQL Azure .....	17
Administración a nivel de servidor y la base de datos maestra .....	19
Administrar inicios de sesión .....	19

Conceder permisos de nivel de servidor a un inicio de sesión .....	20
Conceder acceso a base de datos a un inicio de sesión .....	22
Ver inicios de sesión y bases de datos .....	23
MIGRAR BASES DE DATOS A SQL AZURE .....	23
MIGRAR UNA BASE DE DATOS A SQL AZURE MEDIANTE EL ASISTENTE PARA GENERAR SCRIPTS.....	24
Crear la base de datos de ejemplo SCHOOL .....	24
Crear el script de Transact-SQL.....	25
Ejecutar el script en SQL Azure .....	25
Definición de Transact-SQL de la base de datos School.....	26
MOVER DATOS A SQL AZURE .....	40
<b>Capítulo 3. Copiar bases de datos y supervisar SQL Azure .....</b>	<b>41</b>
COPIAR BASES DE DATOS EN SQL AZURE .....	41
Copiar una base de datos .....	42
Copia en el mismo servidor .....	42
Ejemplo.....	43
COPIA ENTRE SERVIDORES .....	44
Ejemplo.....	45
PERMISOS.....	46
INTERVALOS .....	47
INICIAR EL PROCESO DE COPIA .....	47
Supervisar una copia.....	48
Coherencia transaccional.....	48
DETENER EL PROCESO DE COPIA.....	48
Una vez completada la copia .....	49
SUPERVISAR SQL AZURE MEDIANTE VISTAS DE ADMINISTRACIÓN DINÁMICA....	49
Calcular el tamaño de la base de datos .....	50
Supervisar las conexiones.....	51
Supervisar el rendimiento de las consultas .....	51
Buscar las N consultas superiores.....	51
Supervisar consultas bloqueadas.....	52
Supervisar planes de consulta .....	52
FIREWALL DE SQL AZURE .....	53
Información general .....	53
Conectarse desde Internet .....	54
Conectarse desde Windows Azure .....	54
Crear la primera regla de firewall .....	55
La base de datos maestra .....	55

Solucionar problemas del firewall .....	55
Configurar el firewall de SQL Azure .....	56
Ejemplo.....	58
SOLUCIONAR PROBLEMAS DE SQL AZURE .....	59
<b>Capítulo 4. Desarrollo.....</b>	<b>63</b>
DESARROLLO EN BASE DE DATOS DE SQL AZURE .....	63
Crear servidores SQL Azure .....	63
Crear bases de datos SQL Azure .....	64
Compilar y hospedar aplicaciones SQL Azure .....	65
Desarrollar aplicaciones SQL Azure .....	65
COMPATIBILIDAD CON TRANSACT-SQL (BASE DE DATOS DE SQL AZURE) .....	66
Características de Transact-SQL compatibles y parcialmente compatibles.	67
Características de Transact-SQL no admitidas .....	68
COMPATIBILIDAD DE BASE DE DATOS SQL AZURE CON HERRAMIENTAS Y UTILIDADES DE SQL SERVER .....	69
La utilidad sqlcmd.....	69
SQL Server Management Studio .....	70
El administrador de base de datos para SQL Azure .....	70
Aplicaciones de capa de datos.....	70
Asistente para generar y publicar scripts .....	71
SQL Server Integration Services .....	71
La utilidad bcp .....	72
SQL Server Analysis Services y SQL Server Reporting Services .....	72
SQL Azure Reporting.....	72
Objetos de administración de SQL Server (SMO) .....	72
Asistente de migración de SQL Server (SSMA) .....	73
Microsoft Access.....	73
PowerPivot para Excel .....	73
INSTRUCCIONES Y LIMITACIONES GENERALES (BASE DE DATOS SQL AZURE) .	74
Compatibilidad con controladores, bibliotecas y protocolos .....	74
Compatibilidad con Visual Studio .....	75
Compatibilidad con herramientas y tecnologías .....	75
Compatibilidad con ODBC.....	76
Compatibilidad con la migración de datos .....	77
Trabajos y Agente SQL Server .....	77
Compatibilidad con transacciones.....	77
Niveles de aislamiento basado en versiones de fila.....	78
Compatibilidad con la intercalación de SQL Server .....	78

Requisito de índices clúster .....	78
Número de bases de datos y límites de tamaño.....	79
Requisitos de nomenclatura .....	79
Restricciones de conexión .....	80
INSTRUCCIONES Y LIMITACIONES DE SEGURIDAD EN BASE DE DATOS	
DE SQL AZURE.....	80
Firewall .....	81
Cifrado y validación de certificados .....	81
Autenticación.....	82
Inicios de sesión y usuarios.....	82
Procedimientos recomendados.....	84
LIMITACIONES DE LAS CARACTERÍSTICAS DE SQL SERVER EN BASE DE DATOS	
SQL AZURE.....	84
Características de SQL Server 2008 R2 no admitidas por Base de datos SQL Azure .....	85
Características de SQL Server 2008 no admitidas por Base de datos SQL Azure .....	85
Características de SQL Server 2005 no admitidas por Base de datos SQL Azure .....	86
Otras características de SQL Server R2 no admitidas por Base de datos SQL Azure .....	86
INSTRUCCIONES PARA CONECTAR CON BASE DE DATOS SQL AZURE .....	87
<b>Capítulo 5. Conectar con SQL Azure.....</b>	<b>89</b>
INSTRUCCIONES PARA CONECTAR CON BASE DE DATOS SQL AZURE .....	89
CONECTAR CON SQL AZURE MEDIANTE SQLCMD.....	91
CONECTAR CON SQL AZURE MEDIANTE ADO.NET .....	92
Ejemplo de aplicación de consola.....	92
CONECTAR CON SQL AZURE MEDIANTE ASP.NET .....	97
CONECTAR CON SQL AZURE MEDIANTE SERVICIOS DE DATOS DE WCF (ADO.NET) ..	99
Crear el modelo de datos .....	100
Crear el servicio de datos.....	101
Crear la aplicación cliente.....	102
Tener acceso al servicio de datos School desde el rol web de Windows Azure.	103
Migrar la aplicación a SQL Azure. ....	104
CONECTAR CON SQL AZURE MEDIANTE PHP .....	105
Crear la aplicación Windows Azure .....	105
Crear la aplicación PHP .....	106
CONECTAR CON SQL AZURE MEDIANTE JDBC.....	109
Preparar el entorno de desarrollo .....	109

Ejemplo.....	109
CONECTAR CON SQL AZURE MEDIANTE ADO.NET ENTITY FRAMEWORK .....	111
Crear el modelo de datos .....	111
Crear la aplicación de consola .....	112
Migrar la aplicación a SQL Azure .....	113
<b>Capítulo 6. TRANSACT-SQL en SQL Azure. Tipos de datos, funciones y operadores.</b>	<b>115</b>
TIPOS DE DATOS EN BASE DE DATOS DE SQL AZURE .....	115
Compatibilidad con métodos del tipo de datos geography .....	116
Compatibilidad con métodos del tipo de datos geometry .....	117
Compatibilidad con métodos del tipo de datos hierarchyid .....	118
El tipo de datos xml .....	118
FUNCIONES DE BASE DE DATOS DE SQL AZURE .....	119
Funciones de agregado.....	120
Funciones de categoría.....	120
Funciones de conjuntos de filas.....	120
Funciones escalares .....	120
Funciones escalares de ODBC.....	126
OPERADORES EN BASE DE DATOS DE SQL AZURE .....	127
Operadores aritméticos.....	127
Operadores compuestos .....	127
Operadores bit a bit.....	128
Operadores lógicos .....	128
Operadores de comparación .....	129
Operadores unarios .....	129
Otros operadores misceláneos .....	130
TABLAS DEL SISTEMA DE BASE DE DATOS DE SQL AZURE .....	130
VISTAS DEL SISTEMA DE BASE DE DATOS DE SQL AZURE .....	131
Vistas de bases de datos de SQL Azure.....	131
Vistas de catálogo.....	132
Vistas de compatibilidad.....	134
Vistas de administración dinámica .....	134
Vistas del esquema de información.....	136
<b>Capítulo 7. TRANSACT-SQL en SQL Azure. Instrucciones SQL Server admitidas y procedimientos</b> .....	<b>137</b>
INSTRUCCIONES TRANSACT-SQL ADMITIDAS EN BASE DE DATOS DE SQL AZURE ...	137
Compatibilidad con sugerencias .....	138

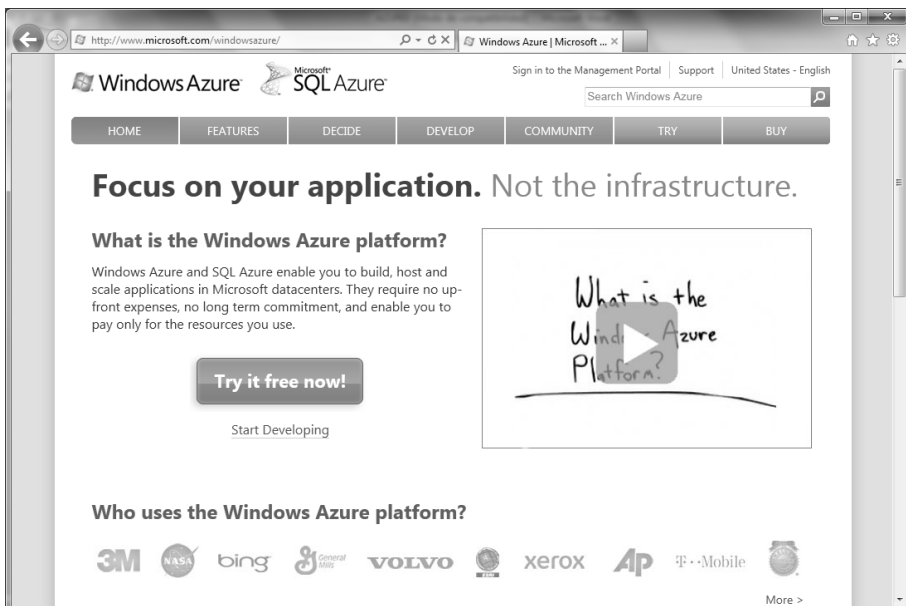
Compatibilidad con instrucciones SET .....	139
INSTRUCCIONES TRANSACT-SQL NO ADMITIDAS EN BASE DE DATOS	
DE SQL AZURE .....	140
PROCEDIMIENTOS ALMACENADOS DEL SISTEMA EN BASE DE DATOS	
DE SQL AZURE .....	144
Procedimientos almacenados de base de datos SQL Azure.....	145
Procedimientos almacenados del catálogo .....	145
Procedimientos almacenados del motor de base de datos .....	146
Procedimientos almacenados de seguridad .....	147
REFERENCIA DE ERRORES Y EXCEPCIONES EN BASE DE DATOS DE SQL AZURE .....	148
Errores generales.....	148
Errores de copia de base de datos.....	149
Errores de pérdida de conexión.....	151
Limitación de SQL Azure y descodificación de códigos de motivo.....	152
Descodificar códigos de motivo .....	153
Excepciones de SMO en base de datos de SQL Azure.....	153
<b>Capítulo 8. TRANSACT-SQL en SQL Azure. Lenguaje de definición de datos DDL ...</b>	<b>155</b>
INSTRUCCIONES TRANSACT-SQL ADMITIDAS DE FORMA PARCIAL	
EN BASE DE DATOS DE SQL AZURE .....	155
Lenguaje DDL. INSTRUCCIONES ALTER EN BASE DE DATOS DE SQL AZURE ...	157
ALTER AUTHORIZATION .....	157
ALTER DATABASE .....	158
ALTER FUNCTION .....	161
ALTER INDEX.....	164
ALTER LOGIN .....	165
ALTER PROCEDURE.....	166
ALTER TABLE.....	167
ALTER TRIGGER .....	170
ALTER USER .....	171
ALTER VIEW .....	172
Lenguaje DDL. INSTRUCCIONES CREATE EN BASE DE DATOS DE SQL AZURE.	172
CREATE DATABASE .....	172
CREATE FUNCTION .....	176
CREATE INDEX .....	179
CREATE LOGIN.....	180
CREATE PROCEDURE .....	181
CREATE SPATIAL INDEX .....	182

CREATE SYNONYM .....	183
CREATE TABLE .....	184
CREATE TRIGGER .....	187
CREATE TYPE .....	188
CREATE USER.....	189
CREATE VIEW .....	190
LENGUAJE DML. INSTRUCCIONES DROP EN BASE DE DATOS DE SQL AZURE...	190
DROP DATABASE .....	191
DROP INDEX .....	191
DROP TABLE .....	192
DROP TRIGGER .....	192
<b>Capítulo 9. TRANSACT-SQL en SQL Azure. Lenguajes de manipulación y control de datos DML y DCL .....</b>	<b>193</b>
LENGUAJE DE MANIPULACIÓN DE DATOS DML. INSTRUCCIÓN INSERT EN BASE DATOS DE SQL AZURE .....	193
LENGUAJE DE CONTROL DE DATOS DCL EN BASE DATOS DE SQL AZURE .....	194
GRANT (permisos de base de datos de Base de Datos de SQL Azure) .....	195
GRANT (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure) .....	195
GRANT (permisos de tipo de Base de datos de SQL Azure).....	196
REVOKE (permisos de base de datos de Base de datos de SQL Azure) .....	197
REVOKE (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure) .....	198
REVOKE (permisos de tipo de Base de datos de SQL Azure) .....	198
DENY (permisos de base de datos de Base de datos de SQL Azure) .....	199
DENY (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure) .....	200
DENY (permisos de tipo de Base de datos de SQL Azure) .....	200
EXECUTE .....	201
EXECUTE AS.....	202
EXECUTE AS Clause .....	202
ENABLE TRIGGER.....	203
DISABLE TRIGGER .....	203
USE (Base de datos de SQL Azure) .....	203
LENGUAJE TCL DE CONTROL DE TRANSACCIONES EN BASE DATOS DE SQL AZURE ..	204
<b>Índice alfabético.....</b>	<b>205</b>

# INTRODUCCIÓN

Microsoft SQL Azure extiende las capacidades de SQL Server a la nube. Asimismo, proporciona un servicio de base de datos relacional denominado Base de datos de Microsoft SQL Azure, un servicio de generación de informes denominado Servicio de informes Microsoft SQL Azure, y una interfaz de administración y diseño de bases de datos basada en web como parte de la plataforma Windows Azure.

Para acceder a información completa sobre la plataforma Windows Azure, vea la página de Microsoft <http://www.microsoft.com/windowsazure/> cuyo aspecto se muestra a continuación.



Microsoft Base de datos de SQL Azure es una plataforma de base de datos relacional basada en la nube creada con tecnologías de SQL Server que permite implementar fácilmente aplicaciones de base de datos relacional en la nube, y aprovechar un centro de datos distribuido que proporciona disponibilidad, escalabilidad y seguridad con las ventajas de protección de datos integrada y recuperación automática. Se trata del servicio de base de datos relacional de la plataforma Windows Azure.

Entre las ventajas de utilizar Base de datos de SQL Azure se incluyen la facilidad de uso, alta disponibilidad, escalabilidad, un modelo de desarrollo conocido y un modelo de datos relacionales.

*Facilidad de uso.* La capacidad de administración automática permite aprovisionar servicios de datos para las aplicaciones sin agregar carga de soporte técnico al departamento central de tecnologías de la información. Cuando las necesidades cambien, puede extender fácilmente el almacenamiento de datos basado en la nube para satisfacer esas necesidades.

*Alta disponibilidad.* Base de datos de SQL Azure se ha creado con tecnologías probadas de Windows Server y SQL Server. El servicio replica varias copias redundantes de sus datos en varios servidores físicos para mantener la disponibilidad de los datos. En caso de que se produzca un error de hardware, Base de datos de SQL Azure proporciona conmutación automática por error para optimizar la disponibilidad de la aplicación.

*Escalabilidad.* Una ventaja primordial de Base de datos de SQL Azure es la facilidad con la que pueden escalar las aplicaciones. Después de crear particiones de los datos, el servicio se escala a medida que crecen los datos.

*Modelo de desarrollo conocido.* Base de datos de SQL Azure proporciona la misma interfaz TDS que SQL Server con vistas a usar las mismas herramientas y bibliotecas para compilar aplicaciones cliente para los datos almacenados en Base de datos de SQL Azure.

*Modelo de datos relacionales.* Base de datos de SQL Azure resultará muy familiar a los desarrolladores y administradores porque los datos se almacenan mediante Transact-SQL. Se aprovechan los diseños de bases de datos relacionales y los conocimientos de programación en Transact-SQL existentes, simplificando el proceso de migración de aplicaciones de base de datos locales existentes a Base de datos de SQL Azure.

# CONCEPTOS DE BASE DE DATOS DE SQL AZURE

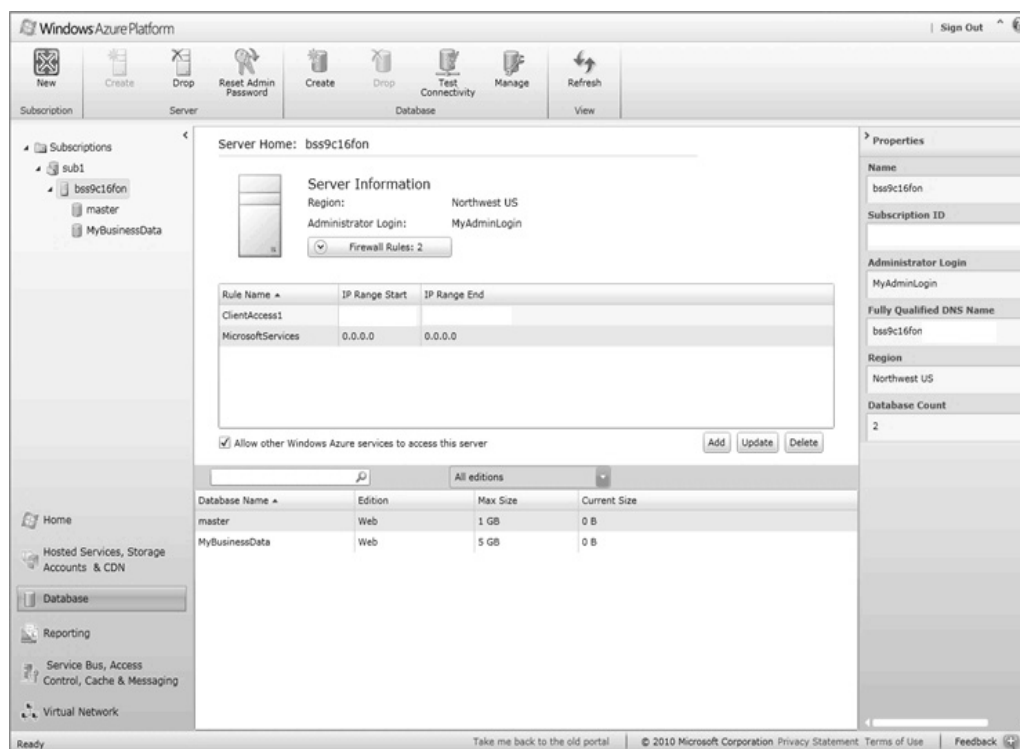
## EL NUEVO PORTAL DE ADMINISTRACIÓN

Hasta fechas recientes han coexistido dos portales de administración de la plataforma Windows Azure que proporcionan la interfaz de usuario para aprovisionar servidores e inicios de sesión y para crear bases de datos rápidamente.

El nuevo portal brinda acceso a una herramienta de administración de bases de datos basada en web para las bases de datos SQL Azure existentes.

La herramienta de administración admite tareas básicas de administración de base de datos como diseñar y editar tablas, vistas y procedimientos almacenados, y crear y ejecutar consultas de Transact-SQL.

La herramienta de administración de base de datos está disponible cuando se selecciona una base de datos para administrar en el nuevo portal de administración y se hace clic en el botón **Manage** (Administrar) de la barra de herramientas. Obsérvese la pantalla siguiente.



## INFORMACIÓN GENERAL DE SQL AZURE

Microsoft Base de datos de SQL Azure es un servicio de base de datos relacional basado en la nube creado con tecnologías SQL Server que se ejecuta en centros de datos de Microsoft en hardware propiedad de Microsoft, que lo hospeda y lo mantiene. En este tema se proporciona información general de Base de datos de SQL Azure y se describen algunas diferencias con respecto a SQL Server.

## SIMILITUDES Y DIFERENCIAS CON SQL SERVER

De manera similar a una instancia de SQL Server, Base de datos de SQL Azure expone una interfaz de flujo TDS para el acceso a bases de datos basado en Transact-SQL. Esto permite que las aplicaciones de base de datos utilicen Base de datos de SQL Azure del mismo modo en que emplean SQL Server. Puesto que Base de datos de SQL Azure es un servicio, la administración en Base de datos de SQL Azure es ligeramente diferente.

A diferencia de la administración para una instancia local de SQL Server, Base de datos de SQL Azure abstrae la administración lógica de la administración física. El usuario continúa administrando las bases de datos, los inicios de sesión, los usuarios y los roles, pero Microsoft administra el hardware físico como discos duros, servidores y almacenamiento. Este enfoque ayuda a Base de datos de SQL Azure a ofrecer un servicio de base de datos de varios inquilinos a gran escala que proporciona disponibilidad, escalabilidad, seguridad y recuperación automática.

Puesto que Microsoft se encarga de toda la administración física, hay algunas diferencias entre Base de datos de SQL Azure y una instancia local de SQL Server en lo que se refiere a la administración, aprovisionamiento, compatibilidad con Transact-SQL, modelo de programación y características.

### **Administración lógica frente a administración física**

Aunque Base de datos de SQL Azure desempeña un papel activo en la administración de los recursos físicos de la base de datos, el administrador de la base de datos (DBA) desempeña un papel muy importante en la administración de aplicaciones de base de datos basadas en SQL Azure. Con Base de datos de SQL Azure, los DBA administran la creación de esquemas, la administración de estadísticas, la optimización de índices, la optimización de consultas y la administración de la seguridad (inicios de sesión, usuarios, roles, etc.).

La administración de bases de datos en Base de datos de SQL Azure difiere en gran medida de SQL Server en lo que se refiere a la administración física. Base de datos de SQL Azure replica automáticamente todos los datos para proporcionar alta disponibilidad. También administra el equilibrio de carga y, en caso de que se produzca un error en un servidor, la conmutación por error transparente.

Para proporcionar este nivel de administración física, el usuario no puede controlar los recursos físicos de Base de datos de SQL Azure. Por ejemplo, no puede especificar el disco duro físico o el grupo de archivos donde residirá una base de datos o un índice. Como el sistema de archivos del equipo no es accesible y todos los datos se replican automáticamente, los comandos de copias de seguridad y restauración de SQL Server no son aplicables a Base de datos de SQL Azure.

SQL Azure permite hacer copia de seguridad de la base de datos del usuario copiándola a una nueva base de datos en SQL Azure.

Aunque los comandos de copias de seguridad y restauración no están disponibles, también puede emplear SQL Server Integration Services y la utilidad SQLCMD para realizar copias masivas de datos.

## **Aprovisionamiento**

Al preparar una implementación local de SQL Server, quizás sea el DBA o el departamento de informática el que prepare y configure el hardware y el software necesario. Cuando se utiliza Base de datos de SQL Azure, estas tareas las realiza el proceso de aprovisionamiento de SQL Azure.

Puede empezar a aprovisionar sus bases de datos SQL Azure después de crear una cuenta de la plataforma Windows Azure. Esta cuenta le permite tener acceso a todos los servicios, como Windows Azure, Windows Azure AppFabric y Base de datos de SQL Azure, y se utiliza para configurar y administrar sus suscripciones.

Cada suscripción a SQL Azure puede enlazarse a uno o varios servidores SQL Azure en el centro de datos de Microsoft. Su servidor SQL Azure es una abstracción que define una agrupación de bases de datos. Para habilitar equilibrio de carga y alta disponibilidad, las bases de datos asociadas a su servidor SQL Azure pueden residir en equipos físicos diferentes en el centro de datos de Microsoft.

## **Compatibilidad con Transact-SQL**

Muchas instrucciones de SQL Server Transact-SQL tienen parámetros que le permiten especificar grupos de archivos o rutas de acceso a archivos físicos. Estos tipos de parámetros no se admiten en Base de datos de SQL Azure porque tienen dependencias de la configuración física. En estos casos, se considera que el comando es parcialmente compatible.

## **Características y tipos**

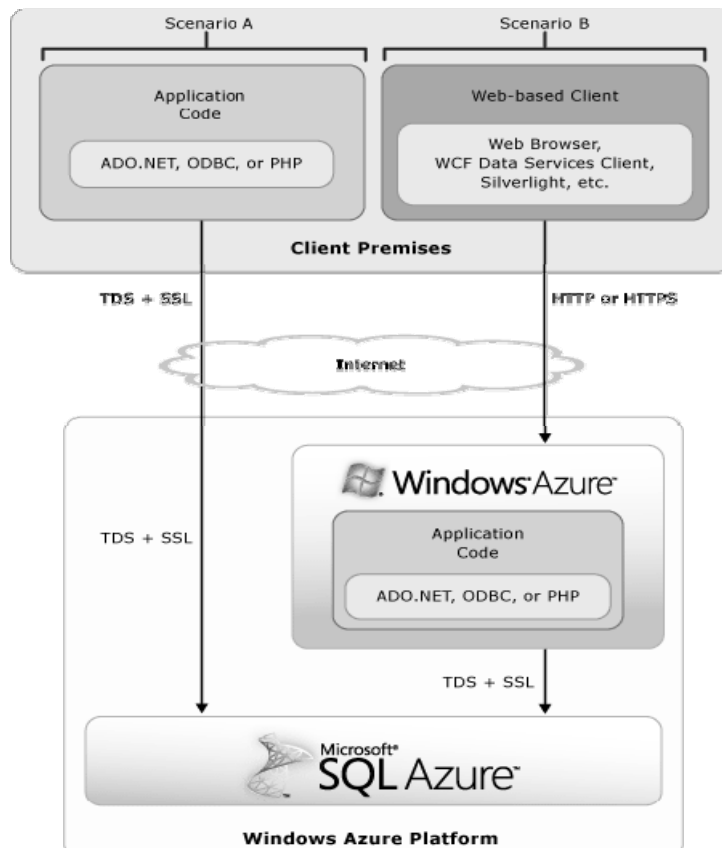
Base de datos de SQL Azure no admite todas las características y los tipos de datos de SQL Server. Analysis Services, Replication y Service Broker no se proporcionan actualmente como servicios en la plataforma Windows Azure. Puesto que Base de datos de SQL Azure realiza la administración física, se bloqueará cualquier instrucción y opción que intente manipular los recursos físicos directamente, como el regulador de recursos, referencias a grupos de archivos y algunas instrucciones de DDL del servidor físico. Tampoco es posible establecer opciones del servidor y marcas de seguimiento de SQL ni emplear las utilidades Generador de perfiles o Asistente para la optimización de base de datos de SQL Server.

Base de datos de SQL Azure admite muchos tipos de datos de SQL Server 2008 pero no admite tipos de datos desusados de SQL Server 2008.

## ACCESO A DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure es el servicio de base de datos relacional de la plataforma Windows Azure. Hay muchas maneras de incorporar Base de datos de SQL Azure a su aplicación de base de datos. En este tema se describen algunos ejemplos generales de cómo su aplicación de base de datos puede tener acceso a datos en Base de datos de SQL Azure.

En una aplicación local tradicional, el código de la aplicación y la base de datos se encuentran en el mismo centro de datos físico. SQL Azure y la plataforma Windows Azure ofrecen muchas alternativas a esa arquitectura. En el diagrama siguiente se muestran dos alternativas generalizadas disponibles sobre cómo su aplicación puede tener acceso a datos con Base de datos de SQL Azure.



En el escenario A de la izquierda, el código de la aplicación sigue siendo local de su centro de datos corporativo, pero la base de datos reside en Base de datos de SQL Azure. El código de la aplicación usa bibliotecas de cliente para tener acceso a las bases de datos en Base de datos de SQL Azure. Independientemente de la biblioteca de cliente elegida, los datos se transfieren utilizando el flujo TDS a través de una capa de sockets seguros (SSL).

En el escenario B de la derecha, el código de la aplicación se hospeda en Windows Azure y la base de datos reside en Base de datos de SQL Azure. Para tener acceso a las bases de datos en Base de datos de SQL Azure, su aplicación puede utilizar las mismas bibliotecas de cliente que están disponibles en el escenario A. Puede hospedar muchos tipos diferentes de aplicaciones en la plataforma Windows Azure.

Las instalaciones cliente del escenario B pueden representar el explorador web de un usuario final que se utiliza para tener acceso a su aplicación web. Las instalaciones cliente del escenario B también pueden ser una aplicación de escritorio o Silverlight que utiliza las ventajas de Entity Data Model y el cliente Servicios de datos de Microsoft WCF para tener acceso a los datos hospedados en Base de datos de SQL Azure.

## **Hospedar sus datos en la nube**

La apariencia y la funcionalidad de su experiencia de base de datos de SQL Server no cambian cuando utiliza SQL Azure en lugar de una instancia local de SQL Server. Hay algunas pequeñas diferencias que incluyen cambios en el código de Transact-SQL y la administración del inicio de sesión.

Cuando solo hospeda datos en el centro de datos de Microsoft, es importante tener en cuenta la latencia de red entre SQL Azure y los servidores locales que ejecutan el código de la aplicación. La latencia de red con SQL Azure es mayor que si la base de datos también estuviera en sus instalaciones y quizás necesite tenerlo en cuenta en el código de la aplicación.

Una manera de minimizar la latencia de red de las solicitudes a la base de datos consiste en hospedar también la aplicación en la plataforma Windows Azure.

## **Hospedar su aplicación en la nube**

Hospedar aplicaciones en Windows Azure es opcional. Según sus necesidades, puede migrar las aplicaciones a/o desde Windows Azure.

Hospedar el código de la aplicación en Windows Azure es beneficioso para el rendimiento de la aplicación porque minimiza la latencia de red asociada a las solicitudes de datos de la aplicación a SQL Azure.

Por ejemplo, puede decidir crear una interfaz de usuario basada en web en Windows Azure para su aplicación de base de datos hospedada en SQL Azure. En este escenario, el código cliente y los datos se hospedan en el mismo centro de datos físico, por lo que la latencia de red es mínimo entre los servidores correspondientes. Los clientes todavía tendrán que atravesar Internet para alcanzar su aplicación de base de datos, pero el tráfico de red entre el código cliente y los datos es mucho más rápido porque esas solicitudes de datos numerosas tienen que recorrer una distancia mucho más corta.

## ARQUITECTURA DE SQL AZURE

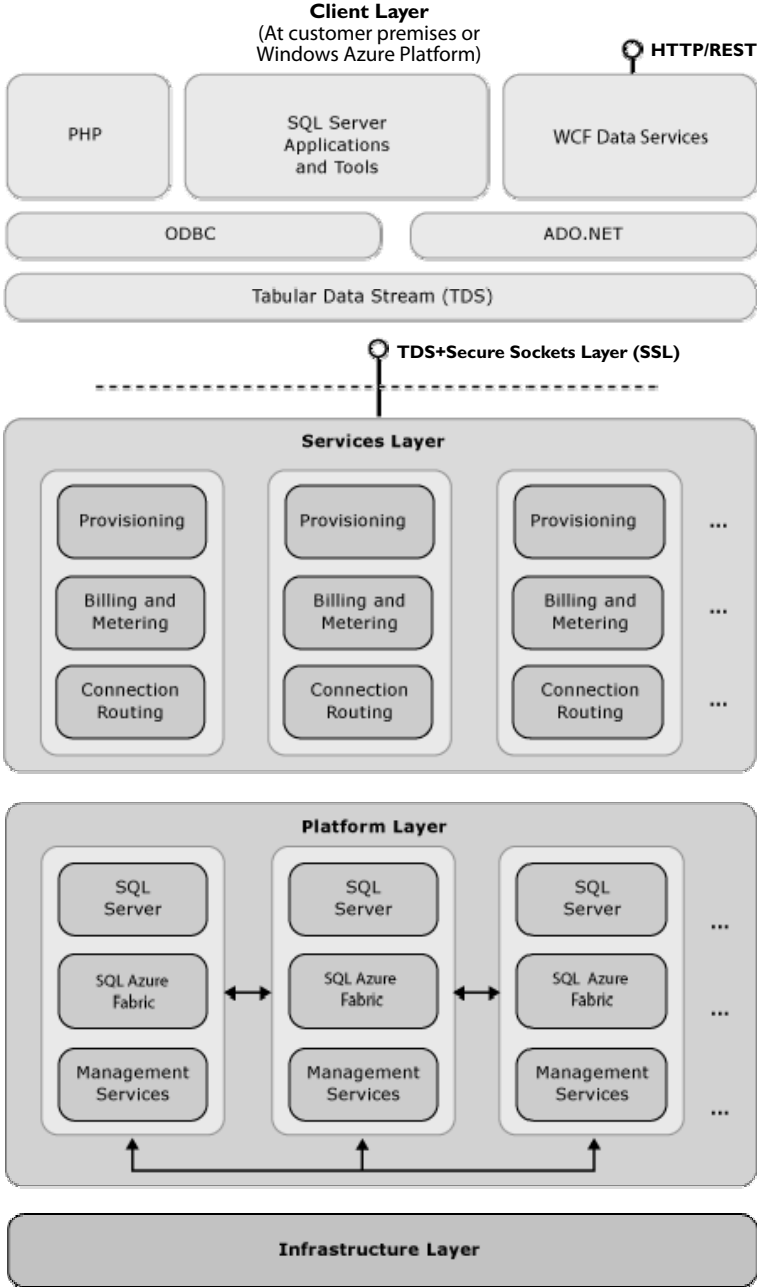
Microsoft Base de datos de SQL Azure se hospeda en servidores que ejecutan tecnologías de SQL Server en centros de datos de Microsoft. De una perspectiva arquitectónica, hay cuatro niveles distintos de abstracción que funcionan conjuntamente para proporcionar una base de datos relacional para su aplicación: el nivel de cliente, el nivel de servicios, el nivel de plataforma y el nivel de infraestructura.

Como se ve en el diagrama de la página siguiente, SQL Azure funciona con aplicaciones de terceros, código abierto y muchas aplicaciones conocidas de Microsoft.

### El nivel de cliente

El nivel de cliente reside más cerca de su aplicación y esta lo utiliza para comunicarse directamente con SQL Azure. El nivel de cliente puede residir de forma local en su centro de datos o estar hospedado en Windows Azure. Puesto que SQL Azure proporciona la misma interfaz de flujo TDS que SQL Server, puede utilizar herramientas y bibliotecas conocidas para compilar aplicaciones cliente para datos que están en la nube.

Hay muchas opciones disponibles para que su aplicación se comunique con SQL Azure.



## **El nivel de servicios**

El nivel de servicios funciona como puerta de enlace entre el nivel de cliente y el nivel de plataforma, donde residen los datos. El nivel de servicios proporciona tres funciones: aprovisionamiento, facturación y medida, y enrutamiento de conexión.

El nivel de servicios aprovisiona las bases de datos que especifica con su cuenta de la plataforma Windows Azure. El aspecto de facturación y medida del nivel de servicios habilita la compatibilidad con múltiples inquilinos proporcionando supervisión y facturación para el uso de la base de datos según las cuentas individuales de la plataforma Windows Azure. SQL Azure se ha creado sobre una plataforma escalable que implica numerosos servidores físicos; este nivel administra todo el enrutamiento de conexiones entre su aplicación y los servidores físicos donde residen sus datos.

## **El nivel de plataforma**

El nivel de plataforma incluye los servidores físicos y los servicios que admiten el nivel de servicios. El nivel de plataforma consta de muchas instancias de SQL Server, cada una de las cuales está administrada por el tejido de SQL Azure.

El tejido de SQL Azure es un sistema informático distribuido compuesto por redes firmemente integradas, servidores y almacenamiento. Habilita la conmutación automática por error, el equilibrio de carga y la replicación automática entre los servidores físicos.

Los servicios de administración supervisan el mantenimiento de los servidores individuales, y permiten la instalación automatizada de actualizaciones de los servicios y revisiones de software.

## **Nivel de infraestructura**

El nivel de infraestructura representa la administración de TI del hardware físico y los sistemas operativos que admiten el nivel de servicios.

## **MODELO DE APROVISIONAMIENTO DE SQL AZURE**

El proceso de aprovisionamiento de Microsoft SQL Azure prepara y configura el hardware y el software necesarios para mantener sus datos en un centro de datos de

Microsoft. Este proceso de aprovisionamiento de SQL Azure describe la jerarquía lógica de su cuenta de la plataforma Windows Azure, y de sus servidores y bases de datos SQL Azure; también le permite aprovisionar el almacenamiento de datos rápidamente.

## Introducción

Para utilizar Base de datos de SQL Azure, debe crear primero una cuenta de la plataforma Windows Azure, que le permite tener acceso a todos los servicios, como Windows Azure, Windows Azure AppFabric y Base de datos de SQL Azure. La cuenta de la plataforma Windows Azure se utiliza para configurar y administrar sus suscripciones, y para facturar el consumo de los servicios Azure. La suscripción se adquiere desde Microsoft Online Services - Portal del cliente. Una vez completada la compra, recibirá una notificación de confirmación por correo electrónico con instrucciones de acceso a su cuenta en la plataforma SQL Azure.

Una vez creada la cuenta de la plataforma Windows Azure, puede utilizar el Portal de administración de la plataforma Windows Azure o la API de REST de administración de bases de datos SQL Azure para crear uno o más servidores SQL Azure. El Portal de administración proporciona una interfaz de usuario que puede usar para aprovisionar servidores e inicios de sesión y para crear bases de datos rápidamente. La API de administración de bases de datos es una API de REST para administrar mediante programación reglas de firewall y servidores SQL Azure.

A partir de la versión de marzo de 2011, SQL Azure admite la creación de varios servidores para cada suscripción a la plataforma Windows Azure. Puede aprovisionar varios servidores SQL Azure en los mismos o diferentes centros de datos de todo el mundo y todos sus servidores pueden enlazarse a una sola suscripción. De este modo se proporciona una sola factura para todos sus servidores y puede crear servidores adicionales sin necesidad de crear una nueva suscripción al Portal del cliente de Microsoft Online Services. Cada servidor SQL Azure tiene sus propios usuarios, inicios de sesión, bases de datos y reglas de firewall.

De forma predeterminada, SQL Azure admite hasta 6 servidores por suscripción y 150 bases de datos en cada servidor SQL Azure, incluida la base de datos maestra. Este límite se puede ampliar. Para obtener más información, póngase en contacto con un representante del servicio de soporte al cliente en el Portal del cliente de Microsoft Online Services. Cuando habilite o deshabilite su suscripción al citado portal, SQL Azure habilitará o deshabilitará automáticamente la totalidad de servidores y bases de datos asociados a esa suscripción.

Una vez creado el servidor SQL Azure, puede utilizar las herramientas tradicionales para trabajar con SQL Azure.

## **Información general sobre el modelo de aprovisionamiento de SQL Azure**

Cada cuenta de la plataforma Windows Azure puede estar asociada a varios servidores SQL Azure y cada servidor SQL Azure puede estar asociado a una o más bases de datos.

### **Servidores**

Un servidor SQL Azure es un grupo lógico de bases de datos y actúa como punto administrativo central para varias bases de datos. Cada servidor SQL Azure incluye inicios de sesión similares a los de las instancias de SQL Server en sus instalaciones.

Cada servidor SQL Azure tiene un nombre de dominio completo único, que se genera durante el proceso de aprovisionamiento de servidores SQL Azure.

### **Bases de datos**

Cada servidor SQL Azure puede contener varias bases de datos. En cada base de datos, puede crear tablas, vistas, índices, procedimientos almacenados y otros objetos conocidos de base de datos. Para crear una nueva base de datos, puede utilizar el Portal de administración o la instrucción CREATE DATABASE (Base de datos SQL Azure).

El proceso de aprovisionamiento crea una base de datos maestra (master) de solo lectura automáticamente. La base de datos maestra (master) realiza un seguimiento de qué inicios de sesión tienen permiso para crear bases de datos u otros inicios de sesión. Debe estar conectado a la base de datos maestra (master) siempre que utilice CREATE, ALTER o DROP para crear, modificar o quitar inicios de sesión o bases de datos, respectivamente. La base de datos maestra (master) también proporciona métricas de uso de SQL Azure que puede ver.

### **Inicios de sesión**

Durante el proceso de aprovisionamiento, SQL Azure crea un inicio de sesión para usted que es la entidad de seguridad a nivel de servidor de su servidor SQL Azure. Esta entidad de seguridad a nivel de servidor es similar al inicio de sesión sa de SQL Server. Es posible crear bases de datos e inicios de sesión SQL Azure adicionales en el servidor, según sea necesario.

La administración a nivel de servidor para los inicios de sesión y las bases de datos en SQL Azure difiere de una instancia local de SQL Server.

SQL Azure proporciona el mismo conjunto de entidades de seguridad que hay disponibles en SQL Server, como:

- Inicios de sesión de SQL Server: se emplean para autenticar el acceso a SQL Azure a nivel de servidor.
- Usuarios de base de datos: se usan para conceder acceso a SQL Azure a nivel de base de datos.
- Roles de base de datos: se utilizan para agrupar usuarios y conceder acceso a SQL Azure a nivel de base de datos.

La administración a nivel de base de datos de los usuarios y roles en SQL Azure es igual que para una instancia local de SQL Server.

## CUENTAS Y FACTURACIÓN EN SQL AZURE

Cuando utiliza Microsoft Base de datos de SQL Azure, puede almacenar los datos de la manera que mejor se adapte al tamaño y a las necesidades de su negocio. En SQL Azure, la facturación se basa en el uso y la edición de la base de datos, por lo que puede empezar con una pequeña inversión y agregar espacio a medida que su negocio crece. SQL Azure proporciona dos ediciones de base de datos diferentes: Business Edition y Web Edition. A diferencia de lo que ocurre en SQL Server local, donde las características de la edición se aplican a una instancia individual, las características de edición de SQL Azure se aplican a la base de datos individual. De esta forma, puede mezclar y usar distintas ediciones de la base de datos dentro del mismo servidor SQL Azure.

### Ediciones de SQL Azure

Las bases de datos SQL Azure están disponibles en dos ediciones: Web Edition y Business Edition. Ambas ediciones proporcionan escalabilidad, alta disponibilidad automatizada y aprovisionamiento automático.

- La base de datos Web Edition es más adecuada para pequeñas aplicaciones web y para aplicaciones de grupos de trabajo o departamentales. Esta edición admite una base de datos con un tamaño máximo de 1 o 5 GB de datos.

- La base de datos Business Edition es más indicada para fabricantes de software independientes (ISV), aplicaciones de línea de negocio (LOB) y aplicaciones empresariales. Esta edición admite una base de datos de hasta 50 GB de datos, en incrementos de 10 GB.

En ambas ediciones se cobra una cuota según el ancho de banda adicional cuando la transferencia de datos incluye un cliente fuera de la plataforma Windows Azure o fuera de la región de la base de datos SQL Azure.

La edición y el tamaño máximo de la base de datos se especifican al crearla, y también puede cambiar la edición y el tamaño máximo después de su creación. Cuando crea una nueva base de datos mediante Transact-SQL, la cláusula MAXSIZE de la instrucción CREATE DATABASE establece un límite en cuanto al tamaño al que puede crecer la base de datos. Si la base de datos alcanza el límite MAXSIZE, recibirá un error y no podrá agregar datos a la base de datos hasta que elimine datos o aumente el valor de MAXSIZE.

Cuando desee cambiar la edición o el valor de MAXSIZE después de crear la base de datos, use la cláusula MODIFY de la instrucción ALTER DATABASE. Cuando lo haga, la facturación se basará en el nuevo tipo de edición (y el tamaño máximo que alcance la base de datos, diariamente).

## Facturación

Se cobra una cuota mensual por cada base de datos de usuario de SQL Azure (las bases de datos maestras no cuestan nada). La cuota de la base de datos se amortiza durante el mes y se carga diariamente. La cuota diaria depende del tamaño máximo que cada base de datos alcanzó ese día, la edición (Web Edition o Business Edition) de cada base de datos y el número máximo de bases de datos que utiliza. Se emplea un multiplicador de 10 GB para el precio de las bases de datos Business Edition y un multiplicador de 1 o 5 GB para las bases de datos Web Edition. Usted paga por las bases de datos que tiene, durante los días que las tiene. La factura mensual es la suma de las cuotas diarias. Si actualiza una base de datos de Web Edition a Business Edition y después vuelve a Web Edition en el mismo día, durante ese día se le cobrará por Business Edition.

Por ejemplo, tiene una única base de datos Business Edition que alcanzó un tamaño máximo de 25 GB de datos durante un día. Con el multiplicador de 10 GB,

una base de datos Business Edition de 25 GB sería equivalente a tres bases de datos Business Edition de 10 GB durante un mes. Para calcular la cuota diaria, divida entre el número promedio de días del mes. Suponiendo un mes de 30 días, su uso de datos diario durante ese día sería de 1 GB de una base de datos Business Edition o  $(3 \times 10 \text{ GB por mes} / 30 \text{ días por mes}) = 1 \text{ GB/día}$ .

## Suscripciones de SQL Azure

Microsoft Online Services - Portal del cliente es el lugar donde se compran las suscripciones a la plataforma Windows Azure y a otros servicios de Microsoft Online Services. Una vez completada la compra, recibirá una notificación de confirmación por correo electrónico con instrucciones para tener acceso a su cuenta en la plataforma SQL Azure.

Al comprar su suscripción al servicio a través de Microsoft Online Services - Portal del cliente, se le pedirá que cree un perfil para el propietario de la cuenta. El propietario de la cuenta o administrador de la cuenta es la persona que se suscribe a la plataforma SQL Azure y otros servicios de Microsoft Online Services. Como propietario de la cuenta, usted posee la relación de facturación con Microsoft. Puede iniciar sesión en Microsoft Online Services - Portal del cliente para administrar sus suscripciones y en el Portal de administración de la plataforma Windows Azure para administrar sus servidores SQL Azure.

Para activar su suscripción, necesitará especificar los detalles de contacto de la persona de su compañía que será responsable de configurar y administrar el servicio. Esta persona se denomina administrador de servicios y lo designa el propietario de la cuenta. El administrador de servicios se convierte de forma predeterminada en el propietario de la cuenta de su perfil, en caso de que el administrador de servicios sea el mismo que el propietario de la cuenta. El administrador de servicios solo puede tener acceso al Portal de administración. En cuanto la activación se haya completado, se enviará un mensaje de correo electrónico al propietario de la cuenta y al administrador de servicios donde se indican los pasos que hay que realizar a continuación.

## Ancho de banda en la plataforma Windows Azure

El ancho de banda empleado entre SQL Azure y Windows Azure o Windows Azure AppFabric es gratuito dentro de la misma subregión o el mismo centro de datos. A la hora de diseñar el sistema, organícelo de forma que las operaciones que implican grandes transferencias de datos se realicen dentro de la plataforma Windows Azure y

de la misma subregión. Al implementar una aplicación Windows Azure, ponga la aplicación y la base de datos SQL Azure dentro de la misma subregión para evitar costos de ancho de banda.

## Limitar el tamaño de la base de datos con el valor MAXSIZE

Al elegir una edición de la base de datos, también selecciona el tamaño máximo (MAXSIZE). MAXSIZE ofrece la posibilidad de limitar el tamaño de la base de datos. Si el tamaño de la base de datos alcanza el valor de MAXSIZE, recibirá un código de error 40544. Cuando esto ocurre, no puede insertar o actualizar datos ni crear objetos nuevos (como tablas, procedimientos almacenados, vistas y funciones). Sin embargo, todavía puede leer y eliminar datos, truncar tablas, quitar tablas e índices y volver a generar índices. Después, puede actualizar MAXSIZE a un valor mayor que el tamaño actual de la base de datos o eliminar algunos datos para liberar espacio de almacenamiento. Pueden transcurrir hasta quince minutos antes de que se puedan insertar datos nuevos. En el caso de Web Edition, los valores válidos de MAXSIZE son: 1 GB o 5 GB. Para Business Edition, los valores válidos son: 10 GB, 20 GB, 30 GB, 40 GB o 50 GB. Se le facturará según la edición y el tamaño máximo que alcance la base de datos, diariamente. Por ejemplo, si tiene cuatro bases de datos Web Edition con un valor MAXSIZE de 1 GB y cada una de ellas tiene 250 MB, se le cobrará la cuota de 1 GB por mes por cada base de datos. El uso de la base de datos se calcula diariamente, por lo que si crea una base de datos a mediados de mes, se le cobrará una cuota prorrateada por esa base de datos.

El tamaño de la base de datos incluye los datos, los índices y los objetos. El tamaño no incluye registros, tablas del sistema, catálogos del servidor, la base de datos **master** o réplicas adicionales. Si supera la capacidad de almacenamiento, el servicio rechazará inserciones adicionales. La base de datos no se convertirá automáticamente a una edición que admita inserciones adicionales.

## Acceso a los datos de facturación y uso

Puede tener acceso a los detalles de uso de las cuentas de SQL Azure mediante dos vistas del sistema en el servidor SQL Azure. La vista `sys.database_usage` muestra el número, el tipo y la duración de las bases de datos del servidor y la vista `sys.bandwidth_usage` describe el ancho de banda utilizado con cada base de datos. Para obtener más información, vea `sys.database_usage` (Base de datos de SQL Azure) y `sys.bandwidth_usage` (Base de datos de SQL Azure).

# ADMINISTRAR BASES DE DATOS E INICIOS DE SESIÓN EN SQL AZURE

## ADMINISTRACIÓN EN SQL AZURE

En Microsoft Base de datos de SQL Azure, al suscribirse al servicio, el proceso de aprovisionamiento crea un servidor SQL Azure, una base de datos denominada **master** y un inicio de sesión que es la entidad de seguridad a nivel de servidor de su servidor SQL Azure. Ese inicio de sesión es similar a la entidad de seguridad a nivel de servidor, **sa**, de una instancia de SQL Server local.

La cuenta de entidad de seguridad a nivel de servidor de SQL Azure siempre tiene permiso para administrar toda la seguridad a nivel de servidor y de base de datos. En este tema se describe cómo puede utilizar la entidad de seguridad a nivel de servidor y otras cuentas para administrar inicios de sesión y bases de datos en Base de datos de SQL Azure.

## Administración de la seguridad de SQL Azure

La administración de la seguridad en Base de datos de SQL Azure es similar a la administración de la seguridad en una instancia local de SQL Server. La administración de la seguridad a nivel de base de datos es casi idéntica, con algunas diferencias únicamente en los parámetros disponibles. Puesto que las bases de datos

SQL Azure pueden escalarse a uno o más equipos físicos, Base de datos de SQL Azure utiliza una estrategia diferente para la administración a nivel de servidor. En la tabla siguiente se resumen las diferencias entre la administración de la seguridad en SQL Server local y en Base de datos de SQL Azure.

Diferencia	SQL Server local	Base de datos de SQL Azure
Dónde se administra la seguridad a nivel de servidor	Carpeta <b>Seguridad</b> del <b>Explorador de objetos</b> de SQL Server Management Studio	Base de datos <b>master</b>
Rol de seguridad a nivel de servidor para crear inicios de sesión	securityadmin	loginmanager
Comandos para administrar inicios de sesión	CREATE LOGIN ALTER LOGIN DROP LOGIN	CREATE LOGIN ALTER LOGIN DROP LOGIN (Hay algunas limitaciones de parámetros y debe estar conectado a la base de datos <b>master</b> )
Vista que muestra todos los inicios de sesión	sys.sql_logins	sys.sql_logins (Debe estar conectado a la base de datos <b>master</b> )
Rol a nivel de servidor para crear bases de datos	dbcreator	dbmanager
Comando para crear una base de datos	CREATE DATABASE	CREATE DATABASE (Hay algunas limitaciones de parámetros y debe estar conectado a la base de datos <b>master</b> )
Quitar bases de datos	DROP DATABASE	DROP DATABASE Si un usuario está en el rol <b>dbmanager</b> , tienen permiso para quitar (DROP) cualquier base de datos, independientemente de qué usuario la creara originalmente.
Vista que muestra todas las bases de datos	sys.databases (vista)	sys.databases (Debe estar conectado a la base de datos <b>master</b> )

## Administración a nivel de servidor y la base de datos maestra

El servidor SQL Azure es una abstracción que define una agrupación de bases de datos. Las bases de datos asociadas al servidor SQL Azure pueden residir en equipos físicos diferentes del centro de datos de Microsoft. Se debe realizar la administración a nivel de servidor de todos ellos usando una única base de datos denominada **master**.

La base de datos **master** realiza un seguimiento de los inicios de sesión y de qué inicios de sesión tienen permiso para crear bases de datos u otros inicios de sesión. Debe estar conectado a la base de datos **master** siempre que utilice CREATE, ALTER o DROP para crear, modificar o quitar inicios de sesión o bases de datos, respectivamente. La base de datos **master** también tiene las vistas sys.sql\_logins y sys.databases, que puede utilizar para ver inicios de sesión y bases de datos, respectivamente.

No se admite el comando USE para la conmutación entre las bases de datos. Hay que establecer una conexión directamente con la base de datos de destino.

Puede administrar la seguridad a nivel de base de datos para los usuarios y objetos en Base de datos de SQL Azure de la misma manera que en una instancia local de SQL Server. Solo hay diferencias en cuanto a los parámetros disponibles en los comandos correspondientes.

## Administrar inicios de sesión

Para administrar inicios de sesión con el inicio de sesión de entidad de seguridad a nivel de servidor, conéctese a la base de datos **master**. Puede utilizar las instrucciones CREATE LOGIN, ALTER LOGIN o DROP LOGIN. En el ejemplo siguiente se crea un inicio de sesión denominado login1:

```
-- first, connect to the master database  
CREATE LOGIN login1 WITH password='<ProvidePassword>';
```

Debe utilizar una contraseña segura al crear un inicio de sesión.

## Utilizar nuevos inicios de sesión

Para conectarse a Base de datos de SQL Azure utilizando los inicios de sesión que crea, primero debe conceder permisos a nivel de base de datos a cada inicio de sesión mediante el comando CREATE USER. Vea Conceder permisos de nivel de base de datos a un inicio de sesión para obtener más información.

Puesto que algunas herramientas implementan el flujo TDS de manera diferente, quizás tenga que anexar el nombre del servidor SQL Azure al inicio de sesión en la cadena de conexión usando la notación <login>@<server>. En estos casos, separe el inicio de sesión y el nombre del servidor SQL Azure con el símbolo @. Por ejemplo, si su inicio de sesión se denomina login1 y el nombre completo de su servidor SQL Azure es servername.database.windows.net, el parámetro de nombre de usuario de la cadena de conexión debe ser: login1@servername. Esta restricción impone algunas limitaciones sobre el texto que puede elegir para el nombre de inicio de sesión. Para obtener más información, vea CREATE LOGIN (Base de datos de SQL Azure).

## Conceder permisos de nivel de servidor a un inicio de sesión

Para que los inicios de sesión distintos de la entidad de seguridad a nivel de servidor administren la seguridad a nivel de servidor, Base de datos de SQL Azure proporciona dos roles de seguridad: loginmanager para crear inicios de sesión y dbmanager para crear bases de datos. Estos roles solo se pueden asignar a usuarios en la base de datos **master**.

Para crear inicios de sesión o bases de datos, debe estar conectado a la base de datos **master**.

### El rol loginmanager

Al igual que el rol securityadmin para una instancia local de SQL Server, el rol loginmanager en Base de datos de SQL Azure es necesario para crear inicios de sesión. Solo el inicio de sesión de entidad de seguridad a nivel de servidor (creado por el proceso de aprovisionamiento) u otros inicios de sesión que tengan el rol loginmanager pueden crear otros inicios de sesión.

## El rol dbmanager

El rol dbmanager de Base de datos de SQL Azure es similar al rol dbcreator para una instancia local de SQL Server. Es necesario para crear bases de datos. Solo el inicio de sesión de entidad de seguridad a nivel de servidor (creado por el proceso de aprovisionamiento) u otros inicios de sesión que tengan el rol dbmanager pueden crear bases de datos. Cuando un inicio de sesión tiene el rol dbmanager, puede crear una base de datos con el comando CREATE DATABASE de SQL Azure, pero ese comando se debe ejecutar en la base de datos master. Para obtener más información, vea CREATE DATABASE (Base de datos SQL Azure).

## Asignar roles de nivel de servidor SQL Azure

Para crear un inicio de sesión que pueda crear bases de datos u otros inicios de sesión, siga estos pasos:

1. Conéctese a la base de datos **master**.
2. Cree un inicio de sesión mediante el comando CREATE LOGIN.
3. Cree a un nuevo usuario para ese inicio de sesión en la base de datos **master** mediante el comando CREATE USER.
4. Utilice el procedimiento almacenado sp\_addrolemember para agregar al usuario el rol dbmanager , el rol loginmanager o ambos.

En el ejemplo de código siguiente se muestra cómo crear un inicio de sesión, login1, y un usuario de base de datos correspondiente denominado login1User que puede crear bases de datos u otros inicios de sesión mientras está conectado a la base de datos **master**:

```
-- first, connect to the master database
CREATE LOGIN login1 WITH password='<ProvidePassword>';
CREATE USER login1User FROM LOGIN login1;
EXEC sp_addrolemember 'dbmanager', 'login1User';
EXEC sp_addrolemember 'loginmanager', 'login1User';
```

Debe utilizar una contraseña segura al crear un inicio de sesión.

## Conceder acceso a base de datos a un inicio de sesión

Todos los inicios de sesión se deben crear en la base de datos **master**. Una vez creado un inicio de sesión, puede crear una cuenta de usuario en otra base de datos para ese inicio de sesión. Base de datos de SQL Azure también admite roles de base de datos igual que una instancia local de SQL Server.

Para crear una cuenta de usuario en otra base de datos, suponiendo que no haya creado un inicio de sesión o una base de datos, siga estos pasos:

1. Conéctese a la base de datos **master** (con un inicio de sesión que tenga los roles loginmanager y dbmanager).
2. Cree un nuevo inicio de sesión mediante el comando CREATE LOGIN. No se admite la autenticación de Windows.
3. Cree una nueva base de datos mediante el comando CREATE DATABASE.
4. Establezca una conexión a la nueva base de datos (con el inicio de sesión que creó la base de datos).
5. Cree a un nuevo usuario en la nueva base de datos mediante el comando CREATE USER.

En el ejemplo de código siguiente se muestra cómo crear un inicio de sesión denominado login1 y una base de datos denominada database1:

```
-- first, connect to the master database
CREATE LOGIN login1 WITH password='<ProvidePassword>';
CREATE DATABASE database1;
```

Debe utilizar una contraseña segura al crear un inicio de sesión.

En el ejemplo siguiente se muestra cómo crear a un usuario de base de datos denominado login1User en la base de datos database1 correspondiente al inicio de sesión login1:

```
-- Establish a new connection to the database1 database
CREATE USER login1User FROM LOGIN login1;
```

Este modelo de permisos a nivel de base de datos de Base de datos de SQL Azure es el mismo que en una instancia local de SQL Server.

Las instrucciones de Transact-SQL relacionadas con la seguridad en Base de datos de SQL Azure pueden diferir ligeramente en los parámetros que hay disponibles.

## Ver inicios de sesión y bases de datos

Para ver los inicios de sesión y las bases de datos de su servidor SQL Azure, utilice las vistas `sys.sql_logins` y `sys.databases` de la base de datos **master**, respectivamente. En el ejemplo siguiente se muestra cómo ver una lista de todos los inicios de sesión y las bases de datos de su servidor SQL Azure.

```
-- first, connect to the master database
SELECT * FROM sys.sql_logins;
SELECT * FROM sys.databases;
```

## MIGRAR BASES DE DATOS A SQL AZURE

Para migrar una base de datos SQL Server existente a Base de datos de SQL Azure, exporte el esquema de la base de datos y, a continuación, transfiera los datos de la base de datos a Base de datos de SQL Azure.

### Migrar bases de datos a SQL Azure generando un script de una base de datos existente

Para migrar una base de datos SQL Server existente a SQL Azure, puede exportar el esquema como un script de Transact-SQL y, a continuación, ejecutar ese script en Base de datos de SQL Azure de la manera siguiente:

1. Exporte un script de SQL Server Management Studio y establezca las opciones de exportación adecuadas. Puesto que Base de datos de SQL Azure admite un subconjunto de características de SQL Server, quizás tenga que realizar algunas modificaciones al script antes de poder ejecutarlo en la nube.
2. Ejecute el script para crear el esquema en Base de datos de SQL Azure.

Puede utilizar el Asistente para generar y publicar scripts para transferir una base de datos de un equipo local a Base de datos de SQL Azure. El Asistente para generar y publicar scripts crea scripts de Transact-SQL para su base de datos local. El asistente los utiliza para publicar objetos de base de datos en Base de datos de SQL Azure.

## Migrar bases de datos a SQL Azure mediante Microsoft Sync Framework 2.1

Microsoft Sync Framework 2.1 proporciona capacidades de sincronización entre servidores locales y SQL Azure, así como entre dos o más bases de datos SQL Azure en el mismo centro de datos o en centros de datos diferentes. Con Sync Framework 2.1, puede extender el esquema y los datos dentro de su base de datos SQL Server o SQL Azure a centros de datos SQL Azure de todo el mundo para proporcionar acceso a datos disponibles geográficamente.

## Migrar bases de datos a SQL Azure mediante el Asistente para migración de SQL Azure

El Asistente para migración de SQL Azure le guía por la selección de objetos SQL, crea scripts SQL adecuados para Base de datos de SQL Azure, y le permite migrar datos entre SQL Server 2005 o 2008 local y servidores SQL Azure, así como entre dos o más bases de datos SQL Azure en el mismo centro de datos o en centros de datos diferentes.

## MIGRAR UNA BASE DE DATOS A SQL AZURE MEDIANTE EL ASISTENTE PARA GENERAR SCRIPTS

Al implementar una aplicación en Microsoft Base de datos de SQL Azure, puede que necesite migrar una base de datos de una instancia local de SQL Server a SQL Azure.

Base de datos de SQL Azure admite un subconjunto del lenguaje Transact-SQL. Debe modificar el script generado para incluir solo las instrucciones de Transact-SQL admitidas antes de implementar la base de datos en SQL Azure.

## Crear la base de datos de ejemplo SCHOOL

1. En SQL Server Management Studio, en el menú **Archivo**, seleccione **Nuevo** y, a continuación, haga clic en **Consulta de motor de base de datos**.
2. En el cuadro de diálogo **Conectarse al motor de base de datos**, escriba la información de su instancia local de SQL Server y, a continuación, haga clic en **Conectar**.

3. Pegue el script de Transact-SQL incluido al final de este tema en la ventana de consulta y, a continuación, haga clic en **Ejecutar**.

## Crear el script de Transact-SQL

1. En el Explorador de objetos, haga clic con el botón secundario en la base de datos **School**, seleccione **Tareas** y, a continuación, haga clic en **Generar scripts**.
2. En el cuadro de diálogo **Asistente para generar y publicar scripts**, haga clic en **Siguiente** para ir al paso **Elegir objetos**. Seleccione **Incluir en el script toda la base de datos y todos los objetos de base de datos** y, a continuación, haga clic en **Siguiente**.
3. En **Establecer opciones de scripting**, establezca las siguientes opciones:
  - Establezca **Tipo de salida** como **Guardar scripts en una ubicación específica**. Seleccione **Guardar en el archivo**. Haga clic en **Un solo archivo**. Escriba el nombre y la ubicación del archivo en **Nombre de archivo**. Haga clic en **Opciones avanzadas**.
  - En **Opciones de scripting avanzadas**, establezca la opción "Script para el tipo de motor de base de datos" como "Base de datos de SQL Azure", establezca "Convertir UDDTs en tipos base" como "True" y establezca "Tipos de datos que se deben incluir en el script" como "Esquema y datos". Haga clic en **Aceptar**.
4. Haga clic sucesivamente en **Siguiente**, en **Siguiente** y en **Finalizar**.

## Ejecutar el script en SQL Azure

1. Cree la base de datos School en el servidor SQL Azure a través del Portal de administración de la plataforma Windows Azure o ejecutando el siguiente comando Transact-SQL:

```
CREATE DATABASE School
```

2. En SQL Server Management Studio, abra el menú **Consulta**, seleccione **Conexión** y, a continuación, seleccione **Cambiar conexión**.
3. Escriba el nombre del servidor SQL Azure y sus credenciales y, a continuación, haga clic en **Opciones >>**.

4. En la pestaña **Propiedades de conexión**, escriba **School** en el menú desplegable **Conectar con base de datos** y, a continuación, haga clic en **Conectar**.
5. Haga clic con el botón secundario en la base de datos **School** en SQL Azure y seleccione **Nueva consulta**.
6. En SQL Server Management Studio, en el menú **Archivo**, seleccione **Abrir** y, a continuación, haga clic en **Archivo**. En la ventana **Abrir archivo**, abra el archivo de script school de la ubicación que especificó anteriormente en el Asistente para generar y publicar scripts.
7. Presione F5 para ejecutar el script.

## Definición de Transact-SQL de la base de datos School

Ejecute el script que se presenta a continuación para crear la base de datos School en la instancia local de SQL Server.

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

USE [master];
GO

IF EXISTS (SELECT * FROM sys.databases WHERE name = 'School')
    DROP DATABASE School;
GO

-- Create the School database.
CREATE DATABASE School;
GO

-- Specify a simple recovery model
-- to keep the log growth to a minimum.
ALTER DATABASE School
    SET RECOVERY SIMPLE;
GO

USE School;
GO

-- Create the Department table.
IF NOT EXISTS (SELECT * FROM sys.objects
    WHERE object_id = OBJECT_ID(N'[dbo].[Department]')
    AND type in (N'U'))

```

```

BEGIN
CREATE TABLE [dbo].[Department](
    [DepartmentID] [int] NOT NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Budget] [money] NOT NULL,
    [StartDate] [datetime] NOT NULL,
    [Administrator] [int] NULL,
    CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED
(
    [DepartmentID] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

-- Create the Person table.
IF NOT EXISTS (SELECT * FROM sys.objects
    WHERE object_id = OBJECT_ID(N'[dbo].[Person]')
    AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[Person](
    [PersonID] [int] IDENTITY(1,1) NOT NULL,
    [LastName] [nvarchar](50) NOT NULL,
    [FirstName] [nvarchar](50) NOT NULL,
    [HireDate] [datetime] NULL,
    [EnrollmentDate] [datetime] NULL,
    CONSTRAINT [PK_School.Student] PRIMARY KEY CLUSTERED
(
    [PersonID] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

-- Create the OnsiteCourse table.
IF NOT EXISTS (SELECT * FROM sys.objects
    WHERE object_id = OBJECT_ID(N'[dbo].[OnsiteCourse]')
    AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[OnsiteCourse](
    [CourseID] [int] NOT NULL,
    [Location] [nvarchar](50) NOT NULL,
    [Days] [nvarchar](50) NOT NULL,
    [Time] [smalldatetime] NOT NULL,
    CONSTRAINT [PK_OnsiteCourse] PRIMARY KEY CLUSTERED
(
    [CourseID] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

```

## MICROSOFT SQL AZURE Y LA NUBE

```
-- Create the OnlineCourse table.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[OnlineCourse]')
               AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[OnlineCourse] (
    [CourseID] [int] NOT NULL,
    [URL] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_OnlineCourse] PRIMARY KEY CLUSTERED
    (
        [CourseID] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
```

```
--Create the StudentGrade table.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[StudentGrade]')
               AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[StudentGrade] (
    [EnrollmentID] [int] IDENTITY(1,1) NOT NULL,
    [CourseID] [int] NOT NULL,
    [StudentID] [int] NOT NULL,
    [Grade] [decimal](3, 2) NULL,
    CONSTRAINT [PK_StudentGrade] PRIMARY KEY CLUSTERED
    (
        [EnrollmentID] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
```

```
-- Create the CourseInstructor table.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id =
OBJECT_ID(N'[dbo].[CourseInstructor]')
               AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[CourseInstructor] (
    [CourseID] [int] NOT NULL,
    [PersonID] [int] NOT NULL,
    CONSTRAINT [PK_CourseInstructor] PRIMARY KEY CLUSTERED
    (
        [CourseID] ASC,
        [PersonID] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
```

```

-- Create the Course table.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[Course]')
               AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[Course] (
    [CourseID] [int] NOT NULL,
    [Title] [nvarchar](100) NOT NULL,
    [Credits] [int] NOT NULL,
    [DepartmentID] [int] NOT NULL,
    CONSTRAINT [PK_School.Course] PRIMARY KEY CLUSTERED
    (
        [CourseID] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

-- Create the OfficeAssignment table.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id =
OBJECT_ID(N'[dbo].[OfficeAssignment]')
               AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[OfficeAssignment] (
    [InstructorID] [int] NOT NULL,
    [Location] [nvarchar](50) NOT NULL,
    [Timestamp] [timestamp] NOT NULL,
    CONSTRAINT [PK_OfficeAssignment] PRIMARY KEY CLUSTERED
    (
        [InstructorID] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

-- Define the relationship between OnsiteCourse and Course.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
               WHERE object_id = OBJECT_ID(N'[dbo].[FK_OnsiteCourse_Course]')
               AND parent_object_id = OBJECT_ID(N'[dbo].[OnsiteCourse]'))
ALTER TABLE [dbo].[OnsiteCourse] WITH CHECK ADD
    CONSTRAINT [FK_OnsiteCourse_Course] FOREIGN KEY([CourseID])
REFERENCES [dbo].[Course] ([CourseID])
GO
ALTER TABLE [dbo].[OnsiteCourse] CHECK
    CONSTRAINT [FK_OnsiteCourse_Course]
GO

-- Define the relationship between OnlineCourse and Course.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
               WHERE object_id = OBJECT_ID(N'[dbo].[FK_OnlineCourse_Course]')
               AND parent_object_id = OBJECT_ID(N'[dbo].[OnlineCourse]'))

```

## MICROSOFT SQL AZURE Y LA NUBE

```
ALTER TABLE [dbo].[OnlineCourse] WITH CHECK ADD
    CONSTRAINT [FK_OnlineCourse_Course] FOREIGN KEY([CourseID])
REFERENCES [dbo].[Course] ([CourseID])
GO
ALTER TABLE [dbo].[OnlineCourse] CHECK
    CONSTRAINT [FK_OnlineCourse_Course]
GO

-- Define the relationship between StudentGrade and Course.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_StudentGrade_Course]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[StudentGrade]'))
ALTER TABLE [dbo].[StudentGrade] WITH CHECK ADD
    CONSTRAINT [FK_StudentGrade_Course] FOREIGN KEY([CourseID])
REFERENCES [dbo].[Course] ([CourseID])
GO
ALTER TABLE [dbo].[StudentGrade] CHECK
    CONSTRAINT [FK_StudentGrade_Course]
GO

--Define the relationship between StudentGrade and Student.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_StudentGrade_Student]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[StudentGrade]'))
ALTER TABLE [dbo].[StudentGrade] WITH CHECK ADD
    CONSTRAINT [FK_StudentGrade_Student] FOREIGN KEY([StudentID])
REFERENCES [dbo].[Person] ([PersonID])
GO
ALTER TABLE [dbo].[StudentGrade] CHECK
    CONSTRAINT [FK_StudentGrade_Student]
GO

-- Define the relationship between CourseInstructor and Course.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_CourseInstructor_Course]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[CourseInstructor]'))
ALTER TABLE [dbo].[CourseInstructor] WITH CHECK ADD
    CONSTRAINT [FK_CourseInstructor_Course] FOREIGN KEY([CourseID])
REFERENCES [dbo].[Course] ([CourseID])
GO
ALTER TABLE [dbo].[CourseInstructor] CHECK
    CONSTRAINT [FK_CourseInstructor_Course]
GO

-- Define the relationship between CourseInstructor and Person.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_CourseInstructor_Person]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[CourseInstructor]'))
ALTER TABLE [dbo].[CourseInstructor] WITH CHECK ADD
    CONSTRAINT [FK_CourseInstructor_Person] FOREIGN KEY([PersonID])
REFERENCES [dbo].[Person] ([PersonID])
GO
```

```

ALTER TABLE [dbo].[CourseInstructor] CHECK
    CONSTRAINT [FK_CourseInstructor_Person]
GO

-- Define the relationship between Course and Department.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_Course_Department]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[Course]'))
ALTER TABLE [dbo].[Course] WITH CHECK ADD
    CONSTRAINT [FK_Course_Department] FOREIGN KEY([DepartmentID])
REFERENCES [dbo].[Department] ([DepartmentID])
GO
ALTER TABLE [dbo].[Course] CHECK CONSTRAINT [FK_Course_Department]
GO

--Define the relationship between OfficeAssignment and Person.
IF NOT EXISTS (SELECT * FROM sys.foreign_keys
    WHERE object_id = OBJECT_ID(N'[dbo].[FK_OfficeAssignment_Person]')
    AND parent_object_id = OBJECT_ID(N'[dbo].[OfficeAssignment]'))
ALTER TABLE [dbo].[OfficeAssignment] WITH CHECK ADD
    CONSTRAINT [FK_OfficeAssignment_Person] FOREIGN KEY([InstructorID])
REFERENCES [dbo].[Person] ([PersonID])
GO
ALTER TABLE [dbo].[OfficeAssignment] CHECK
    CONSTRAINT [FK_OfficeAssignment_Person]
GO

-- Create InsertOfficeAssignment stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
    WHERE object_id =
OBJECT_ID(N'[dbo].[InsertOfficeAssignment]')
    AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[InsertOfficeAssignment]
    @InstructorID int,
    @Location nvarchar(50)
AS
INSERT INTO dbo.OfficeAssignment (InstructorID,
Location)
VALUES (@InstructorID, @Location);
IF @@ROWCOUNT > 0
BEGIN
        SELECT [Timestamp] FROM OfficeAssignment
        WHERE InstructorID=@InstructorID;
END
'
END
GO

--Create the UpdateOfficeAssignment stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects

```

## MICROSOFT SQL AZURE Y LA NUBE

```
        WHERE object_id =
OBJECT_ID(N'[dbo].[UpdateOfficeAssignment]')
        AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[UpdateOfficeAssignment]
    @InstructorID int,
    @Location nvarchar(50),
    @OrigTimestamp timestamp
AS
UPDATE OfficeAssignment SET Location=@Location
WHERE InstructorID=@InstructorID AND
[Timestamp]=@OrigTimestamp;
IF @@ROWCOUNT > 0
BEGIN
        SELECT [Timestamp] FROM OfficeAssignment
        WHERE InstructorID=@InstructorID;
END
'
END
GO

-- Create the DeleteOfficeAssignment stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
        WHERE object_id =
OBJECT_ID(N'[dbo].[DeleteOfficeAssignment]')
        AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[DeleteOfficeAssignment]
    @InstructorID int
AS
DELETE FROM OfficeAssignment
WHERE InstructorID=@InstructorID;
'
END
GO

-- Create the DeletePerson stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
        WHERE object_id = OBJECT_ID(N'[dbo].[DeletePerson]')
        AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[DeletePerson]
    @PersonID int
AS
DELETE FROM Person WHERE PersonID = @PersonID;
'
END
GO
```

```

-- Create the UpdatePerson stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[UpdatePerson]')
               AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[UpdatePerson]
    @PersonID int,
    @LastName nvarchar(50),
    @FirstName nvarchar(50),
    @HireDate datetime,
    @EnrollmentDate datetime
AS
UPDATE Person SET LastName=@LastName,
                  FirstName=@FirstName,
                  HireDate=@HireDate,
                  EnrollmentDate=@EnrollmentDate
WHERE PersonID=@PersonID;
'
END
GO

-- Create the InsertPerson stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[InsertPerson]')
               AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[InsertPerson]
    @LastName nvarchar(50),
    @FirstName nvarchar(50),
    @HireDate datetime,
    @EnrollmentDate datetime
AS
INSERT INTO dbo.Person (LastName,
                       FirstName,
                       HireDate,
                       EnrollmentDate)
VALUES (@LastName,
        @FirstName,
        @HireDate,
        @EnrollmentDate);
SELECT SCOPE_IDENTITY() as NewPersonID;
'
END
GO

-- Create GetStudentGrades stored procedure.
IF NOT EXISTS (SELECT * FROM sys.objects
               WHERE object_id = OBJECT_ID(N'[dbo].[GetStudentGrades]')
               AND type in (N'P', N'PC'))
BEGIN

```

## MICROSOFT SQL AZURE Y LA NUBE

```
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[GetStudentGrades]
    @StudentID int
AS
    SELECT EnrollmentID, Grade, CourseID, StudentID FROM
dbo.StudentGrade
    WHERE StudentID = @StudentID
'
END
GO

-- Insert data into the Person table.
USE School
GO
SET IDENTITY_INSERT dbo.Person ON
GO
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (1, 'Abercrombie', 'Kim', '1995-03-11', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (2, 'Barzdukas', 'Gytis', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (3, 'Justice', 'Peggy', null, '2001-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (4, 'Fakhouri', 'Fadi', '2002-08-06', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (5, 'Harui', 'Roger', '1998-07-01', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (6, 'Li', 'Yan', null, '2002-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (7, 'Norman', 'Laura', null, '2003-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (8, 'Olivotto', 'Nino', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (9, 'Tang', 'Wayne', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (10, 'Alonso', 'Meredith', null, '2002-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (11, 'Lopez', 'Sophia', null, '2004-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (12, 'Browning', 'Meredith', null, '2000-09-01');
```

```

INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (13, 'Anand', 'Arturo', null, '2003-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (14, 'Walker', 'Alexandra', null, '2000-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (15, 'Powell', 'Carson', null, '2004-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (16, 'Jai', 'Damien', null, '2001-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (17, 'Carlson', 'Robyn', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (18, 'Zheng', 'Roger', '2004-02-12', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (19, 'Bryant', 'Carson', null, '2001-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (20, 'Suarez', 'Robyn', null, '2004-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (21, 'Holt', 'Roger', null, '2004-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (22, 'Alexander', 'Carson', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (23, 'Morgan', 'Isaiah', null, '2001-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (24, 'Martin', 'Randall', null, '2005-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (25, 'Kapoor', 'Candace', '2001-01-15', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (26, 'Rogers', 'Cody', null, '2002-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (27, 'Serrano', 'Stacy', '1999-06-01', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (28, 'White', 'Anthony', null, '2001-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (29, 'Griffin', 'Rachel', null, '2004-09-01');

```

## MICROSOFT SQL AZURE Y LA NUBE

```
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (30, 'Shan', 'Alicia', null, '2003-09-01');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (31, 'Stewart', 'Jasmine', '1997-10-12', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (32, 'Xu', 'Kristen', '2001-7-23', null);
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (33, 'Gao', 'Erica', null, '2003-01-30');
INSERT INTO dbo.Person (PersonID, LastName, FirstName, HireDate,
EnrollmentDate)
VALUES (34, 'Van Houten', 'Roger', '2000-12-07', null);
GO
SET IDENTITY_INSERT dbo.Person OFF
GO

-- Insert data into the Department table.
INSERT INTO dbo.Department (DepartmentID, [Name], Budget, StartDate,
Administrator)
VALUES (1, 'Engineering', 350000.00, '2007-09-01', 2);
INSERT INTO dbo.Department (DepartmentID, [Name], Budget, StartDate,
Administrator)
VALUES (2, 'English', 120000.00, '2007-09-01', 6);
INSERT INTO dbo.Department (DepartmentID, [Name], Budget, StartDate,
Administrator)
VALUES (4, 'Economics', 200000.00, '2007-09-01', 4);
INSERT INTO dbo.Department (DepartmentID, [Name], Budget, StartDate,
Administrator)
VALUES (7, 'Mathematics', 250000.00, '2007-09-01', 3);
GO

-- Insert data into the Course table.
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (1050, 'Chemistry', 4, 1);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (1061, 'Physics', 4, 1);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (1045, 'Calculus', 4, 7);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (2030, 'Poetry', 2, 2);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (2021, 'Composition', 3, 2);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (2042, 'Literature', 4, 2);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (4022, 'Microeconomics', 3, 4);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (4041, 'Macroeconomics', 3, 4);
```

```

INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (4061, 'Quantitative', 2, 4);
INSERT INTO dbo.Course (CourseID, Title, Credits, DepartmentID)
VALUES (3141, 'Trigonometry', 4, 7);
GO

```

```

-- Insert data into the OnlineCourse table.
INSERT INTO dbo.OnlineCourse (CourseID, URL)
VALUES (2030, 'http://www.fineartschool.net/Poetry');
INSERT INTO dbo.OnlineCourse (CourseID, URL)
VALUES (2021, 'http://www.fineartschool.net/Composition');
INSERT INTO dbo.OnlineCourse (CourseID, URL)
VALUES (4041, 'http://www.fineartschool.net/Macroeconomics');
INSERT INTO dbo.OnlineCourse (CourseID, URL)
VALUES (3141, 'http://www.fineartschool.net/Trigonometry');

```

```

--Insert data into OnsiteCourse table.
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (1050, '123 Smith', 'MTWH', '11:30');
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (1061, '234 Smith', 'TWHF', '13:15');
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (1045, '121 Smith', 'MWHF', '15:30');
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (4061, '22 Williams', 'TH', '11:15');
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (2042, '225 Adams', 'MTWH', '11:00');
INSERT INTO dbo.OnsiteCourse (CourseID, Location, Days, [Time])
VALUES (4022, '23 Williams', 'MWF', '9:00');

```

```

-- Insert data into the CourseInstructor table.
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (1050, 1);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (1061, 31);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (1045, 5);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (2030, 4);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (2021, 27);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (2042, 25);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (4022, 18);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (4041, 32);
INSERT INTO dbo.CourseInstructor(CourseID, PersonID)
VALUES (4061, 34);
GO

```

```

--Insert data into the OfficeAssignment table.

```

## MICROSOFT SQL AZURE Y LA NUBE

```
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (1, '17 Smith');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (4, '29 Adams');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (5, '37 Williams');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (18, '143 Smith');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (25, '57 Adams');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (27, '271 Williams');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (31, '131 Smith');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (32, '203 Williams');
INSERT INTO dbo.OfficeAssignment(InstructorID, Location)
VALUES (34, '213 Smith');

-- Insert data into the StudentGrade table.
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2021, 2, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2030, 2, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2021, 3, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2030, 3, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2021, 6, 2.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2042, 6, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2021, 7, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2042, 7, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2021, 8, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (2042, 8, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 9, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 10, null);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 11, 2.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 12, null);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4061, 12, null);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 14, 3);
```

```

INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 13, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4061, 13, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 14, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 15, 2.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 16, 2);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 17, null);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 19, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4061, 20, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4061, 21, 2);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 22, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4041, 22, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4061, 22, 2.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (4022, 23, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1045, 23, 1.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 24, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 25, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1050, 26, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 26, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 27, 3);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1045, 28, 2.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1050, 28, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 29, 4);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1050, 30, 3.5);
INSERT INTO dbo.StudentGrade (CourseID, StudentID, Grade)
VALUES (1061, 30, 4);
GO

```

## MOVER DATOS A SQL AZURE

Base de datos de SQL Azure admite el uso de SQL Server Integration Services. SQL Server Integration Services es una manera cómoda de mover datos hacia y desde Base de datos de SQL Azure. O bien, puede transferir datos a Base de datos de SQL Azure mediante la utilidad de copia masiva (bcp.exe).

### SQL Server Integration Services

Puede transferir datos a Base de datos de SQL Azure mediante SQL Server 2008 Integration Services. En SQL Server 2008 R2 o posterior, el Asistente para importación y exportación proporciona compatibilidad con Base de datos de SQL Azure. Puede utilizar esta herramienta para migrar bases de datos locales a Base de datos de SQL Azure.

### La utilidad bcp

La utilidad **bcp** se puede usar para importar un número elevado de filas nuevas en tablas de SQL Server o para exportar datos de tablas a archivos de datos.

# COPIAR BASES DE DATOS Y SUPERVISAR SQL AZURE

## COPIAR BASES DE DATOS EN SQL AZURE

Una vez que haya creado una base de datos en SQL Azure, puede hacer una copia de seguridad de la misma copiándola a una nueva base de datos en SQL Azure. La nueva base de datos que se crea a partir del proceso de copia es una base de datos normal una vez completada la copia que funciona totalmente y es independiente de la base de datos de origen. Además, la nueva base de datos tendrá la misma edición y tamaño máximo que la base de datos de origen.

La característica de copia de base de datos habilita muchos de los escenarios de copia de seguridad con los que puede estar familiarizado para las bases de datos locales:

- **Copia de seguridad de datos de aplicaciones:** copie la base de datos de aplicación a una nueva base de datos para ayudar a protegerla frente a errores del usuario y de la aplicación. Para recuperar el estado de la base de datos de aplicación, cambie el nombre de la nueva base de datos por el nombre de la base de datos de aplicación. O bien, recupere un subconjunto concreto de datos de la nueva base de datos mediante consultas de Transact-SQL.

- **Desarrollo y prueba de aplicaciones:** copie la base de datos de producción a una nueva base de datos que se puede emplear para desarrollo y pruebas.
- **Actualización de la aplicación:** copie la base de datos de aplicación a una nueva base de datos antes de realizar actualizaciones importantes de la aplicación. Si se produce un error durante la actualización, recupere el estado anterior de la base de datos cambiando el nombre de la nueva base de datos al nombre de la base de datos de aplicación.

## Copiar una base de datos

Las bases de datos se copian de forma asincrónica, por lo que no se necesita una conexión al servidor SQL Azure mientras dura todo el proceso. Puede copiar una base de datos iniciando sesión en la base de datos **master** del servidor de destino y ejecutando la instrucción CREATE DATABASE de Transact-SQL con la cláusula AS COPY OF. Después, puede supervisar el proceso de copia usando las vistas sys.dm\_database\_copies y sys.databases en el servidor de destino.

## Copia en el mismo servidor

Al copiar una base de datos para crear una nueva base de datos en el mismo servidor SQL Azure, se pueden usar los mismos inicios de sesión en ambas bases de datos. La entidad de seguridad que emplea para copiar la base de datos se convierte en el propietario de la base de datos (DBO) en la nueva base de datos cuando se crea.

Los pasos para copiar su base de datos a una nueva base de datos en el mismo servidor son estos:

- Inicie sesión en la base de datos **master** utilizando el inicio de sesión de entidad de seguridad a nivel de servidor o el inicio de sesión que creó la base de datos que desea copiar. Los inicios de sesión que no son la entidad de seguridad a nivel de servidor deben ser miembros del rol dbmanager para poder copiar bases de datos.
- Empiece a copiar la base de datos de origen con la instrucción CREATE DATABASE de SQL Azure. Al ejecutar esta instrucción, se inicia el proceso de copia de la base de datos. Puesto que se trata de un proceso asincrónico, la instrucción CREATE DATABASE terminará antes de que se complete la copia de la base de datos.

- Supervise el proceso de copia consultando las vistas `sys.databases` y `sys.dm_database_copies`.
  - Mientras la copia esté en curso, la columna `state_desc` de la vista `sys.databases` para la nueva base de datos mostrará `COPYING`.
  - Si se produce un error en la copia, la columna `state_desc` de la vista `sys.databases` para la nueva base de datos mostrará `SUSPECT`. En este caso, ejecute de nuevo la instrucción `DROP` en la nueva base de datos e inténtelo de nuevo más tarde.
  - Si la copia se realiza correctamente, la columna `state_desc` de la vista `sys.databases` para la nueva base de datos mostrará `ONLINE`. En este caso, la copia se ha completado y la nueva base de datos es una base de datos normal, que se puede modificar independientemente de la base de datos de origen.
- (Opcional) Si decide cancelar la copia mientras está en curso, ejecute la instrucción `DROP DATABASE` en la nueva base de datos. La ejecución de la instrucción `DROP DATABASE` en la base de datos de origen también cancelará el proceso de copia.

## Ejemplo

En el ejemplo de código siguiente se inicia el proceso de copia de la base de datos, copiando la base de datos `Database1A` a una nueva base de datos denominada `Database1B` en el mismo servidor SQL Azure.

```
-- execute on the master database
-- start copying
CREATE DATABASE Database1B AS COPY OF Database1A
```

En el ejemplo de código siguiente se supervisa el proceso de copia de la base de datos consultando las vistas `sys.databases` y `sys.dm_database_copies` para obtener información sobre la nueva base de datos denominada `Database1B`.

```
-- execute on the master database
-- retrieve state of the new database, Database1B
select name, state, state_desc from sys.databases
where name = 'Database1B'

-- retrieve copying details
select * from sys.dm_database_copies
where database_id = DB_ID('Database1B')
```

## COPIA ENTRE SERVIDORES

También puede copiar una base de datos entre dos servidores SQL Azure diferentes que estén en la misma subregión o en el mismo centro de datos. Como la nueva base de datos se crea en un servidor SQL Azure diferente, está asociada a una base de datos **master** distinta. Todos los usuarios de la nueva base de datos mantienen los permisos que tenían en la base de datos de origen. La entidad de seguridad que emplee para copiar la base de datos se convertirá en el DBO en la nueva base de datos cuando se cree y se le asignará un nuevo identificador de seguridad (SID).

Una vez completado el proceso de copia entre servidores, use el inicio de sesión DBO y la instrucción ALTER USER para asignar los usuarios de la nueva base de datos a los inicios de sesión en el nuevo servidor SQL Azure. Por ejemplo: ALTER USER userName WITH LOGIN='loginName'.

Para copiar su base de datos a una nueva base de datos en un servidor diferente se tendrá en cuenta lo siguiente:

- Inicie sesión en la base de datos **master** del servidor de destino, el servidor SQL Azure donde se creará la nueva base de datos. Debe utilizar un inicio de sesión que tenga el mismo nombre y la misma contraseña que el propietario de la base de datos (DBO) de la base de datos de origen en el servidor SQL Azure de origen. El inicio de sesión en el servidor de destino también debe ser miembro del rol dbmanager o debe ser el inicio de sesión de entidad de seguridad a nivel de servidor.
- Empiece a copiar la base de datos de origen con la instrucción CREATE DATABASE de SQL Azure. Al ejecutar esta instrucción en el servidor de destino se inicia el proceso de copia de la base de datos. Puesto que se trata de un proceso asíncrono, la instrucción CREATE DATABASE terminará antes de que se complete la copia de la base de datos.
- Supervise el proceso de copia consultando las vistas sys.databases y sys.dm\_database\_copies en el servidor de destino.
  - Mientras la copia esté en curso, la columna state\_desc de la vista sys.databases para la nueva base de datos mostrará COPYING.

- Si se produce un error en la copia, la columna `state_desc` de la vista `sys.databases` para la nueva base de datos mostrará `SUSPECT`. En este caso, ejecute de nuevo la instrucción `DROP` en la nueva base de datos e inténtelo de nuevo más tarde.
- Si la copia se realiza correctamente, la columna `state_desc` de la vista `sys.databases` para la nueva base de datos mostrará `ONLINE`. En este caso, la copia se ha completado y la nueva base de datos es una base de datos normal, que se puede modificar independientemente de la base de datos de origen.
- Si decide cancelar la copia mientras está en curso, ejecute la instrucción `DROP DATABASE` en la nueva base de datos. La ejecución de la instrucción `DROP DATABASE` en la base de datos de origen también cancelará el proceso de copia.
- Una vez que la nueva base de datos está Online en el servidor de destino, utilice la instrucción `ALTER USER` para volver a asignar los usuarios de la nueva base de datos a inicios de sesión en el servidor de destino. Todos los usuarios de la nueva base de datos mantienen los permisos que tenían en la base de datos de origen. El usuario que inició la copia de la base de datos se convierte en el propietario de la nueva base de datos y se le asigna un nuevo identificador de seguridad (SID). Después de que se complete la copia correctamente y antes de volver a asignar otros usuarios, solo el inicio de sesión que inició la copia, el propietario de la base de datos (DBO), podrá iniciar sesión en la nueva base de datos.

## Ejemplo

En el ejemplo de código siguiente se inicia el proceso de copia de la base de datos desde `Server2` y se muestra cómo copiar la base de datos `Database1A` del servidor SQL Azure denominado `Server1` a una nueva base de datos denominada `Database2A` en `Server2`.

```
-- execute on the master database of Server2
-- start copying from Server1 to Server2
```

```
CREATE DATABASE Database2A AS COPY OF Server1.Database1A
```

En el ejemplo de código siguiente se supervisa el proceso de copia de la base de datos consultando las vistas `sys.databases` y `sys.dm_database_copies` en `Server2` para obtener información sobre la nueva base de datos denominada `Database2A`.

```
-- execute on the master database of Server2
-- retrieve state of the new database, Database2A
select name, state, state_desc from sys.databases
where name = 'Database2A'

-- retrieve copying details
select * from sys.dm_database_copies
where database_id = DB_ID('Database2A')
```

En el código siguiente se vuelven a asignar usuarios de la nueva base de datos a inicios de sesión en el servidor de destino, asignando los usuarios de Database2A denominados user1, user2 y user3 a los inicios de sesión de Server2 denominados login1, login2 y login3.

```
-- execute on the new Server2 database, named Database2A
-- re-map database users to logins on the destination server
ALTER USER user1 WITH LOGIN = login1
ALTER USER user2 WITH LOGIN = login2
ALTER USER user3 WITH LOGIN = login3
```

## PERMISOS

Para copiar una base de datos en SQL Azure, su inicio de sesión necesita los permisos siguientes:

- **En ambos servidores:** el inicio de sesión debe tener el mismo nombre de inicio de sesión y la misma contraseña en ambos servidores SQL Azure.
- **En el servidor de destino:** el inicio de sesión debe ser miembro del rol dbmanager de nivel de servidor. Nota: la entidad de seguridad a nivel de servidor de su servidor SQL Azure no es miembro del rol dbmanager, pero tiene los mismos permisos automáticamente. Para obtener más información sobre cómo administrar inicios de sesión en SQL Azure, vea Administrar bases de datos e inicios de sesión en SQL Azure.
- **En el servidor de origen:** el inicio de sesión debe ser el DBO de la base de datos de origen. Solo el inicio de sesión que creó la base de datos de origen, el DBO, puede copiar esa base de datos a otra base de datos en el mismo servidor o en un servidor diferente.

Cuando se cumplen estos requisitos de permisos, su inicio de sesión puede ejecutar instrucciones ALTER DATABASE y DROP DATABASE en la nueva base de datos.

SQL Azure no permite cambiar el propietario de la base de datos usando la instrucción ALTER AUTHORIZATION ON DATABASE. Como en SQL Server, puede crear usuarios adicionales dentro de la base de datos en SQL Azure y agregarlos al rol de nivel de base de datos db\_owner. Estos usuarios adicionales se pueden vincular a un inicio de sesión distinto del inicio de sesión que creó la base de datos inicialmente. Sin embargo, estos usuarios adicionales no pueden realizar una copia de base de datos.

## INTERVALOS

La carga de trabajo de copia de base de datos afectará al rendimiento de los servidores SQL Azure implicados y se administra de manera similar a otras cargas de trabajo de base de datos. Por tanto, es posible que la carga de trabajo de copia tarde mucho tiempo en completarse (en caso de que se ralentice debido a un uso excesivo de recursos).

Si se produce un error en la copia de la base de datos, reinicie el proceso de copia cuando haya poca demanda en la base de datos de origen y suspenda otras cargas de trabajo en la base de datos de origen hasta que el proceso de copia se complete.

## INICIAR EL PROCESO DE COPIA

Inicie la copia de la base de datos de origen con la instrucción Base de datos de SQL Azure CREATE DATABASE. Debe estar conectado a la base de datos **master** con un inicio de sesión que cumpla los requisitos de permisos descritos en Permisos. Para iniciar la copia entre servidores, debe estar conectado a la base de datos **master** del servidor SQL Azure donde se creará la nueva base de datos: el servidor de destino.

```
CREATE DATABASE destination_database_name
AS COPY OF [source_server_name.]source_database_name
```

Al ejecutar esta instrucción, se inicia el proceso de copia de la base de datos. Si no se proporciona source\_server\_name, la base de datos de origen debe estar en el mismo servidor SQL Azure al que esté conectado su inicio de sesión. Puesto que se trata de un proceso asíncronico, la instrucción CREATE DATABASE terminará antes de que se complete la copia de la base de datos.

## Supervisar una copia

Una vez que comience la copia de base de datos, puede consultar las vistas `sys.databases` y `sys.dm_database_copies` en la base de datos **master** del servidor de destino para recuperar más información sobre el progreso de la copia. En el siguiente ejemplo de Transact-SQL se muestra cómo obtener el progreso de la copia para una nueva base de datos denominada `databaseBackup`.

```
-- execute on the master database
-- retrieve state of the new database
select name, state, state_desc from sys.databases
where name = 'databaseBackup'

-- retrieve copying details
select * from sys.dm_database_copies
where database_id = DB_ID('databaseBackup')
```

El cliente paga por las bases de datos que tiene, durante los días que las tiene. Si se produce un error durante la copia, quite la base de datos de destino inmediatamente.

## Coherencia transaccional

Una nueva base de datos creada a partir del proceso de copia es coherente transaccionalmente con la base de datos de origen en el momento en que se completa la copia.

Desde una perspectiva de copia de seguridad de base de datos, copiar una base de datos en SQL Azure con una programación periódica es similar a realizar copias de seguridad completas de una base de datos local de SQL Server. En cada repetición, después de que la copia se complete correctamente, puede quitar la base de datos que copió de la repetición anterior.

## DETENER EL PROCESO DE COPIA

Si desea detener el proceso de copia antes de que se complete, puede usar la instrucción `DROP DATABASE` para quitar las bases de datos de destino. Al quitar la base de datos de origen también se cancelará el proceso de copia. Para que cualquiera de estas opciones funcione, use un inicio de sesión que cumpla los requisitos de permisos descritos en Permisos.

## Una vez completada la copia

Cuando se complete la copia de base de datos, la nueva base de datos entrará en uno de dos estados, tal y como indica la columna `state_desc` de la vista `sys.databases`. Si la copia se realizó correctamente, la nueva base de datos entrará en el estado Online. Cuando la nueva base de datos está en el estado Online, es una base de datos normal, y se puede administrar y emplear independientemente de la base de datos de origen.

Si la copia no se realiza correctamente, la nueva base de datos entrará en el estado Suspect. Cuando está en el estado Suspect, la nueva base de datos no está disponible y no se puede obtener acceso a ella ni recuperarla. En este caso, quite la base de datos de destino e intente copiar la base de datos de origen de nuevo. Especifique la edición y el tamaño máximo de la base de datos al crearla. Una vez creada una base de datos en SQL Azure, puede hacer una copia de seguridad de ella copiándola a una nueva base de datos en SQL Azure. Cuando la copia se complete, la nueva base de datos tendrá la misma edición y el mismo tamaño máximo que la base de datos de origen.

## SUPERVISAR SQL AZURE MEDIANTE VISTAS DE ADMINISTRACIÓN DINÁMICA

Microsoft Base de datos de SQL Azure habilita un subconjunto de vistas de administración dinámica para diagnosticar los problemas de rendimiento, que podrían deberse a consultas bloqueadas o de ejecución prolongada, cuellos de botella de recursos, planes de consulta deficientes, etc. En este tema se proporciona información sobre cómo detectar problemas de rendimiento frecuentes usando las vistas de administración dinámica de Base de datos de SQL Azure.

Base de datos de SQL Azure admite parcialmente tres categorías de vistas de administración dinámica:

- Vistas de administración dinámica relacionadas con la base de datos.
- Vistas de administración dinámica relacionadas con ejecuciones.
- Vistas de administración dinámica relacionadas con transacciones.

En Base de datos de SQL Azure, para consultar una vista de administración dinámica se necesitan permisos `VIEW DATABASE STATE`. El permiso `VIEW DATABASE STATE` devuelve información sobre todos los objetos de la base de datos actual.

Para conceder el permiso **VIEW DATABASE STATE** a un usuario concreto de la base de datos, ejecute la consulta siguiente:

```
GRANT VIEW DATABASE STATE TO database_user;
```

En una instancia de SQL Server local, las vistas de administración dinámica devuelven información sobre el estado del servidor. En Base de datos de SQL Azure, devuelven información relativa a su base de datos lógica actual únicamente.

Al ejecutar las vistas `sys.dm_exec_sessions` y `sys.dm_exec_requests`, si el usuario tiene el permiso **VIEW DATABASE STATE** en la base de datos, verá todas las sesiones en ejecución de la base de datos; de lo contrario, el usuario solo verá la sesión actual.

## Calcular el tamaño de la base de datos

Se le cobra según la edición y la capacidad de sus bases de datos SQL Azure. Si el tamaño de la base de datos alcanza el valor de `MAXSIZE`, recibirá un código de error 40544. No puede insertar ni actualizar datos, ni crear objetos nuevos (como tablas, procedimientos almacenados, vistas y funciones) a menos que actualice el valor `MAXSIZE` de su base de datos o elimine datos. Para obtener más información, vea Cuentas y facturación en SQL Azure. La vista `sys.dm_db_partition_stats` devuelve información sobre el número de páginas y filas para cada partición de la base de datos, lo que se puede emplear para calcular el tamaño de la base de datos.

La consulta siguiente devuelve el tamaño de la base de datos (en megabytes):

```
-- Calculates the size of the database.
SELECT SUM(reserved_page_count)*8.0/1024
FROM sys.dm_db_partition_stats;
GO
```

La consulta siguiente devuelve el tamaño de los objetos individuales (en megabytes) de la base de datos:

```
-- Calculates the size of individual database objects.
SELECT sys.objects.name, SUM(reserved_page_count) * 8.0 / 1024
FROM sys.dm_db_partition_stats, sys.objects
WHERE sys.dm_db_partition_stats.object_id =
sys.objects.object_id
GROUP BY sys.objects.name;
GO
```

## Supervisar las conexiones

Puede usar la vista `sys.dm_exec_connections` (Base de datos de SQL Azure) para recuperar información sobre las conexiones establecidas con un servidor SQL Azure concreto y los detalles de cada conexión. Además, la vista `sys.dm_exec_sessions` es útil para recuperar información sobre todas las conexiones de usuario activas y las tareas internas.

La consulta siguiente recupera información sobre la conexión actual:

```
-- monitor connections
SELECT
    e.connection_id,
    s.session_id,
    s.login_name,
    s.last_request_end_time,
    s.cpu_time
FROM
    sys.dm_exec_sessions s
    INNER JOIN sys.dm_exec_connections e
    ON s.session_id = e.session_id
GO
```

## Supervisar el rendimiento de las consultas

Las consultas de ejecución lenta o prolongada pueden consumir muchos recursos del sistema. En esta sección se muestra cómo utilizar vistas de administración dinámica para detectar algunos problemas frecuentes de rendimiento de las consultas.

## Buscar las N consultas superiores

En el ejemplo siguiente se devuelve información acerca de las cinco primeras consultas clasificadas por el promedio de tiempo de CPU. Este ejemplo agrega las consultas según su hash de consulta para agrupar las consultas lógicamente equivalentes según su consumo acumulado de los recursos.

```
-- Find top 5 queries
SELECT TOP 5 query_stats.query_hash AS "Query Hash",
    SUM(query_stats.total_worker_time) /
SUM(query_stats.execution_count) AS "Avg CPU Time",
    MIN(query_stats.statement_text) AS "Statement Text"
FROM
```

```

        (SELECT QS.*,
        SUBSTRING(ST.text, (QS.statement_start_offset/2) + 1,
        ((CASE statement_end_offset
            WHEN -1 THEN DATALENGTH(st.text)
            ELSE QS.statement_end_offset END
            - QS.statement_start_offset)/2) + 1) AS
statement_text
        FROM sys.dm_exec_query_stats AS QS
        CROSS APPLY sys.dm_exec_sql_text(QS.sql_handle) as ST) as
query_stats
GROUP BY query_stats.query_hash
ORDER BY 2 DESC;
GO

```

## Supervisar consultas bloqueadas

Las consultas de ejecución lenta o prolongada pueden contribuir al consumo excesivo de recursos y ser la consecuencia de consultas bloqueadas. La causa del bloqueo puede ser un mal diseño de las aplicaciones, planes de consulta no válidos, la ausencia de índices útiles, etc. Puede utilizar la vista **sys.dm\_tran\_locks** para obtener información sobre la actividad de bloqueo actual en su base de datos SQL Azure.

## Supervisar planes de consulta

Un plan de consulta ineficaz también puede aumentar el consumo de la CPU. En el ejemplo siguiente se usa la vista **sys.dm\_exec\_query\_stats** para determinar qué consulta emplea más CPU acumulada.

```

-- Monitor query plans
SELECT
    highest_cpu_queries.plan_handle,
    highest_cpu_queries.total_worker_time,
    q.dbid,
    q.objectid,
    q.number,
    q.encrypted,
    q.[text]
FROM
    (SELECT TOP 50
        qs.plan_handle,
        qs.total_worker_time
    FROM
        sys.dm_exec_query_stats qs
    ORDER BY qs.total_worker_time desc) AS highest_cpu_queries
    CROSS APPLY sys.dm_exec_sql_text(plan_handle) AS q
ORDER BY highest_cpu_queries.total_worker_time desc

```

## FIREWALL DE SQL AZURE

Microsoft SQL Azure proporciona un servicio de base de datos relacional para Windows Azure y otras aplicaciones basadas en Internet. Para ayudar a proteger sus datos, el firewall de SQL Azure impide todo acceso a su servidor SQL Azure hasta que especifique qué equipos tienen permiso. El firewall concede acceso según la dirección IP de origen de cada solicitud.

Para configurar su firewall, cree reglas de firewall que especifiquen intervalos de direcciones IP aceptables. Esto se puede conseguir utilizando el Portal de administración de la plataforma Windows Azure o utilizando mediante programación las Operaciones en reglas de firewall expuestas por la API de REST de administración de bases de datos SQL Azure. O bien, después de haber establecido el acceso, puede usar mediante programación la base de datos maestra (**master**) para revisar y modificar la configuración del firewall.

En este tema se describe el firewall de SQL Azure y cómo puede definir su configuración para especificar qué clientes deben tener acceso a su servidor SQL Azure.

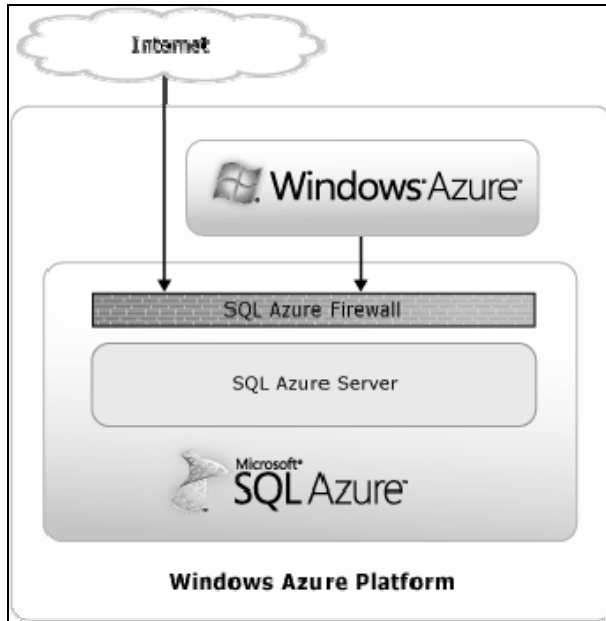
### Información general

Inicialmente, todo acceso a su servidor SQL Azure está bloqueado por el firewall de SQL Azure y los intentos de conexión procedentes de Internet o de Windows Azure no podrá alcanzar su servidor SQL Azure.

Para poder empezar a usar su servidor SQL Azure, debe ir al Portal de administración y especificar una o más configuraciones de firewall que permitan el acceso a su servidor SQL Azure.

Use la configuración del firewall para especificar qué intervalos de direcciones IP de Internet se permiten y si las aplicaciones Windows Azure pueden intentar o no conectarse a su servidor SQL Azure.

Los intentos de conexión desde Internet y Windows Azure deben atravesar primero el firewall de SQL Azure antes de poder alcanzar su servidor SQL Azure, como se muestra en el diagrama siguiente.



## Conectarse desde Internet

Cuando un equipo intenta conectarse a su servidor SQL Azure desde Internet, el firewall de SQL Azure comprueba la dirección IP de origen de la solicitud con el conjunto completo de configuraciones del firewall. Si la dirección IP de la solicitud no está dentro de uno de los intervalos especificados, el intento de conexión se bloqueará y no alcanzará el servidor SQL Azure.

Además de configurar el firewall de SQL Azure, puede que también necesite configurar el firewall en su red y en el equipo local. Para tener acceso a una base de datos SQL Azure desde su equipo, asegúrese de que el firewall de su red y del equipo local permita la comunicación TCP de salida en el puerto TCP 1433. (El servicio Base de datos de SQL Azure solo está disponible con el puerto TCP 1433).

## Conectarse desde Windows Azure

Cuando una aplicación de Windows Azure intenta conectarse a su servidor SQL Azure, el firewall de SQL Azure busca una configuración del firewall concreta que indica si se permiten las conexiones de Windows Azure.

Una configuración del firewall con una dirección inicial y final igual a 0.0.0.0 indica que se permiten las conexiones de Windows Azure. Si no se permite el intento de conexión, la solicitud no alcanzará el servidor SQL Azure.

En el Portal de administración, puede habilitar las conexiones de Windows Azure con una única casilla.

## Crear la primera regla de firewall

Para establecer conexión con su servidor SQL Azure por primera vez, la primera configuración del firewall debe especificarse utilizando el Portal de administración o utilizando mediante programación las Operaciones en reglas de firewall proporcionadas por la API de administración de bases de datos. Para empezar a configurar el firewall, en el Portal de administración, haga clic en la pestaña **Firewall Settings** (Configuración del firewall) en la página **Server Administration** (Administración de servidores). Si está utilizando el nuevo Portal de administración, haga clic en el servidor bajo su suscripción. En el panel derecho están los botones **Add** (Agregar), **Update** (Actualizar) y **Delete** (Eliminar) para administrar las reglas del firewall. Si las reglas del firewall y los botones no están visibles, haga clic en el botón **Firewall Rules** (Reglas del firewall) para alternar la vista de las reglas del firewall.

El Portal de administración también le permite quitar la configuración del firewall.

## La base de datos maestra

Después de haber usado el Portal de administración para crear una configuración del firewall que habilita la conexión a su servidor SQL Azure, puede utilizar el inicio de sesión de entidad de seguridad a nivel de servidor y la base de datos maestra (**master**) para ver y editar su configuración del firewall.

En la base de datos maestra (**master**), la configuración del firewall se conoce como reglas. La vista `sys.firewall_rules` muestra la configuración del firewall actual, y los procedimientos almacenados `sp_set_firewall_rule` y `sp_delete_firewall_rule` le permiten cambiar la configuración del firewall.

## Solucionar problemas del firewall

Tenga en cuenta lo siguiente cuando el acceso al servicio Base de datos de SQL Azure no se comporte de la manera esperada:

- **Configuración del firewall local:** antes de que su equipo pueda tener acceso a SQL Azure, quizás necesite crear una excepción del firewall en su equipo para el puerto TCP 1433.
- **Traducción de direcciones de red (NAT):** debido a NAT, la dirección IP usada por su equipo para conectarse a SQL Azure puede ser diferente de la dirección IP mostrada en los valores de configuración IP del equipo. Para ver qué dirección IP se está usando, utilice ese equipo para conectarse al Portal de administración y haga clic en la pestaña **Firewall Settings**. Al hacer clic en **Add Rule** (Agregar regla) o en **Edit Rule** (Editar regla), se mostrará su dirección IP en el cuadro de diálogo con la etiqueta **Your IP Address is** (Su dirección IP es).
- **Los cambios a la lista de permitidos no han surtido efecto todavía:** pueden transcurrir hasta cinco minutos para que los cambios a la configuración del firewall de SQL Azure surtan efecto.
- **El inicio de sesión no ha sido autorizado o se ha utilizado una contraseña incorrecta:** si un inicio de sesión no tiene permisos en el servidor SQL Azure o la contraseña utilizada es incorrecta, se denegará la conexión al servidor SQL Azure. Al crear una configuración del firewall, solo se proporciona a los clientes una oportunidad para intentar conectarse a su servidor SQL Azure; cada cliente debe suministrar las credenciales de seguridad necesarias.

## Configurar el firewall de SQL Azure

El servicio Microsoft Base de datos de SQL Azure impide el acceso a su servidor SQL Azure con el firewall de SQL Azure. Puede utilizar el Portal de administración de la plataforma Windows Azure o la base de datos **master** para revisar y editar la configuración del firewall. En este tema se describe cómo puede definir la configuración del firewall para especificar qué clientes deben tener acceso a su servidor SQL Azure.

Para conectarse por primera vez a su servidor SQL Azure, debe habilitar la conectividad a través del firewall mediante el Portal de administración.

Para *configurar el firewall con el Portal de administración*, se tendrá en cuenta lo siguiente:

- Conéctese al Portal de administración.
- Inicie sesión y haga clic en su suscripción.

- Vea la configuración del firewall correspondiente a su servidor SQL Azure haciendo clic en la pestaña **Firewall Settings** (Configuración del firewall) de la página **Server Administration** (Administración de servidores). Si está usando el nuevo Portal de administración, tendrá acceso a las reglas del firewall haciendo clic en el servidor bajo su suscripción. En el panel derecho están los botones **Add** (Agregar), **Update** (Actualizar) y **Delete** (Eliminar). Si las reglas del firewall y los botones no están visibles, haga clic en el botón **Firewall Rules** (Reglas del firewall) para alternar la vista de las reglas del firewall.
- Use la pestaña **Firewall Settings** para configurar el firewall de SQL Azure.
  - a. Habilite los intentos de conexión desde Windows Azure haciendo clic en **Allow Microsoft Services access to this server** (Permitir que los servicios de Microsoft tengan acceso a este servidor).
  - b. Agregue una nueva configuración del firewall para las conexiones basadas en Internet haciendo clic en **Add Rule** (Agregar regla). En el cuadro de diálogo, especifique un nombre único en el cuadro **Name** (Nombre) con el intervalo de direcciones IP correspondiente en los cuadros **From** (Desde) y **To** (Hasta). A continuación, seleccione **Submit** (Enviar).
  - c. Actualice una configuración del firewall existente para las conexiones basadas en Internet seleccionando el registro adecuado y haciendo clic en **Edit Rule** (Editar regla). Especifique el nuevo intervalo de direcciones IP en los cuadros **From** y **To**. A continuación, seleccione **Submit**.
- Quite una configuración del firewall existente seleccionando el registro adecuado y haciendo clic en **Delete Rule** (Eliminar regla).

Para *configurar el firewall con la base de datos maestra*, se tendrá en cuenta lo siguiente:

- Utilice el Portal de administración para confirmar que hay una configuración del firewall que permite a su equipo intentar la conexión al servidor SQL Azure. La dirección IP de su equipo debe estar dentro del intervalo de direcciones IP de una de las configuraciones del firewall.
- Conéctese a la base de datos **master** del servidor SQL Azure usando su inicio de sesión de entidad de seguridad a nivel de servidor.

- Ve a la configuración del firewall correspondiente a su servidor SQL Azure ejecutando la consulta: `select * from sys.firewall_rules`.
- Configure el firewall con los procedimientos almacenados `sp_set_firewall_rule` y `sp_delete_firewall_rule`.
  - a. Habilite los intentos de conexión desde Windows Azure usando el procedimiento almacenado `sp_set_firewall_rule` con los parámetros `start_ip_address` y `end_ip_address` establecidos en 0.0.0.0.
  - b. Agregue una nueva configuración del firewall para las conexiones basadas en Internet especificando un nombre único en el parámetro `name` del procedimiento almacenado `sp_set_firewall_rule`. Especifique la menor dirección IP deseada de ese intervalo con el parámetro `start_ip_address` y la mayor dirección IP deseada de ese intervalo con el parámetro `end_ip_address`. El parámetro `name` tiene el tipo de datos `nvarchar`, y los parámetros `start_ip_address` y `end_ip_address` tienen el tipo de datos `varchar`.
  - c. Actualice una configuración del firewall existente para las conexiones basadas en Internet especificando un nombre existente en el parámetro `name` del procedimiento almacenado `sp_set_firewall_rule`. Especifique el nuevo intervalo de direcciones IP con los parámetros `start_ip_address` y `end_ip_address`.
- Quite una configuración del firewall especificando nombre de la configuración no deseada en el parámetro `name` del procedimiento almacenado `sp_delete_firewall_rule`.

## Ejemplo

En este ejemplo se muestra cómo configurar el firewall con Transact-SQL. En primer lugar, debe utilizar el Portal de administración para permitir el acceso a la dirección IP de su equipo. A continuación, puede ver la configuración del firewall con el código siguiente:

```
-- view firewall settings
select * from sys.firewall_rules
```

Solo el inicio de sesión de entidad de seguridad a nivel de servidor, mientras está conectado a la base de datos **master**, puede configurar el firewall para su servidor SQL Azure.

Puede usar el procedimiento almacenado `sp_set_firewall_rule` para agregar o cambiar la configuración del firewall. Al agregar una configuración de firewall donde las direcciones IP inicial y final son iguales a 0.0.0.0, se permite al acceso a su servidor SQL Azure desde Windows Azure. Proporcione un valor para el parámetro `name` que le ayude a recordar para qué sirve la configuración del firewall. En el código siguiente se crea una configuración de firewall denominada Allow Windows Azure que habilita el acceso desde Windows Azure.

```
-- enable Windows Azure connections
exec sp_set_firewall_rule N'Allow Windows
Azure', '0.0.0.0', '0.0.0.0'
```

En el código siguiente se crea una configuración de firewall denominada Example setting 1 solo para la dirección IP 0.0.0.2. Después, se llama de nuevo al procedimiento almacenado `sp_set_firewall_rule` para permitir una dirección IP adicional, 0.0.0.3, en esa configuración de firewall.

```
-- create new firewall setting for only IP 0.0.0.2
exec sp_set_firewall_rule N'Example setting
1', '0.0.0.2', '0.0.0.2'

-- update firewall setting to also allow IP 0.0.0.3
exec sp_set_firewall_rule N'Example setting
1', '0.0.0.2', '0.0.0.3'
```

Para quitar una configuración del firewall, use el procedimiento almacenado `sp_delete_firewall_rule`, como se muestra en el código siguiente.

```
-- Remove example firewall setting
exec sp_delete_firewall_rule N'Example setting 1'
```

## SOLUCIONAR PROBLEMAS DE SQL AZURE

A continuación, veremos cómo solucionar problemas relacionados específicamente con Microsoft Base de datos de SQL Azure.

### Error de inicio de sesión

Base de datos de SQL Azure necesita el uso de la autenticación de SQL. Si no puede iniciar sesión correctamente, sus credenciales no serán válidas o la base de datos que solicitó no estará disponible.

Si necesita recuperar nombres de inicio de sesión o contraseñas, póngase en contacto con el administrador de servicios, que puede concederle acceso al servidor y la base de datos. Los administradores de servicios también pueden restablecer sus propias contraseñas mediante el Portal de administración de la plataforma Windows Azure.

## El servicio no está disponible o no existe

Si el servidor ya se ha aprovisionado y el servicio Base de datos de SQL Azure está disponible (puede comprobarlo mediante la página de estado de mantenimiento de Azure), puede que haya problemas de configuración en su instalación local. Por ejemplo, quizás no pueda resolver el nombre (lo que se puede probar mediante herramientas como **tracert**), puede que haya un firewall que bloquee el puerto 1433 empleado por SQL Azure o puede que esté usando un servidor proxy que no esté configurado correctamente. Use las mismas técnicas para solucionar estos problemas que emplearía para SQL Server.

Para ver el estado de todos los servicios de la plataforma Windows Azure, use la página de estado de mantenimiento de Azure. La página de estado contiene información actualizada sobre el estado de los servicios y datos históricos de los servicios. Para conectar con la página de estado y comprobar la disponibilidad del servicio Base de datos de SQL Azure en tiempo real, haga clic en el botón **Service Dashboard** de la página de soporte de Windows Azure o haga clic en Azure Health Status <http://go.microsoft.com/fwlink/?LinkId=168847> (Estado de mantenimiento de Azure) para ir directamente a la página de estado.

## Errores generales de red

Puede recibir errores generales de red porque Base de datos de SQL Azure puede desconectar usuarios en las circunstancias siguientes:

- Cuando una conexión está inactiva durante un período de tiempo prolongado.
- Cuando una conexión consume una cantidad excesiva de recursos o se bloquea en una transacción durante mucho tiempo.

## **Si el servidor está demasiado ocupado**

Si es posible, se envía un token de error de flujo TDS antes de desconectar a los usuarios. Para mejorar la experiencia de la aplicación, se recomienda implementar lógica de reintento en sus aplicaciones SQL Azure para detectar estos errores. Cuando se produzca un error, vuelva a establecer la conexión y, a continuación, ejecute de nuevo los comandos o la consulta.

Las duraciones permitidas máximas están sujetas a cambios dependiendo del uso de los recursos. Una sesión iniciada que haya estado inactiva durante 30 minutos se terminará automáticamente.

## **Rendimiento de la optimización de consultas**

Para mejorar el rendimiento en Base de datos de SQL Azure, use las mismas técnicas que emplearía con SQL Server.

## **Errores de SQL Server**

Base de datos de SQL Azure utiliza un subconjunto de los mensajes de error de SQL Server.

# DESARROLLO

## DESARROLLO EN BASE DE DATOS DE SQL AZURE

Ya sabemos que Microsoft Base de datos de SQL Azure es el servicio de base de datos relacional de la plataforma Windows Azure. Hay pocas diferencias entre desarrollar para Microsoft Base de datos de SQL Azure y para SQL Server. Al escribir aplicaciones para Base de datos de SQL Azure, puede utilizar uno de los lenguajes de programación de .NET Framework, que vienen con Visual Studio 2008 o posterior: Microsoft Visual Basic, Microsoft Visual C# o Microsoft Visual C++. También puede utilizar el proveedor de datos ADO.NET para SQL Server, el controlador ODBC de SQL Server 2008 Native Client y el controlador SQL Server 2008 para PHP versión 1.1. Además, Base de datos de SQL Azure le permite usar un subconjunto del lenguaje de consultas Transact-SQL para consultar una base de datos e insertar, actualizar y eliminar datos de tablas de una base de datos.

### Crear servidores SQL Azure

Para utilizar Base de datos de SQL Azure, debe crear primero una cuenta de la plataforma Windows Azure. La cuenta de Windows Azure se utiliza para configurar y administrar sus suscripciones, y para facturar el consumo de Windows Azure, Windows Azure AppFabric y Base de datos de SQL Azure. Una vez creada la cuenta de Windows Azure, puede usar el Portal de administración de la plataforma Windows Azure para agregar o quitar un servidor y una base de datos SQL Azure. También puede usar la API de administración de bases de datos SQL Azure para agregar o quitar mediante programación servidores SQL Azure y administrar reglas de firewall asociadas a un servidor.

Un servidor SQL Azure es un grupo lógico de bases de datos y actúa como punto administrativo central para varias bases de datos. Cuando se crea un servidor SQL Azure, se crea automáticamente una base de datos maestra (**master**) de solo lectura. La base de datos maestra (**master**) realiza un seguimiento de qué inicios de sesión tienen permiso para crear bases de datos u otros inicios de sesión. Debe estar conectado a la base de datos maestra (**master**) siempre que utilice CREATE, ALTER o DROP para crear, modificar o quitar inicios de sesión o bases de datos, respectivamente.

De forma predeterminada, el firewall de SQL Azure bloquea todo acceso a su servidor SQL Azure. Para empezar a utilizar su servidor SQL Azure, debe especificar una o más reglas de firewall que habiliten el acceso a su servidor SQL Azure. Las reglas de firewall se pueden administrar utilizando la interfaz de usuario del Portal de administración o utilizando mediante programación la API de administración de bases de datos. Antes de poder utilizar la API de administración de bases de datos, debe agregar un certificado para la autenticación, como se describe en Autenticar solicitudes de API de administración de bases de datos SQL Azure.

Después de haber creado una configuración del firewall, puede usar el inicio de sesión de entidad de seguridad a nivel de servidor y la base de datos maestra (**master**) para ver y editar la citada configuración. En la base de datos maestra (**master**), la configuración del firewall se conoce como *reglas*. La vista **sys.firewall\_rules** muestra la configuración del firewall actual, y los procedimientos almacenados **sp\_set\_firewall\_rule** y **sp\_delete\_firewall\_rule** le permiten cambiar la configuración del firewall. Para obtener más información, vea Firewall de SQL Azure.

Puede tener acceso a los detalles de facturación de sus cuentas de SQL Azure en el servidor SQL Azure mediante las vistas del sistema **sys.database\_usage** y **sys.bandwidth\_usage**.

## Crear bases de datos SQL Azure

Hay dos maneras de crear bases de datos SQL Azure:

- Utilizando el Portal de administración.
- Mediante la instrucción CREATE DATABASE.

Además, un ejemplo de código proporcionado en Conectar con SQL Azure mediante ADO.NET muestra cómo usar la instrucción CREATE DATABASE en código de aplicación cliente.

Para cambiar la edición y el nombre de la base de datos después de su creación, puede emplear la instrucción ALTER DATABASE.

## Compilar y hospedar aplicaciones SQL Azure

Hay dos maneras de compilar y hospedar aplicaciones Base de datos de SQL Azure:

- Hospedar el código de la aplicación de forma local en su propio centro de datos corporativo, pero hospedar la base de datos en SQL Azure. El código de la aplicación usa bibliotecas de cliente para tener acceso a las bases de datos en SQL Azure.
- Hospedar el código de la aplicación en Windows Azure y la base de datos en SQL Azure. Su aplicación puede usar las mismas bibliotecas de cliente para tener acceso a las bases de datos en SQL Azure. En este caso, su aplicación cliente puede ser una aplicación de escritorio o Silverlight que aproveche las ventajas de Entity Data Model y el cliente Servicios de datos de Microsoft WCF para tener acceso a los datos hospedados en SQL Azure.

Puede minimizar la latencia de red de las solicitudes a la base de datos SQL Azure si hospeda su aplicación en la plataforma Windows Azure. La implementación de la aplicación en Windows Azure proporciona transacciones más eficaces entre su aplicación y SQL Azure en comparación con una aplicación hospedada fuera de Windows Azure.

El ancho de banda empleado entre SQL Azure y Windows Azure o Windows Azure AppFabric es gratuito dentro de la misma subregión o el mismo centro de datos. Al implementar una aplicación Windows Azure, ponga la aplicación y la base de datos SQL Azure en la misma subregión para evitar los costos del ancho de banda.

## Desarrollar aplicaciones SQL Azure

El desarrollo de aplicaciones para Base de datos de SQL Azure es muy similar al desarrollo de aplicaciones para SQL Server. Puede elegir entre muchos tipos de aplicaciones y tecnologías al desarrollar una aplicación que tiene acceso a Base de

datos de SQL Azure. Base de datos de SQL Azure funciona con aplicaciones de terceros, PHP y muchas aplicaciones Microsoft como ADO.NET, Entity Framework, Servicios de datos de Microsoft WCF y ODBC.

Base de datos de SQL Azure proporciona un servicio de base de datos de varios inquilinos a gran escala en recursos compartidos. Para proporcionar una buena experiencia a todos los clientes de Base de datos de SQL Azure, su conexión al servicio se puede cerrar debido a las condiciones siguientes:

- Uso excesivo de recursos.
- Consultas de ejecución prolongada.
- Transacciones únicas de ejecución prolongada, entre las instrucciones BEGIN TRAN y END TRAN.
- Conexiones inactivas.

Esto es diferente del funcionamiento de una instancia local de SQL Server.

Para proporcionar una experiencia de usuario uniforme cuando se cierre una conexión, incorpore lógica de reintento a su aplicación para detectar una conexión cerrada e intentar completar la acción interrumpida.

Cuando la aplicación cliente se conecta a Base de datos de SQL Azure, se establece automáticamente CONTEXT\_INFO (Transact-SQL) con un valor GUID de sesión único. Recupere este valor GUID y úselo en su aplicación para hacer un seguimiento de los problemas de conectividad.

## **COMPATIBILIDAD CON TRANSACT-SQL (BASE DE DATOS DE SQL AZURE)**

Transact-SQL es un lenguaje que contiene comandos usados para administrar instancias de SQL Server, incluidas la creación y administración de todos los objetos de una instancia de SQL Server, y la inserción, recuperación, modificación y eliminación de todos los datos de tablas. Las aplicaciones pueden comunicarse con una instancia de SQL Server enviando instrucciones de Transact-SQL al servidor.

Transact-SQL es una extensión del lenguaje definido en los estándares de SQL publicados por la Organización para la estandarización (ISO) y el American National Standards Institute (ANSI).

Microsoft Base de datos de SQL Azure admite un subconjunto de Transact-SQL para SQL Server.

La compatibilidad con Transact-SQL en Base de datos de SQL Azure se puede describir en tres categorías principales:

- Elementos del lenguaje Transact-SQL que se admiten tal cual.
- Elementos del lenguaje Transact-SQL que no se admiten.
- Elementos del lenguaje Transact-SQL que proporcionan un subconjunto de los argumentos y opciones en sus elementos de Transact-SQL correspondientes en SQL Server 2008.

Las características del motor de base de datos de SQL Server marcadas como obsoletas no se admiten en Base de datos de SQL Azure. Las características desusadas no se deben utilizar en aplicaciones nuevas. Compruebe la lista de características marcadas como obsoletas en Referencia de Transact-SQL (motor de base de datos) en los libros en pantalla de SQL Server.

## **Características de Transact-SQL compatibles y parcialmente compatibles**

Base de datos de SQL Azure admite o admite parcialmente las siguientes características de Transact-SQL:

- Constantes.
- Restricciones.
- Cursores.
- Extensión de tipos de datos espaciales y métodos mediante CLR.
- Administración de índices y nueva generación de índices.
- Tablas temporales locales.
- Palabras clave reservadas.
- Datos e índices espaciales.
- Procedimientos almacenados.

- Administración de estadísticas.
- Transacciones.
- Desencadenadores.
- Tablas, combinaciones y variables de tabla.
- Elementos del lenguaje Transact-SQL como:
  - Crear/quitar bases de datos.
  - Crear/modificar/quitar tablas.
  - Crear/modificar/quitar usuarios e inicios de sesión.
- Funciones definidas por el usuario.
- Vistas.

## Características de Transact-SQL no admitidas

Base de datos de SQL Azure **no admite** las siguientes características de Transact-SQL:

- Common Language Runtime (CLR).
- Colocación de archivos de base de datos.
- Creación de reflejo de la base de datos.
- Consultas distribuidas.
- Transacciones distribuidas.
- Administración de grupos de archivos.
- Tablas temporales globales.
- Opciones de configuración de SQL Server.
- SQL Server Service Broker.
- Tablas del sistema.
- Marcas de seguimiento.

## COMPATIBILIDAD DE BASE DE DATOS SQL AZURE CON HERRAMIENTAS Y UTILIDADES DE SQL SERVER

Con Microsoft Base de datos de SQL Azure se pueden usar diversas herramientas y utilidades, que se describen en los títulos siguientes:

- La utilidad `sqlcmd`.
- SQL Server Management Studio.
- The Database Manager for SQL Azure.
- Aplicaciones de capa de datos.
- Asistente para generar y publicar scripts.
- SQL Server Integration Services.
- La utilidad `bcp`.
- SQL Server Analysis Services.
- SQL Server Reporting Services.
- SQL Azure Reporting.
- Objetos de administración de SQL Server (SMO).
- Asistente de migración de SQL Server (SSMA).
- Microsoft Access.
- PowerPivot for Excel.

### La utilidad `sqlcmd`

La utilidad `sqlcmd` le permite escribir instrucciones, procedimientos del sistema y archivos de script de Transact-SQL en el símbolo del sistema, en el Editor de consultas en modo `sqlcmd` o en un archivo de script de Windows: `sqlcmd` también se puede ejecutar en un paso de trabajo del sistema operativo (`cmd.exe`) de un trabajo del Agente SQL Server en una instancia local de SQL Server. La utilidad `sqlcmd` emplea OLE DB para ejecutar lotes de Transact-SQL.

Aunque Base de datos de SQL Azure admite la utilidad **sqlcmd**, no se admite una utilidad similar denominada **osql**.

## SQL Server Management Studio

SQL Server Management Studio de SQL Server 2008 R2 y SQL Server 2008 R2 Express se puede utilizar para configurar, administrar y tener acceso a Base de datos de SQL Azure. No se admiten las versiones anteriores de SQL Server Management Studio.

SQL Server Management Studio combina herramientas gráficas con una serie de editores de scripts para proporcionar acceso a SQL Server a los desarrolladores y administradores. Para obtener más información, vea Usar SQL Server Management Studio en los Libros en pantalla de SQL Server.

## El administrador de base de datos para SQL Azure

El administrador de base de datos para SQL Azure es una herramienta de administración basada en la web para las bases de datos SQL Azure existentes. Como usuario del portal, puede seleccionar una base de datos en el nuevo Portal de administración de la plataforma Windows Azure e iniciar al administrador de base de datos para administrar esa base de datos.

El administrador de base de datos admite tareas básicas de administración de base de datos como diseñar y editar tablas, vistas y procedimientos almacenados, y crear y ejecutar consultas de Transact-SQL. La herramienta de administración de base de datos está disponible cuando se selecciona una base de datos para administrar en el nuevo Portal de administración y hace clic en el botón **Manage** (Administrar) de la barra de herramientas.

## Aplicaciones de capa de datos

A partir de Microsoft SQL Server 2008 R2 y Microsoft Visual Studio 2010, se introdujeron las aplicaciones de capa de datos (DAC) para ayudar a los desarrolladores y administradores de bases de datos a empaquetar esquemas y objetos en una única entidad denominada paquete DAC. Para obtener más información, vea Explicar las aplicaciones de capa de datos en Libros en pantalla de SQL Server. Para crear un paquete DAC, puede utilizar el proyecto de aplicación de capa de datos de SQL Server en Microsoft Visual Studio 2010. Para extraer un DAC y generar un archivo de paquete DAC para una base de datos existente, puede emplear el asistente Extraer aplicación de capa de datos de SQL Server Management Studio.

Base de datos de SQL Azure admite la eliminación, implementación, extracción, registro y actualización en contexto de paquetes DAC. SQL Server 2008 R2 y Microsoft Visual Studio 2010 incluían el marco MDAC 1.0, que solo admitía actualizaciones en paralelo. Las actualizaciones en paralelo no admiten SQL Azure. Visual Studio 2010 SP1 incluye un nuevo marco MDAC 1.1 que se puede descargar del Centro de descarga de Microsoft. El marco MDAC 1.1 introduce actualizaciones en contexto que se admiten tanto en SQL Azure como en instancias de Motor de base de datos de SQL Server.

Puede usar el asistente Implementar aplicación de capa de datos de SQL Server Management Studio para implementar una aplicación de capa de datos (DAC) desde un paquete DAC en Base de datos de SQL Azure. El proceso de implementación registra una instancia de DAC, para lo cual almacena la definición y los metadatos de la DAC en la base de datos maestra (**master**) en SQL Azure, crea una base de datos y, a continuación, la rellena con todos los objetos de base de datos definidos en la DAC.

## Asistente para generar y publicar scripts

Puede utilizar el Asistente para generar y publicar scripts para transferir una base de datos de un equipo local a Base de datos de SQL Azure. El Asistente para generar y publicar scripts crea scripts de Transact-SQL para su base de datos local y los utiliza para publicar objetos de base de datos en Base de datos de SQL Azure.

Hay un nuevo parámetro de tipo de motor para Base de datos de SQL Azure. Cuando incluya en el script una base de datos y el destino sea Base de datos de SQL Azure, establezca el parámetro de motor en SQL Azure para que el Asistente para generar y publicar scripts cree código de Transact-SQL que sea compatible con Base de datos de SQL Azure.

## SQL Server Integration Services

Puede transferir datos a Base de datos de SQL Azure mediante SQL Server 2008 Integration Services (SSIS).

SQL Server 2008 R2 o posterior admite la copia masiva para la transferencia de datos entre una instancia de Microsoft SQL Server y Base de datos de SQL Azure.

## La utilidad bcp

Puede transferir datos a Base de datos de SQL Azure mediante la utilidad de copia masiva (BCP.exe). La utilidad **bcp** realiza copias masivas de datos entre una instancia de Microsoft SQL Server y un archivo de datos en un formato especificado por el usuario. La utilidad **bcp** se puede usar para importar un número elevado de filas nuevas en tablas de SQL Server o para exportar datos de tablas a archivos de datos.

A partir de SQL Server 2008 R2, Base de datos de SQL Azure admite el argumento *queryout* cuando se emplea la utilidad bcp. Cuando use el argumento *queryout*, asegúrese de anexar *-d database\_name argument*. De lo contrario, la utilidad bcp no podrá conectarse a la base de datos SQL Azure.

## SQL Server Analysis Services y SQL Server Reporting Services

A partir de SQL Server 2008 R2, se admite el uso de SQL Azure como origen de datos al ejecutar SQL Server Analysis Services de forma local.

A partir de SQL Server 2008 R2, se admite el uso de SQL Azure como origen de datos al ejecutar SQL Server Reporting Services de forma local.

## SQL Azure Reporting

También está disponible Customer Technology Preview de SQL Azure Reporting. Microsoft SQL Azure Reporting es un servicio de informes basado en la nube creado con tecnologías Base de datos de SQL Azure, SQL Server y SQL Server Reporting Services. Puede publicar, ver y administrar informes que muestran datos de orígenes de datos SQL Azure.

## Objetos de administración de SQL Server (SMO)

Base de datos de SQL Azure habilita un conjunto parcial de Objetos de administración de SQL Server (SMO). El conjunto parcial de SMO solo está habilitado para proporcionar acceso a Management Studio a SQL Azure. Estos objetos ofrecen una funcionalidad limitada y no están pensados para su uso en aplicaciones. Están habilitados los objetos de los espacios de nombres **Microsoft.SqlServer.Management.SqlParser.Intellisense** y **Microsoft.SqlServer.Management.Smo** que se indican a continuación:

Check	Index	Synonym
Column	IndexedColumn	Table
Database	Login	Trigger
DatabaseDdlTrigger	Schema	User
DatabaseRole	SystemDataType	UserDefinedDataType
DefaultConstraint	Server	UserDefinedFunction
ForeignKey	Statistic	UserDefinedTableType
ForeignKeyColumn	StatisticColumn	View
Parameter	StoredProcedure	

Si utiliza un objeto que no se admite, SMO producirá una excepción.

## Asistente de migración de SQL Server (SSMA)

A partir del Asistente de migración de SQL Server 2008 para Access versión 4.2, SSMA permite migrar el esquema y los datos de Microsoft Access a Base de datos de SQL Azure y agrega compatibilidad con bases de datos de Access 2010. Además, el Asistente de migración de Microsoft SQL Server 2008 para MySQL versión 1.0 admite migrar el esquema y los datos de MySQL a Base de datos de SQL Azure.

## Microsoft Access

Microsoft Access 2010 admite conexiones a Base de datos de SQL Azure a través de ODBC cuando se usa junto con los controladores ODBC de SQL Server más recientes.

## PowerPivot para Excel

PowerPivot para Excel es un complemento que puede emplear para realizar análisis de datos eficaces en Excel 2010, lo que aporta Business Intelligence de autoservicio a su escritorio. PowerPivot para Excel incluye una ventana para agregar y preparar datos y una pestaña PowerPivot en la cinta de Excel que puede usar para manipular los datos en una hoja de cálculo de Excel.

PowerPivot para Excel también incluye un asistente que puede emplear para importar datos de distintos orígenes, como bases de datos SQL Azure, grandes bases de datos corporativas de la intranet, fuentes de distribución de datos públicas, hojas de cálculo y archivos de texto de un equipo.

## **INSTRUCCIONES Y LIMITACIONES GENERALES (BASE DE DATOS SQL AZURE)**

A continuación, se describen las instrucciones y limitaciones generales de Microsoft Base de datos de SQL Azure relativas a los aspectos siguientes:

- Compatibilidad con controladores, bibliotecas y protocolos.
- Compatibilidad con Visual Studio.
- Compatibilidad con herramientas y tecnologías.
- Compatibilidad con la migración de datos.
- Trabajos y Agente SQL Server.
- Compatibilidad con transacciones.
- Niveles de aislamiento basado en versiones de fila.
- Compatibilidad con la intercalación de SQL Server.
- Requisito de índices clúster.
- Número de bases de datos y límite de tamaño.
- Requisitos de nomenclatura.
- Restricciones de conexión.

### **Compatibilidad con controladores, bibliotecas y protocolos**

Al escribir aplicaciones para Base de datos de SQL Azure, puede utilizar los siguientes controladores y bibliotecas:

- Proveedor de datos .NET Framework para SQL Server (System.Data.SqlClient) de .NET Framework 3.5 Service Pack 1 o posterior.
- Entity Framework de .NET Framework 3.5 Service Pack 1 o posterior.
- Controlador ODBC de SQL Server 2008 R2 Native Client. También se admite el controlador ODBC de SQL Server 2008 Native Client, pero tiene menos funcionalidad.

- Controlador de SQL Server 2008 para PHP versión 1.1 o posterior.
- Una versión actualizada de Controlador JDBC de SQL Server 3.0 que admite la base de datos SQL Azure.

Base de datos de SQL Azure admite el cliente del protocolo de flujo TDS versión 7.3 o posterior. Las versiones anteriores del protocolo TDS no son compatibles.

No se admite la conexión a Base de datos de SQL Azure mediante OLE DB.

## Compatibilidad con Visual Studio

Al escribir aplicaciones para Base de datos de SQL Azure, puede usar uno de los lenguajes de programación de .NET Framework incluidos con Visual Studio 2008 o posterior: Microsoft Visual Basic, Microsoft Visual C# o Microsoft Visual C++. Visual Studio proporciona una consola de administración de servidores, el **Explorador de servidores**, para abrir conexiones de datos e iniciar sesión en servidores y explorar bases de datos. A partir de Visual Studio 2010, puede emplear el **Explorador de servidores** para conectarse a sus bases de datos en SQL Azure y explorarlas. No se admiten las versiones anteriores del **Explorador de servidores**. Visual Studio 2010 es totalmente compatible.

## Compatibilidad con herramientas y tecnologías

Tenga en cuenta lo siguiente cuando utilice herramientas para conectarse a Base de datos de SQL Azure:

- Solo se permiten conexiones TCP/IP.
- Se admiten conjuntos de resultados activos múltiples (MARS).
- Puesto que algunas herramientas implementan el flujo TDS de manera diferente, quizás tenga que anexar el nombre del servidor SQL Azure a la parte de inicio de sesión de la cadena de conexión usando la notación <login>@<server>. Para obtener más información, vea Administrar bases de datos e inicios de sesión en SQL Azure.
- No se admite el explorador SQL Server 2008 SQL Server porque Base de datos de SQL Azure no tiene puertos dinámicos, solo el puerto 1433.

## Compatibilidad con ODBC

Puede usar **Configurar orígenes de datos ODBC** para definir orígenes de datos de usuario y del sistema para Base de datos de SQL Azure. Para ver la lista de los orígenes de datos de usuario y del sistema, compruebe las pestañas **User DSN** o **System DSN** del cuadro de diálogo Administrador de orígenes de datos ODBC.

Cuando utilice el asistente **Nombre del origen de datos (DSN)** para definir un origen de datos para Base de datos de SQL Azure, haga clic en la opción **Con la autenticación de SQL Server, mediante un Id. de inicio de sesión y una contraseña escritos por el usuario** y seleccione **Conectar con SQL Server para obtener la configuración predeterminada de las opciones de configuración adicionales**. Escriba su nombre de usuario y su contraseña para conectarse a su servidor SQL Azure como **Login ID** y **Password**. Cuando use SQL Server 2008 Native Client, desactive la casilla **Conectar con SQL Server para obtener la configuración predeterminada de las opciones de configuración adicionales**. A partir del controlador ODBC de SQL Server 2008 R2 Native Client, no es necesario desactivar la casilla **Conectar con SQL Server para obtener la configuración predeterminada de las opciones de configuración adicionales**. Haga clic en **Establecer la siguiente base de datos como predeterminada** y escriba el nombre de su base de datos SQL Azure aunque no aparezca en la lista. Observe que el asistente muestra varios idiomas en la lista **Establecer el siguiente idioma para los mensajes del sistema de SQL Server**.

En esta versión, Base de datos de SQL Azure solo admite inglés, así que seleccione **Inglés** como idioma. Base de datos de SQL Azure no admite **Servidor reflejado** ni **Adjuntar base de datos**, por lo que debe dejar esos elementos vacíos. Haga clic en **Probar conexión**.

- Cuando se usa el controlador ODBC de SQL Server 2008 Native Client, el botón **Probar conexión** puede producir un error porque no se admite **master.dbo.syscharsets**. Omita este error, guarde el DSN y utilícelo. Además, cuando decida conectarse a una base de datos distinta de la base de datos maestra (master) mientras configura el DSN, puede que no aparezca el mensaje **PRUEBAS COMPLETADAS CORRECTAMENTE** aunque no haya ningún error.
- Con el controlador ODBC de SQL Server 2008 R2 Native Client, no recibirá el error que indica que no se admite **master.dbo.syscharsets** aunque se conecte a una base de datos diferente.

## Compatibilidad con la migración de datos

Puede transferir datos a Base de datos de SQL Azure usando lo siguiente:

- SQL Server 2008 Integration Services (SSIS).
- La utilidad de copia masiva (BCP.exe).
- La clase `System.Data.SqlClient.SqlBulkCopy`.
- Scripts que usan instrucciones INSERT para cargar datos en la base de datos.

Base de datos de SQL Azure **no** admite:

- La instrucción RESTORE.
- Adjuntar una base de datos al servidor SQL Azure.

## Trabajos y Agente SQL Server

Base de datos de SQL Azure no admite Agente SQL Server ni trabajos de SQL Server. Sin embargo, puede ejecutar Agente SQL Server en su SQL Server local y conectarse a Base de datos de SQL Azure.

## Compatibilidad con transacciones

Base de datos de SQL Azure no admite transacciones distribuidas, que son transacciones que afectan a varios recursos.

A partir de la versión 2.0 de .NET Framework, las transacciones de la aplicación se pueden promover automáticamente a transacciones distribuidas. Esto se aplica a las aplicaciones que utilizan la clase `System.Data.SqlClient` para realizar operaciones de base de datos en el contexto de una transacción `System.Transactions`.

La promoción de la transacción se produce al abrir varias conexiones a servidores o bases de datos diferentes dentro de un `TransactionScope`, o al inscribir varias conexiones en un objeto `System.Transactions` mediante el método `EnlistTransaction`. La promoción de la transacción también se produce al abrir varias conexiones simultáneas al mismo servidor y a la misma base de datos dentro del mismo `TransactionScope` o mediante el método `EnlistTransaction`.

A partir de la versión 3.5 de .NET Framework, no se promoverá la transacción si las cadenas de conexión para las conexiones simultáneas son exactamente las mismas.

Es posible que Base de datos de SQL Azure no conserve las marcas de tiempo no confirmadas de la base de datos actual (DBTS) en las conmutaciones por error.

## Niveles de aislamiento basado en versiones de fila

Las opciones de base de datos `READ_COMMITTED_SNAPSHOT` y `ALLOW_SNAPSHOT_ISOLATION` están establecidas en `ON` en Base de datos de SQL Azure. Como no se admite `SET <snapshot_option>` en la instrucción `ALTER DATABASE` de Transact-SQL, no se pueden cambiar estas opciones de base de datos. Para obtener más información sobre los niveles de aislamiento basado en versiones de fila, vea Descripción de los niveles de aislamiento basado en el control de versiones de filas.

## Compatibilidad con la intercalación de SQL Server

La intercalación de base de datos predeterminada utilizada por Base de datos de SQL Azure es `SQL_LATIN1_GENERAL_CP1_CI_AS`, donde `LATIN1_GENERAL` es Inglés (Estados Unidos), `CP1` es página de códigos 1252, `CI` es sin distinción entre mayúsculas y minúsculas, y `AS` es con distinción de acentos.

Cuando utiliza SQL Server local, puede establecer las intercalaciones en los niveles de servidor, base de datos, columna y expresión. Base de datos de SQL Azure no permite establecer la intercalación en el nivel de servidor o de base de datos. Para utilizar la intercalación no predeterminada con Base de datos de SQL Azure, establezca la intercalación en el nivel de columna o de expresión.

## Requisito de índices clúster

Base de datos de SQL Azure no admite tablas sin índices clúster. Una tabla debe tener un índice clúster. Si se crea una tabla sin una restricción clúster, se debe crear un índice clúster antes de que se permita una operación de inserción en la tabla.

Por ejemplo, el siguiente código de Transact-SQL crea una nueva tabla, crea un índice clúster en la tabla y, a continuación, inserta datos en la tabla. Realizar la inserción sin crear el índice clúster producirá un error.

```
CREATE TABLE Table1 (Col1 int, Col2 char(20))
CREATE CLUSTERED INDEX Table1_Index ON Table1 (Col1)
INSERT INTO Table1 VALUES (1, 'string1')
```

## Número de bases de datos y límites de tamaño

De forma predeterminada, Base de datos de SQL Azure admite hasta 150 bases de datos en cada servidor SQL Azure, incluida la base de datos **master**. Puede crear hasta 149 bases de datos en cada servidor SQL Azure. Puede que haya disponible una extensión de este límite para su servidor SQL Azure.

Base de datos de SQL Azure proporciona dos ediciones de base de datos: Web Edition y Business Edition. Las bases de datos Web Edition pueden crecer hasta un tamaño de 5 GB y las bases de datos Business Edition puede crecer hasta un tamaño de 50 GB. Se especifica MAXSIZE la primera vez que se crea la base de datos y se puede cambiar después mediante ALTER DATABASE. MAXSIZE ofrece la posibilidad de limitar el tamaño de la base de datos. Si el tamaño de la base de datos alcanza el valor de MAXSIZE, recibirá un código de error 40544. Cuando esto ocurre, no puede insertar o actualizar datos ni crear objetos nuevos como tablas, procedimientos almacenados, vistas y funciones. Sin embargo, todavía puede leer y eliminar datos, truncar tablas, quitar tablas e índices y volver a generar índices. Si quita algunos datos para liberar espacio de almacenamiento, puede haber un retraso de hasta quince minutos antes de que pueda insertar nuevos datos.

## Requisitos de nomenclatura

Algunos nombres de usuario no se permiten por razones de seguridad. No puede utilizar los nombres siguientes:

- admin
- administrator
- guest
- root
- sa

Los nombres de todos los objetos nuevos deben cumplir las reglas de SQL Server para los identificadores.

Además, los nombres de inicio de sesión y de usuario no pueden contener el carácter \ (no se admite la autenticación de Windows).

## Restricciones de conexión

Base de datos de SQL Azure proporciona un servicio de base de datos de varios inquilinos a gran escala en recursos compartidos. Para proporcionar una buena experiencia a todos los clientes de Base de datos de SQL Azure, su conexión al servicio se puede cerrar debido a las condiciones siguientes:

- Uso excesivo de recursos.
- Conexiones que han estado inactivas durante 30 minutos o más.
- Conmutación por error debido a errores del servidor.

Las duraciones permitidas máximas están sujetas a cambios dependiendo del uso de los recursos. Una sesión iniciada que haya estado inactiva durante 30 minutos se terminará automáticamente. Se recomienda encarecidamente que utilice la agrupación de conexiones y cierre siempre la conexión cuando termine de usarla para que la conexión no usada se devuelva al grupo.

Cuando se cierre su conexión al servicio, recibirá uno de los errores siguientes: 40197, 40501, 40544, 40549, 40550, 40551, 40552, 40553 y 40613.

Se recomienda que implemente lógica de reintento en sus aplicaciones SQL Azure para detectar estos errores. Cuando se produzca un error, vuelva a establecer la conexión y, a continuación, ejecute de nuevo los comandos o la consulta.

## INSTRUCCIONES Y LIMITACIONES DE SEGURIDAD EN BASE DE DATOS DE SQL AZURE

A continuación, se describen las instrucciones y limitaciones de Microsoft Base de datos de SQL Azure relacionadas con la seguridad. Tenga en cuenta los siguientes aspectos a la hora de administrar la seguridad de sus bases de datos SQL Azure:

- Firewall.
- Cifrado y validación de certificados.
- Autenticación.
- Inicios de sesión y usuarios.
- Procedimientos recomendados.

## Firewall

El servicio Base de datos de SQL Azure solo está disponible a través del puerto TCP 1433. Para tener acceso a una base de datos SQL Azure desde su equipo, asegúrese de que el firewall permite la comunicación de TCP de salida en el puerto TCP 1433.

Antes de poder conectarse por primera vez al servidor SQL Azure, debe utilizar el Portal de administración de la plataforma Windows Azure para configurar el firewall de SQL Azure. Necesitará crear una configuración de firewall que permita los intentos de conexión desde su equipo o desde Windows Azure.

## Cifrado y validación de certificados

Todas las comunicaciones entre Base de datos de SQL Azure y su aplicación necesitan cifrado (SSL) en todo momento. Si su aplicación cliente no valida los certificados en la conexión, su conexión a Base de datos de SQL Azure es susceptible de sufrir ataques de tipo "Man in the middle".

Para validar los certificados con código de aplicación o herramientas, solicite explícitamente una conexión cifrada y no confíe en los certificados de servidor. Si el código de aplicación o las herramientas no solicitan una conexión cifrada, seguirán recibiendo conexiones cifradas. Sin embargo, no pueden validar los certificados de servidor y, por tanto, son susceptibles de sufrir ataques de tipo "Man in the middle".

Para validar los certificados con código de aplicación ADO.NET, establezca `Encrypt=True` y `TrustServerCertificate=False` en la cadena de conexión de base de datos.

SQL Server Management Studio también admite la validación de certificados. En el cuadro de diálogo **Conectar con el servidor**, haga clic en **Cifrar conexión** en la pestaña **Propiedades de conexión**.

SQL Server Management Studio no admite Base de datos de SQL Azure en versiones anteriores a SQL Server 2008 R2. Aunque SQLCMD admitía Base de datos de SQL Azure a partir de SQL Server 2008, no admite la validación de certificados en versiones anteriores a SQL Server 2008 R2. Para validar certificados con SQLCMD a partir de SQL Server 2008 R2, utilice la opción `-N` de la línea de comandos y no emplee la opción `-C`. Cuando se emplea la opción `-N`, SQLCMD solicita una conexión cifrada. Si no se usa la opción `-C`, SQLCMD no confía implícitamente en el certificado del servidor y se ve obligado a validarlo.

## Autenticación

Base de datos de SQL Azure solo admite la autenticación de SQL Server. No se admite la autenticación de Windows (seguridad integrada). Los usuarios deben proporcionar sus credenciales (inicio de sesión y contraseña) cada vez que se conectan a Base de datos de SQL Azure.

Por razones de rendimiento, cuando una contraseña se restablece en Base de datos de SQL Azure, la conexión no se volverá a autenticar inmediatamente, incluso aunque la conexión se restablezca debido a la agrupación de conexiones. Esto es diferente del comportamiento de SQL Server local. En su lugar, Base de datos de SQL Azure se basa en un mecanismo de reautenticación para desconectar las sesiones obsoletas. Cuando se envía una nueva solicitud en cualquier conexión, si han transcurrido más de 60 minutos desde la última reautenticación, se realizará una reautenticación. Si se ha cambiado la contraseña, se producirá un error en la solicitud y la sesión se desconectará (terminará).

## Inicios de sesión y usuarios

Hay ciertas restricciones al administrar inicios de sesión y usuarios en Base de datos de SQL Azure.

- Para el inicio de sesión de entidad de seguridad a nivel de servidor, se aplican las siguientes restricciones:
  - No se puede modificar o quitar el usuario de la base de datos **master** correspondiente al inicio de sesión de entidad de seguridad a nivel de servidor.
  - Aunque el inicio de sesión de entidad de seguridad a nivel de servidor no es miembro de los dos roles de servidor **dbmanager** y **loginmanager**, tiene todos los permisos concedidos con estos dos roles. Este inicio de sesión se crea durante el aprovisionamiento del servidor y es similar al inicio de sesión **sa** en una instancia de SQL Server

Para todos los inicios de sesión, se aplican las restricciones siguientes:

- El idioma predeterminado es el inglés (Estados Unidos).

- Para tener acceso a la base de datos **master**, cada inicio de sesión debe estar asignado a una cuenta de usuario en la base de datos **master**.
- Si no especifica una base de datos en la cadena de conexión, se conectará a la base de datos **master** de manera predeterminada.
- Debe estar conectado a la base de datos **master** al ejecutar las instrucciones CREATE/ALTER/DROP LOGIN y CREATE/ALTER/DROP DATABASE.
- Al ejecutar las instrucciones CREATE/ALTER/DROP LOGIN y CREATE/ALTER/DROP DATABASE en una aplicación ADO.NET, no se permite utilizar comandos parametrizados. Para obtener más información, vea Comandos y parámetros (ADO.NET).
- Al ejecutar las instrucciones CREATE/ALTER/DROP DATABASE y CREATE/ALTER/DROP LOGIN, cada una de ellas debe ser la única instrucción en un lote Transact-SQL. De lo contrario, se producirá un error. Por ejemplo, el siguiente código de Transact-SQL comprueba si la base de datos existe. Si existe, se llama a una instrucción DROP DATABASE para quitarla. Puesto que la instrucción DROP DATABASE no es la única del lote, la ejecución de este código Transact-SQL producirá un error.

```
IF EXISTS (SELECT [name]
           FROM [sys].[databases]
           WHERE [name] = N'database_name')
  DROP DATABASE [database_name];
Go
```

- Cuando se ejecuta la instrucción CREATE USER con la opción FOR/FROM LOGIN, debe ser la única instrucción en un lote Transact-SQL.
- Cuando se ejecuta la instrucción ALTER USER con la opción WITH LOGIN, debe ser la única instrucción de un lote Transact-SQL.
- Solo el inicio de sesión de entidad de seguridad a nivel de servidor y los miembros del rol dbmanager tienen permiso para ejecutar las instrucciones CREATE DATABASE y DROP DATABASE.
- Solo el inicio de sesión de entidad de seguridad a nivel de servidor y los miembros del rol loginmanager tienen permiso para ejecutar las instrucciones CREATE LOGIN, ALTER LOGIN y DROP LOGIN.

- Cuando el propietario de un rol de base de datos intenta agregar o quitar a otro usuario de la base de datos a o de ese rol de base de datos, se puede producir el siguiente error: User or role 'Name' does not exist in this database. Este error se produce porque el usuario no es visible para el propietario. Para resolver este problema, conceda al propietario el permiso VIEW DEFINITION en el usuario.

## Procedimientos recomendados

Tenga en cuenta lo siguiente para que sus aplicaciones de base de datos SQL Azure sean menos vulnerables a las amenazas de seguridad:

- Utilice siempre las actualizaciones más recientes: cuando se conecte a su base de datos SQL Azure, use siempre la versión más actualizada de las herramientas y las bibliotecas para evitar vulnerabilidades de seguridad.
- Bloquee las conexiones entrantes en el puerto TCP 1433: solo las conexiones de salida en el puerto TCP 1433 son necesarias para que las aplicaciones se comuniquen con Base de datos de SQL Azure. Si cualquier otra aplicación no necesita comunicaciones entrantes en ese equipo, asegúrese de que el firewall siga bloqueando las conexiones entrantes en el puerto TCP 1433.
- Evite las vulnerabilidades por inyección de código: para asegurarse de que sus aplicaciones no tienen vulnerabilidades por inyección de código SQL, utilice consultas parametrizadas siempre que sea posible. Asegúrese también de revisar el código detenidamente y ejecute una prueba de penetración antes de implementar la aplicación.

## LIMITACIONES DE LAS CARACTERÍSTICAS DE SQL SERVER EN BASE DE DATOS SQL AZURE

Microsoft Base de datos de SQL Azure es muy similar a SQL Server en lo que se refiere a la compatibilidad con Transact-SQL y el protocolo de flujo TDS que emplea para comunicarse con su cliente de base de datos. En cuanto a la compatibilidad con características, hay algunas diferencias entre Base de datos de SQL Azure y el producto SQL Server. En este apartado se enumeran algunas características de SQL Server 2008, SQL Server 2005 y versiones anteriores de SQL Server que no son compatibles con Base de datos de SQL Azure.

## Características de SQL Server 2008 R2 no admitidas por Base de datos SQL Azure

Base de datos de SQL Azure no admite las siguientes características que eran nuevas en SQL Server 2008 R2:

- Utilidad de SQL Server.
- Proveedor de SQL Server PowerShell. Sin embargo, se pueden ejecutar scripts de PowerShell en un equipo local y conectar con Base de datos de SQL Azure usando objetos compatibles (como Objetos de administración de SQL o Marco de trabajo de la aplicación de capa de datos).
- Master Data Services.

## Características de SQL Server 2008 no admitidas por Base de datos SQL Azure

Base de datos de SQL Azure no admite las siguientes características que eran nuevas en SQL Server 2008:

- Captura de datos modificados.
- Auditoría de datos.
- Compresión de datos.
- Eventos extendidos.
- Extensión de tipos y métodos espaciales mediante Common Language Runtime (CLR).
- Administración de claves externas / Administración extensible de claves.
- Datos FILESTREAM.
- Búsqueda de texto completo integrada.
- Agregados definidos por el usuario (UDA) grandes.
- Tipos definidos por el usuario (UDT) grandes.
- Recopilación de datos de rendimiento (Recopilador de datos).

- Administración basada en directivas.
- Regulador de recursos.
- Columnas dispersas.
- Replicación de SQL Server.
- Cifrado de datos transparente.

## **Características de SQL Server 2005 no admitidas por Base de datos SQL Azure**

Base de datos de SQL Azure no admite las siguientes características que eran nuevas en SQL Server 2005:

- Common Language Runtime (CLR) y tipos de datos definidos por el usuario de CLR.
- Creación de reflejo de la base de datos.
- Service Broker.
- Particionamiento de tablas.
- No se admite la indización de XML ni de XML con tipo. Base de datos de SQL Azure admite el tipo de datos XML.

## **Otras características de SQL Server R2 no admitidas por Base de datos SQL Azure**

Base de datos de SQL Azure no admite las siguientes características de versiones anteriores de SQL Server:

- Copias de seguridad y restauración.
- Replicación.
- Procedimientos almacenados extendidos.
- Trabajos y Agente SQL Server.

## INSTRUCCIONES PARA CONECTAR CON BASE DE DATOS SQL AZURE

Microsoft Base de datos de SQL Azure funciona con aplicaciones de terceros, código abierto y muchas aplicaciones conocidas de Microsoft, como Servicios de datos de Microsoft WCF, ODBC y ADO.NET.

A continuación, se muestra una lista de consideraciones generales aplicables a cualquier conexión con Base de datos de SQL Azure:

- El servicio Base de datos de SQL Azure solo está disponible con el puerto TCP 1433. Para tener acceso a una base de datos SQL Azure desde su equipo, asegúrese de que el firewall permite la comunicación de TCP de salida en el puerto TCP 1433.
- Antes de poder conectarse por primera vez al servidor SQL Azure, debe utilizar el Portal de administración de la plataforma Windows Azure para configurar el firewall de SQL Azure. Necesitará crear una configuración de firewall que permita los intentos de conexión desde su equipo o desde Windows Azure.
- Puesto que algunas herramientas implementan el flujo TDS de manera diferente, quizás tenga que anexar el nombre del servidor SQL Azure al inicio de sesión en la cadena de conexión usando la notación <login>@<server>. En estos casos, separe el inicio de sesión y el nombre del servidor SQL Azure con el símbolo @. Por ejemplo, si su inicio de sesión se denomina login1 y el nombre completo de su servidor SQL Azure es servername.database.windows.net, el parámetro de nombre de usuario de la cadena de conexión debe ser: login1@servername. Esta restricción impone algunas limitaciones sobre el texto que puede elegir para el nombre de inicio de sesión.
- Si no especifica una base de datos en la cadena de conexión, se conectará a la base de datos maestra (**master**).
- Actualmente no se admite el comando USE de Transact-SQL para cambiar de base de datos. Establezca una conexión directamente con la base de datos de destino.
- Base de datos de SQL Azure no admite todas las instrucciones de Transact-SQL incrustadas. Algunas instrucciones que se admiten en Base de datos de SQL Azure pueden no admitir todos los parámetros opcionales de SQL Server 2008.

- Debe conectar con la base de datos maestra (**master**) para crear inicios de sesión y bases de datos. La base de datos maestra (**master**) también tiene las vistas `sys.sql_logins` y `sys.databases`, que puede utilizar para ver inicios de sesión y bases de datos, respectivamente.
- Base de datos de SQL Azure no admite la autenticación de Windows. Use la autenticación de SQL Server en su cadena de conexión.
- No se admite la conexión a Base de datos de SQL Azure mediante OLE DB.
- Base de datos de SQL Azure no admite transacciones distribuidas, que son transacciones que afectan a varios recursos.
- Base de datos de SQL Azure proporciona un servicio de base de datos de varios inquilinos a gran escala en recursos compartidos. Para proporcionar una buena experiencia a todos los clientes de Base de datos de SQL Azure, su conexión al servicio se puede cerrar.

# CONECTAR CON SQL AZURE

## INSTRUCCIONES PARA CONECTAR CON BASE DE DATOS SQL AZURE

Microsoft Base de datos de SQL Azure funciona con aplicaciones de terceros, código abierto y muchas aplicaciones conocidas de Microsoft, como Servicios de datos de Microsoft WCF, ODBC y ADO.NET.

A continuación, se muestra una lista de consideraciones generales aplicables a cualquier conexión con Base de datos de SQL Azure:

- El servicio Base de datos de SQL Azure solo está disponible con el puerto TCP 1433. Para tener acceso a una base de datos SQL Azure desde su equipo, asegúrese de que el firewall permite la comunicación de TCP de salida en el puerto TCP 1433.
- Antes de poder conectarse por primera vez al servidor SQL Azure, debe utilizar el Portal de administración de la plataforma Windows Azure para configurar el firewall de SQL Azure. Necesitará crear una configuración de firewall que permita los intentos de conexión desde su equipo o desde Windows Azure.
- Puesto que algunas herramientas implementan el flujo TDS de manera diferente, quizás tenga que anexar el nombre del servidor SQL Azure al inicio de

sesión en la cadena de conexión usando la notación <login>@<server>. En estos casos, separe el inicio de sesión y el nombre del servidor SQL Azure con el símbolo @. Por ejemplo, si su inicio de sesión se denomina login1 y el nombre completo de su servidor SQL Azure es servername.database.windows.net, el parámetro de nombre de usuario de la cadena de conexión deberá ser: login1@servername. Esta restricción impone algunas limitaciones sobre el texto que puede elegir para el nombre de inicio de sesión.

- Si no especifica una base de datos en la cadena de conexión, se conectará a la base de datos maestra (**master**).
- Actualmente no se admite el comando USE de Transact-SQL para cambiar de base de datos. Establezca una conexión directamente con la base de datos de destino.
- Base de datos de SQL Azure no admite todas las instrucciones de Transact-SQL incrustadas. Algunas instrucciones que se admiten en Base de datos de SQL Azure pueden no admitir todos los parámetros opcionales de SQL Server 2008.
- Debe conectar con la base de datos maestra (**master**) para crear inicios de sesión y bases de datos. La base de datos maestra (**master**) también tiene las vistas sys.sql\_logins y sys.databases, que puede utilizar para ver inicios de sesión y bases de datos, respectivamente.
- Base de datos de SQL Azure no admite la autenticación de Windows. Use la autenticación de SQL Server en su cadena de conexión.
- No se admite la conexión a Base de datos de SQL Azure mediante OLE DB.
- Base de datos de SQL Azure no admite transacciones distribuidas, que son transacciones que afectan a varios recursos.
- Base de datos de SQL Azure proporciona un servicio de base de datos de varios inquilinos a gran escala en recursos compartidos. Para proporcionar una buena experiencia a todos los clientes de Base de datos de SQL Azure, su conexión al servicio se puede cerrar.

## CONECTAR CON SQL AZURE MEDIANTE SQLCMD

No se puede conectar directamente con Microsoft Base de datos de SQL Azure mediante la utilidad del símbolo del sistema **sqlcmd** incluida con SQL Server. La utilidad **sqlcmd** permite especificar instrucciones, procedimientos del sistema y archivos de script de Transact-SQL en el símbolo del sistema.

Para conectar con Base de datos de SQL Azure mediante **sqlcmd**, anexe el nombre del servidor SQL Azure al inicio de sesión en la cadena de conexión usando la notación *<inicioDeSesión>@<servidor>*. Por ejemplo, si su inicio de sesión es *inicioDeSesión1* y el nombre completo del servidor SQL Azure es *nombreDeServidor.baseDeDatos.windows.net*, el parámetro *username* de la cadena de conexión será: *inicioDeSesión1@nombreDeServidor*. Esta restricción impone algunas limitaciones sobre el texto que puede elegir para el nombre de inicio de sesión.

Base de datos de SQL Azure no admite las opciones *-z* y *-Z* usadas para cambiar la contraseña de los usuarios con SQLCMD. Para cambiar las contraseñas de inicio de sesión, puede usar ALTER LOGIN (Base de datos de SQL Azure) después de conectar con la base de datos master.

En el ejemplo siguiente se muestra cómo conectar con la base de datos master en un servidor SQL Azure y, a continuación, crear una base de datos de ejemplo usando la instrucción CREATE DATABASE (Base de datos SQL Azure) de Transact-SQL:

```
C:\>sqlcmd -U <ProvideLogin@Server> -P <ProvidePassword> -S
<ProvideServerName> -d master
1> CREATE DATABASE <ProvideDatabaseName>;
2> GO
3> QUIT
```

Después, puede conectar directamente con la base de datos creada previamente y crear una nueva tabla en la base de datos usando la instrucción CREATE TABLE (Base de datos SQL Azure) de Transact-SQL:

```
C:\>sqlcmd -U <ProvideLogin@Server> -P <ProvidePassword> -S
<ProvideServerName> -d <ProvideDatabaseName>
1> CREATE TABLE table1 (Col1 int primary key, Col2 varchar(20));
2> GO
3> QUIT
```

Base de datos de SQL Azure no admite tablas de montón. Debe crear una clave principal o un índice clúster.

## CONECTAR CON SQL AZURE MEDIANTE ADO.NET

El uso de ADO.NET para conectarse a una base de datos en Microsoft Base de datos de SQL Azure es muy similar a la conexión a una instancia de SQL Server en sus instalaciones. En este tema se describen algunas consideraciones para la conexión a Base de datos de SQL Azure y se proporciona una aplicación de consola de ejemplo que puede usar para iniciarse.

Al conectarse a una base de datos en SQL Azure con una aplicación ADO.NET, tenga en cuenta lo siguiente:

- Evite los ataques por inyección de código SQL usando la clase `SqlConnectionStringBuilder`. Está disponible en .NET Framework para simplificar la creación de la cadena de conexión.
- Proteja cuidadosamente la cadena de conexión. Una cadena de conexión presenta una vulnerabilidad potencial si no está protegida.
- Para proteger completamente la conexión, especialmente al conectarse a SQL Azure a través de Internet, asegúrese de que se establecen los parámetros de conexión `Encrypt` y `TrustServerCertificate` de ADO.NET. Establezca la propiedad de conexión `Encrypt` en `True` (`Encrypt = True`) y la propiedad de conexión `TrustServerCertificate` en `False` (`TrustServerCertificate = False`). Esto garantizará que se cifre la conexión e impedirá cualquier ataque de tipo "Man in the middle".

## Ejemplo de aplicación de consola

El ejemplo siguiente es una aplicación de consola que se conecta a Base de datos de SQL Azure. La aplicación de consola realiza estos pasos:

- En primer lugar, la aplicación de consola utiliza un objeto `SqlConnectionStringBuilder` para conectarse a Base de datos de SQL Azure (la base de datos **master**) y crear una base de datos denominada según específica la cadena `sampleDatabaseName`.
- A continuación, la aplicación de consola usa un segundo objeto `SqlConnectionStringBuilder` para conectarse a la nueva base de datos.

- Una vez conectada, la aplicación de consola emplea el segundo objeto `SqlConnectionStringBuilder` para crear una tabla de base de datos y realizar diversas instrucciones del lenguaje de manipulación de datos (DML).
- Por último, la aplicación de consola devuelve datos a la consola antes y después de que cambien los datos.

Para usar este ejemplo en Visual Studio con su servidor SQL Azure, siga esos pasos:

- Abra Visual Studio y cree una nueva aplicación de consola.
- Reemplace el código del archivo de programa con el código de este ejemplo.
- Reemplace `<ProvideUserName>` con el nombre de un inicio de sesión de Base de datos de SQL Azure al que se haya asignado el rol `dbmanager`. Nota: si usa el formato de nombre de usuario `login@server`, la parte `server` del nombre debe coincidir con la primera parte del nombre completo del servidor. Por ejemplo, si su servidor es `servername.database.windows.net`, su nombre de inicio de sesión será `loginname@servername`. Para obtener más información sobre los roles de Base de datos de SQL Azure, vea [Administrar bases de datos e inicios de sesión en SQL Azure](#).
- Reemplace `<ProvidePassword>` con la contraseña asociada al inicio de sesión. Nota: se recomienda emplear una contraseña segura al crear un inicio de sesión. Para obtener más información, vea [Contraseñas seguras](#).
- Reemplace `<ProvideServerName>` con el nombre de dominio completo de su servidor SQL Azure. Por ejemplo: `servername.database.windows.net`
- Reemplace `<ProvideDatabaseName>` con el nombre de la base de datos que desea que cree el código.
- Ejecute el código.

No hay ninguna instrucción de Transact-SQL en este código para quitar la nueva base de datos (denominada según especifica la cadena `sampleDatabaseName`). Si ejecuta este código más de una vez sin quitar la base de datos, se puede producir una excepción porque intentará crear una base de datos que ya existe.

```

Option Explicit On
Option Strict On

Imports System
Imports System.Collections.Generic
Imports System.Linq
Imports System.Text
Imports System.Data.SqlClient
Imports System.Data

Module Program
    ' Provide the following information
    Private ReadOnly userName As String =
"<ProvideUserName>"
    Private ReadOnly password As String =
"<ProvidePassword>"
    Private ReadOnly dataSource As String =
"<ProvideServerName>"
    Private ReadOnly sampleDatabaseName As String =
"<ProvideDatabaseName>"

    Sub Main()
        ' Create a connection string for the master
database
        Dim connString1Builder As New
SqlConnectionStringBuilder()
        With connString1Builder
            .DataSource = dataSource
            .InitialCatalog = "master"
            .Encrypt = True
            .TrustServerCertificate = False
            .UserID = userName
            .Password = password
        End With

        ' Create a connection string for the sample
database
        Dim connString2Builder As New
SqlConnectionStringBuilder()
        With connString2Builder
            .DataSource = dataSource
            .InitialCatalog = sampleDatabaseName
            .Encrypt = True
            .TrustServerCertificate = False
            .UserID = userName
            .Password = password
        End With
    End Sub
End Module

```

```

        ' Connect to the master database and create the
sample database
        Using conn As New
SqlConnection(connString1Builder.ToString())
        Using command As SqlCommand =
conn.CreateCommand()
            conn.Open()

            ' Create the sample database
Dim cmdText As String
cmdText = String.Format("CREATE DATABASE
{0}", _
sampleDatabaseName)
            command.CommandText = cmdText
            command.ExecuteNonQuery()
            conn.Close()
        End Using
    End Using

        ' Connect to the sample database and perform
various operations
        Using conn As New
SqlConnection(connString2Builder.ToString())
        Using command As SqlCommand =
conn.CreateCommand()
            conn.Open()

            ' Create a table
            command.CommandText = "CREATE TABLE
T1(Col1 int primary key, Col2 varchar(20))"
            command.ExecuteNonQuery()

            ' Insert sample records
            command.CommandText = "INSERT INTO T1
(col1, col2) VALUES (1, 'string 1'), (2, 'string 2'), (3,
'string 3')"
            Dim rowsAdded As Integer =
command.ExecuteNonQuery()

            ' Query the table and print the results
            command.CommandText = "SELECT * FROM T1"
            Using reader As SqlDataReader =
command.ExecuteReader()
                ' Loop over the results

```

```

        While reader.Read()
            Console.WriteLine("Col1: {0},
Col2: {1}", _
reader("Col1").ToString().Trim(), _
reader("Col2").ToString().Trim())
        End While
    End Using

    ' Update a record
    command.CommandText = "UPDATE T1 SET
Col2='string 1111' WHERE Col1=1"
    command.ExecuteNonQuery()

    ' Delete a record
    command.CommandText = "DELETE FROM T1
WHERE Col1=2"
    command.ExecuteNonQuery()

    ' Query the table and print the results
    Console.WriteLine("After update/delete,
the table has these records...")
    command.CommandText = "SELECT * FROM T1"
    Using reader As SqlDataReader =
command.ExecuteReader()
        ' Loop over the results
        While reader.Read()
            Console.WriteLine("Col1: {0},
Col2: {1}", _
reader("Col1").ToString().Trim(), _
reader("Col2").ToString().Trim())
        End While
    End Using

    conn.Close()
End Using
End Using

    Console.WriteLine("Press Enter to continue...")
    Console.ReadLine()

End Sub
End Module

```

## CONECTAR CON SQL AZURE MEDIANTE ASP.NET

Puede enlazar datos de Microsoft Base de datos de SQL Azure a controles ASP.NET como lo haría con SQL Server. La clase `SqlDataSource` es compatible con Base de datos de SQL Azure, por lo que puede convertir sus aplicaciones existentes con solo hacer un cambio en la cadena de conexión.

En el ejemplo siguiente, se enlaza el control ASP.NET GridView a la base de datos SQL Azure desde un servicio hospedado en Windows Azure. Con el control **GridView**, puede insertar, actualizar y eliminar datos de una base de datos SQL Azure sin escribir código.

Para seguir este ejemplo, debe instalar Windows Azure SDK y Windows Azure Tools for Microsoft Visual Studio. Sin embargo, la mayoría de estos pasos es aplicable a un proyecto ASP.NET. Para obtener más información sobre Windows Azure, vea Windows Azure SDK. También debe inicializar el servicio de almacenamiento de desarrollo instalado con las herramientas de Windows Azure mediante la utilidad `DSInit`. Almacenamiento de desarrollo se debe inicializar solo una vez.

### Crear la base de datos `testdb`

- Conéctese al servidor SQL Azure mediante la utilidad `sqlcmd`. Para obtener más información, vea Conectar con SQL Azure mediante `sqlcmd`.

- Ejecute los siguientes comandos de Transact-SQL:

```
1> CREATE DATABASE TestDb;  
2> GO  
3> QUIT
```

- Conéctese a `TestDb` y ejecute los siguientes comandos de Transact-SQL:

```
1> CREATE TABLE Table1 (Col1 int primary key, Col2  
varchar(20));  
2> GO  
3> INSERT INTO Table1 (Col1, Col2) VALUES (1, 'string1'),  
(2, 'string2');  
4> GO  
5> QUIT
```

## Crear el control GridView

- En Visual Studio, en el menú **Archivo**, haga clic en **Nuevo** y seleccione **Proyecto**. Debe ejecutar Visual Studio como administrador para poder utilizar el entramado de desarrollo de Windows Azure.
- En el cuadro de diálogo **Nuevo proyecto**, en la lista **Plantillas instaladas**, seleccione **Nube** para Visual C# o Visual Basic.
- En la lista **Plantillas**, seleccione **Proyecto de Windows Azure**. Escriba **EnlaceDeDatos** como nombre y haga clic en **Aceptar**.
- En el cuadro de diálogo **Nuevo proyecto de Windows Azure**, seleccione **Rol web de ASP.NET** en la lista **Roles** y, a continuación, haga clic en la flecha para agregarlo a la lista **Solución de Windows Azure**. Pulse en **Aceptar**.
- En Default.aspx, haga clic en **Diseño** para cambiar a la vista de diseño.
- Arrastre el control **GridView** desde el **Cuadro de herramientas** hasta la superficie de diseño.
- En la ventana **Propiedades** del control GridView, establezca **AutoGenerateDeleteButton** y **AutoGenerateEditButton** en **true**.

## Enlazar el control al origen de datos

- En el **Explorador de soluciones**, en el proyecto Rol web, haga doble clic en **Web.config**.
- En Web.config, agregue lo siguiente:

```
<connectionStrings>
  <add name="SQLAzureConnection" connectionString="Data
Source=<ProvideServerName>.database.windows.net;Initial
Catalog=TestDb;User
ID=<ProvideUserName>;Password=<ProvidePassword>;Encrypt=t
rue;Trusted_Connection=false;"
  providerName="System.Data.SqlClient" />
</connectionStrings>
```

Esto agregará la cadena de conexión a su base de datos SQL Azure.

- En Default.aspx, haga clic en **Origen** para cambiar a la vista de código fuente.
- Agregue el código siguiente para crear un control **SqlDataSource** para su conexión:

```
<asp:SqlDataSource ID="SqlAzureDataSource" runat="server"
    ConnectionString="<%$
ConnectionString:SQLAzureConnection %>"
    InsertCommand="INSERT INTO [Table1] ([Col1], [Col2])
VALUES (@Col1, @Col2)"
    SelectCommand="SELECT * FROM [Table1]"
    UpdateCommand="UPDATE [Table1] SET [Col2] = @Col2
WHERE [Col1] = @Col1"
    DeleteCommand="DELETE FROM [Table1] WHERE [Col1] =
@Col1">
    <UpdateParameters>
        <asp:Parameter Name="Col2" Type="String" />
        <asp:Parameter Name="Col1" Type="Int32" />
    </UpdateParameters>
    <InsertParameters>
        <asp:formParameter Name="Col1" FormField="TextBox1"
/>
        <asp:formParameter Name="Col2" FormField="TextBox2"
/>
    </InsertParameters>
    <DeleteParameters>
        <asp:Parameter Name="Col1" Type="Int32" />
    </DeleteParameters>
</asp:SqlDataSource>
```

Esto incluye la lógica de SELECT, INSERT, UPDATE y DELETE para el control.

- Agregue los siguientes atributos al código del control **GridView**: DataSourceID="SqlAzureDataSource" y DataKeyNames="Col1".
- Presione **F5** para ejecutar la solución.

## CONECTAR CON SQL AZURE MEDIANTE SERVICIOS DE DATOS DE WCF (ADO.NET)

Servicios de datos de Microsoft WCF (anteriormente conocido como ADO.NET Data Services) le permite crear servicios de datos basados en la transferencia de estado de representación Representational State Transfer (REST) y que son accesibles

mediante aplicaciones cliente de .NET Framework. Utilizando el proveedor Entity Framework, puede crear servicios de datos para una base de datos SQL Azure.

En este tema se muestra cómo crear un servicio de datos sencillo basado en una base de datos SQL Azure y una aplicación Windows Azure que tiene acceso al servicio de datos. En este tema se utiliza la base de datos de ejemplo School que se ha implementado en SQL Azure al exponer la migración una base de datos mediante el Asistente para generar scripts

Para seguir este ejemplo, debe instalar Windows Azure SDK y Windows Azure Tools for Microsoft Visual Studio. Para obtener más información sobre Windows Azure, vea Windows Azure SDK. También debe inicializar el servicio de almacenamiento de desarrollo instalado con las herramientas de Windows Azure mediante la utilidad DSInit. Almacenamiento de desarrollo se debe inicializar solo una vez. Para obtener más información, vea el artículo acerca del almacenamiento de desarrollo.

## Crear el modelo de datos

- En Visual Studio, en el menú **Archivo**, seleccione **Nuevo** y, a continuación, haga clic en **Proyecto**. Debe ejecutar Visual Studio como administrador para poder utilizar el entramado de desarrollo de Windows Azure.
- En el panel **Tipos de proyecto**, seleccione **Web**. En el panel **Plantillas**, seleccione **Proyecto de aplicación web ASP.NET**. Escriba **SchoolApplication** como **Nombre** y, a continuación, haga clic en **Aceptar**.
- En el **Explorador de soluciones**, haga clic con el botón secundario en **SchoolApplication**, seleccione **Agregar** y, a continuación, seleccione **Nuevo elemento**.
- Seleccione **ADO.NET Entity Data Model**, escriba **SchoolDataModel.edmx** como **Nombre** y, a continuación, pulse en **Agregar**.
- En el **Asistente para Entity Data Model**, seleccione **Generar desde la base de datos** y, a continuación, haga clic en **Siguiente**.
- Haga clic en **Nueva conexión** y escriba la información para la base de datos School en su servidor SQL Azure; a continuación, pulse en **Aceptar**. Elija la opción **No, excluir datos confidenciales de la cadena de conexión. Los estableceré en el código de mi aplicación**. A continuación, haga clic en **Siguiente**.

- Seleccione **Tablas** y, a continuación, haga clic en **Finalizar**. Esto crea un archivo .edmx que define un tipo de entidad para cada tabla de la base de datos.

Para seguir este ejemplo en Visual Studio 2010, desactive la casilla **Poner en plural o en singular los nombres de objeto generados**. Cuando se ejecuta este ejemplo en Visual Studio 2008, Entity Framework no puede crear el archivo .edmx ni las clases cliente relacionadas a partir de una base de datos SQL Azure. Debe conectarse a una instancia local de SQL Server para generar los archivos de asignación y de modelo. Sin embargo, Entity Framework se conectará a Base de datos de SQL Azure cuando se hayan creado los archivos de asignación y de modelo.

## Crear el servicio de datos

- En el **Explorador de soluciones**, haga clic con el botón secundario en **SchoolApplication**, seleccione **Agregar** y, a continuación, seleccione **Nuevo elemento**.
- Seleccione **Servicio de datos de WCF**, escriba **SchoolDataService.svc** como **Nombre** y, a continuación, haga clic en **Agregar**.
- En SchoolDataService.svc.cs o en SchoolDataService.svc.vb, reemplace el comentario `/* TODO: put your data source class name here */` de la definición de la clase que define el servicio de datos con el tipo que es el contenedor de entidad del modelo de datos (en este caso, SchoolEntities). La definición de clase debe ser similar a la siguiente:

```
Inherits DataService(Of SchoolEntities)
```

- Reemplace el código del marcador de posición en la función InitializeService con el código siguiente, que concede acceso a las entidades Course, Department y Person:

```
Public Shared Sub InitializeService(ByVal config As
IDataServiceConfiguration)
    config.SetEntitySetAccessRule("Course",
EntitySetRights.All)
    config.SetEntitySetAccessRule("Department",
EntitySetRights.All)
    config.SetEntitySetAccessRule("Person",
EntitySetRights.All)
End Sub
```

## Crear la aplicación cliente

- En el **Explorador de soluciones**, haga clic con el botón secundario en la solución, seleccione **Agregar y**, a continuación, haga clic en **Nuevo proyecto**.
- En el cuadro de diálogo **Agregar nuevo proyecto**, en el panel **Tipos de proyecto**, seleccione **Servicio de nube**. Escriba **SchoolSite** como **Nombre** y, a continuación, haga clic en **Aceptar**.
- En el cuadro de diálogo **Nuevo proyecto de servicio de nube**, agregue un **Rol web de ASP.NET** al panel **Solución de servicio de nube** y, a continuación, haga clic en **Aceptar**.
- Reemplace el código fuente de Default.aspx con el código siguiente:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1">
        <div>

            <asp:DropDownList ID="peopleList" runat="server"
onselectedindexchanged="peopleList_SelectedIndexChanged"
AutoPostBack="True">
                </asp:DropDownList>
                <br />
                <br />
                <asp:Label ID="infoLabel"
runat="server"></asp:Label>
                <br />

            </div>
        </form>
</body>
</html>
```

Esto crea una lista desplegable denominada **peopleList** y un control **Label** denominado **infoLabel**.

- Presione F6 para compilar la aplicación.

- En el **Explorador de soluciones**, haga clic con el botón secundario en el proyecto de rol web y, a continuación, en **Agregar referencia de servicio**.
- En el cuadro de diálogo **Agregar referencia de servicio**, haga clic en **Detectar**. SchoolDataService.svc aparecerá en el panel **Servicios**. Escriba **SchoolServiceReference** como **Nombre** y, a continuación, haga clic en **Aceptar**. Esto agrega el archivo que contiene las clases de datos que tienen acceso a los recursos del servicio de datos.

## Tener acceso al servicio de datos School desde el rol web de Windows Azure

- En **Default.aspx.cs** o en **Default.aspx.vb**, agregue las siguientes instrucciones **using** (**Imports** en Visual Basic):

```
Imports System.Data.Services.Client
Imports WebRole1.SchoolServiceReference
```

- En la clase parcial, agregue las siguientes declaraciones para DataServiceContext y URI al servicio de datos (el identificador URI que aparecía anteriormente en el campo **Dirección** del cuadro de diálogo **Agregar referencia de servicio**):

```
Private schoolContext As SchoolEntities

' Replace the host server and port number with the values
'   for the test server that is hosting your data service
instance.
Private svcUri As Uri = New
Uri("http://localhost:61925/SchoolDataService.svc")
```

- En el método Page\_Load, cree instancias del contexto del servicio de datos y cargue la lista desplegable con información del servicio de datos agregando el código siguiente:

```
' Instantiate the DataServiceContext
schoolContext = New SchoolEntities(svcUri)

' Get all person entities from the data service
Dim people As DataServiceQuery(Of Person) =
schoolContext.Person

' Add each person to the drop down list
For Each p As Person In people
```

```

        peopleList.Items.Add(New
        ListItem(String.Concat(p.FirstName.ToString(), " ", _
        p.LastName.ToString()), p.PersonID.ToString()))
        Next

```

- En la vista Diseño de Default.aspx, haga doble clic en la lista desplegable para agregar el evento `peopleList_SelectedIndexChanged`.
- En el evento `peopleList_SelectedIndexChanged`, rellene el control **Label** basándose en la persona seleccionada agregando el código siguiente:

```

Protected Sub peopleList_SelectedIndexChanged(ByVal
sender As Object, ByVal e As EventArgs) Handles
peopleList.SelectedIndexChanged
    ' Define a LINQ query that returns information
    ' about the selected person.
    Dim personInfo = (From p In schoolContext.Person _
        Where p.PersonID =
        Convert.ToInt32(peopleList.SelectedItem.Value) _
        Select p).FirstOrDefault()

    ' Display information about the person
    infoLabel.Text = String.Concat("ID: ",
        personInfo.PersonID.ToString(), " ", _
        "Enroll Date:",
        personInfo.EnrollmentDate.ToString(), " ", _
        "Hire Date: ",
        personInfo.HireDate.ToString())
    infoLabel.Visible = True
End Sub

```

## Migrar la aplicación a SQL Azure

- Migre la base de datos School a SQL Azure mediante el Asistente para generar scripts.
- En el proyecto **SchoolApplication**, abra el archivo **Web.config**. Cambie la cadena de conexión para que se conecte a su base de datos SQL Azure.

```

<connectionStrings>
    <add name="SchoolEntities"
        connectionString="metadata=res://*/SchoolDataModel.csdl|r
es://*/SchoolDataModel.ssdl|res://*/SchoolDataModel.msl;p
rovider=System.Data.SqlClient;provider connection
string=&quot;Data

```

```
Source=<provideServerName>.database.windows.net;Initial
Catalog=School;Integrated Security=False;User
ID=<provideUserID>;Password=<providePassword>;MultipleAct
iveResultSets=False;Encrypt=True;TrustServerCertificate=F
alse&quot;" providerName="System.Data.EntityClient"/>
</connectionStrings>
```

- En el **Explorador de soluciones**, haga clic con el botón secundario en el proyecto WebRole1 y seleccione **Establecer como proyecto de inicio**.
- Presione F5 para ejecutar la aplicación.

## CONECTAR CON SQL AZURE MEDIANTE PHP

Puede conectarse a Microsoft Base de datos de SQL Azure mediante PHP y Controlador de SQL Server para PHP a partir de la versión 1.1. En este tema se muestra cómo crear una aplicación PHP dentro de un rol web de CGI de Windows Azure que se conecta a Base de datos de SQL Azure. Siga los pasos descritos en la sección Introducción de la documentación de Controlador de SQL Server para PHP para preparar el servidor para el desarrollo con PHP y SQL Server. Puede descargar Controlador de SQL Server para PHP del sitio web del Centro de descarga de Microsoft. Para obtener más información sobre Controlador de SQL Server para PHP, consulte la documentación del Controlador de SQL Server para PHP en MSDN.

### Crear la aplicación Windows Azure

- En Visual Studio, en el menú **Archivo**, haga clic en **Nuevo** y seleccione **Proyecto**. Debe ejecutar Visual Studio como administrador para poder utilizar el entramado de desarrollo de Windows Azure.
- En el cuadro de diálogo **Agregar nuevo proyecto**, en la lista **Tipos de proyecto**, seleccione **Servicio de nube** para Visual C# o Visual Basic.
- En el panel Plantillas, seleccione Servicio de nube de Windows Azure.
- Escriba **PHPCGIExample** como **Nombre** y, a continuación, haga clic en **Aceptar**.
- En el cuadro de diálogo **Nuevo proyecto de servicio de nube**, agregue un rol de web de CGI al panel **Solución de servicio de nube** y, a continuación, haga clic en **Aceptar**.
- Cree una nueva carpeta PHP en su proyecto de rol web y copie los archivos de la carpeta de instalación de PHP, como c:\archivos de programa\PHP, a su nueva

carpeta PHP del proyecto. Los binarios PHP en tiempo de ejecución deben formar parte de la implementación.

- En el **Explorador de soluciones**, haga clic con el botón secundario en **WebCgiRole**, seleccione **Agregar** y, a continuación, pulse en **Elemento existente**. Examine y agregue **php-cgi.exe** a su proyecto de rol web.
- En el panel **Propiedades** para **php-cgi.exe**, establezca **Acción de compilación** en **Contenido** y **Copiar en el directorio de salida** en **Copiar siempre**.

## Crear la aplicación PHP

- Edite el archivo **web.config** para configurar su controlador **FastCGI**. Busque la sección **<handlers>** bajo el nodo **<system.webServer>** y agregue lo siguiente:

```
<add name="PHP_FastCGI"
      verb="*"
      path="*.php"
      scriptProcessor="%RoleRoot%\approot\php-cgi.exe"
      modules="FastCgiModule"
      resourceType="Unspecified" />
```

- Edite el archivo **web.roleconfig** para asegurarse de que la sección **<fastCgi>** tiene el siguiente valor:

```
<application fullPath="%RoleRoot%\approot\php-cgi.exe"/>
```

- En el **Explorador de soluciones**, haga clic con el botón secundario en **WebCgiRole**, seleccione **Agregar** y, a continuación, en **Nuevo elemento**.
- En el cuadro de diálogo **Agregar nuevo elemento**, seleccione **Archivo de texto** en el panel **Plantillas**. Escriba **PHPData.php** como **Nombre** y, a continuación, haga clic en **Agregar**.
- En el panel **Propiedades** para **PHPData.php**, establezca **Acción de compilación** en **Contenido** y **Copiar en el directorio de salida** en **Copiar siempre**.
- En el **Explorador de soluciones**, haga clic con el botón secundario en el proyecto **WebCgiRole** y seleccione **Propiedades**. En la ventana **Propiedades**, seleccione **Web**. En el panel **Acción de inicio**, haga clic en **Página específica** y escriba **PHPData.php**.
- En **PHPData.php**, pegue el código siguiente:

```

<?php

    $serverName                                     =
"tcp:ProvideServerName.database.windows.net,1433";
    $userName = 'ProvideUserName@ProvideServerName';
    $userPassword = 'ProvidePassword';
    $dbName = "TestDB";
    $table = "tablePHP";

    $connectionInfo = array("Database"=>$dbName,
"UID"=>$userName,                "PWD"=>$userPassword,
"MultipleActiveResultSets"=>true);

    sqlsrv_configure('WarningsReturnAsErrors', 0);
    $conn = sqlsrv_connect( $serverName, $connectionInfo);
    if($conn === false)
    {
        FatalError("Failed to connect...");
    }

    CreateTable($conn, $table, "Col1 int primary key, Col2
varchar(20)");

    $tsql = "INSERT INTO [$table] (Col1, Col2) VALUES (1,
'string1'), (2, 'string2)";
    $stmt = sqlsrv_query($conn, $tsql);
    if ($stmt === false)
    {
        FatalError("Failed to insert data into test table:
".$tsql);
    }
    sqlsrv_free_stmt($stmt);

    $tsql = "SELECT Col1, Col2 FROM [$table]";
    $stmt = sqlsrv_query($conn, $tsql);
    if ($stmt === false)
    {
        FatalError("Failed to query test table: ".$tsql);
    }
    else
    {
        while($row = sqlsrv_fetch_array($stmt,
SQLSRV_FETCH_NUMERIC))
        {
            echo "Col1: ".$row[0]."\n";
            echo "Col2: ".$row[1]."\n";
        }
    }

```

```

        sqlsrv_free_stmt($stmt);
    }

    sqlsrv_close($conn);

function CreateTable($conn, $tableName, $dataType)
{
    $sql = "CREATE TABLE [$tableName] ($dataType)";
    DropTable($conn,$tableName);
    $stmt = sqlsrv_query($conn, $sql);
    if ($stmt === false)
    {
        FatalError("Failed to create test table: ".$sql);
    }
    sqlsrv_free_stmt($stmt);
}

function DropTable($conn, $tableName)
{
    $stmt = sqlsrv_query($conn, "DROP TABLE
[$tableName]");
    if ($stmt === false)
    {
    }
    else
    {
        sqlsrv_free_stmt($stmt);
    }
}

function FatalError($errorMsg)
{
    Handle_Errors();
    die($errorMsg."\n");
}

function Handle_Errors()
{
    $errors = sqlsrv_errors(SQLSRV_ERR_ERRORS);
    $count = count($errors);
    if($count == 0)
    {
        $errors = sqlsrv_errors(SQLSRV_ERR_ALL);
        $count = count($errors);
    }
    if($count > 0)

```

```

    {
      for($i = 0; $i < $count; $i++)
      {
        echo $errors[$i]['message']."\n";
      }
    }
  }
?>

```

- Presione **F5** para ejecutar la solución.

## CONECTAR CON SQL AZURE MEDIANTE JDBC

Puede establecer una conexión con Base de datos de Microsoft SQL Azure utilizando Java y Controlador JDBC de Microsoft SQL Server a partir de la versión 3.0. En este tema se muestra la forma de crear una aplicación Java que establezca conexión con la base de datos SQL Azure. Siga estos pasos en la sección Requisitos del sistema para el controlador JDBC de la documentación de Controlador JDBC de Microsoft SQL Server para preparar su equipo para el desarrollo con Java y la base de datos SQL Azure.

### Preparar el entorno de desarrollo

- Descargue e instale la versión actualizada de Controlador JDBC de SQL Server 3.0 que admite la base de datos SQL Azure desde aquí.
- Para ejecutar esta aplicación de ejemplo, debe establecer la ruta de acceso de clase de forma que ejecute el archivo sqljdbc.jar o el archivo sqljdbc4.jar. Si en la ruta de acceso de clase falta una entrada para sqljdbc.jar o sqljdbc4.jar, la aplicación de ejemplo producirá la excepción común "No se encuentra la clase".
- También necesitará acceso a la base de datos de ejemplo School.

### Ejemplo

En el ejemplo siguiente, el código muestra establece unas propiedades de conexión en la cadena de conexión y establece una conexión con la base de datos de ejemplo School. A continuación, el código muestra ejecuta la instrucción SQL y realiza una iteración por las filas de datos contenidas en el conjunto de resultados, y utiliza el método getString para mostrar algunos de los datos que contiene.

Compile y ejecute esta aplicación mediante las utilidades de línea de comandos de Java o su IDE preferido.

```
import java.sql.*;
import com.microsoft.sqlserver.jdbc.*;

public class ConnectToSQLAzure {

    public static void main(String[] args) {

        // Create a variable for the connection string.
        String connectionUrl =
        "jdbc:sqlserver://servername.database.windows.net;" +

        "databaseName=School;user=username@servername;password=us
        erpassword";

        // Declare the JDBC objects.
        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;

        try {
            // Establish the connection.

            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriv
            er");
            con =
            DriverManager.getConnection(connectionUrl);

            // Create and execute an SQL statement that
            returns some data.
            String SQL = "SELECT TOP 10 * FROM dbo.Person";
            stmt = con.createStatement();
            rs = stmt.executeQuery(SQL);

            // Iterate through the data in the result set
            and display it.
            while (rs.next()) {
                System.out.println(rs.getString(2) + " " +
                rs.getString(3));
            }
        }

        // Handle any errors that may have occurred.
        catch (Exception e) {
```

```

        e.printStackTrace();
    }
    finally {
        if (rs != null) try { rs.close(); }
    catch(Exception e) {}
        if (stmt != null) try { stmt.close(); }
    catch(Exception e) {}
        if (con != null) try { con.close(); }
    catch(Exception e) {}
    }
}
}

```

## CONECTAR CON SQL AZURE MEDIANTE ADO.NET ENTITY FRAMEWORK

Entity Framework ADO.NET permite a los desarrolladores crear aplicaciones de acceso a datos programando en un modelo de aplicación conceptual en lugar de programar directamente en un esquema de almacenamiento relacional.

A continuación, se muestra cómo crear un modelo conceptual sencillo asignado a una base de datos SQL Azure y consultar en el modelo conceptual. En el siguiente ejemplo de Visual C#/Visual Basic se crea una consulta fuertemente tipada de los objetos CLR que representan entidades y asociaciones en el modelo School, que devuelve los nombres de departamento del modelo School.

Para seguir este ejemplo, debe instalar Entity Framework desde .NET Framework 3.5 Service Pack 1 o versiones posteriores.

### Crear el modelo de datos

- En Visual Studio, en el menú **Archivo**, seleccione **Nuevo** y, a continuación, haga clic en **Proyecto**.
- En el panel **Tipos de proyecto**, seleccione **Windows**. En el panel **Plantillas**, seleccione **Aplicación de consola**. Escriba **SchoolApplicationEF** como **Nombre** y, a continuación, haga clic en **Aceptar**.
- En el **Explorador de soluciones**, haga clic con el botón secundario en **SchoolApplicationEF**, seleccione **Agregar** y, a continuación, seleccione **Nuevo elemento**.

- Seleccione **ADO.NET Entity Data Model**, escriba **SchoolDataModel.edmx** como **Nombre** y, a continuación, haga clic en **Agregar**.
- En el **Asistente para Entity Data Model**, seleccione **Generar desde la base de datos** y, a continuación, haga clic en **Siguiente**.
- Haga clic en **Nueva conexión** y escriba la información para la base de datos School en su servidor SQL Azure; a continuación, pulse en **Aceptar**. Elija la opción **No, excluir datos confidenciales de la cadena de conexión. Los estableceré en el código de mi aplicación**. A continuación, haga clic en **Siguiente**. Para obtener información sobre cómo migrar la base de datos School a SQL Azure, vea Migrar una base de datos mediante el Asistente para generar scripts (Base de datos de SQL Azure).
- Seleccione **Tablas** y, a continuación, haga clic en **Finalizar**. Esto crea un archivo .edmx que define un tipo de entidad para cada tabla de la base de datos.
- Guarde el archivo **SchoolDataModel.edmx**.

Cuando se ejecuta este ejemplo en Visual Studio 2008, Entity Framework no puede crear el archivo .edmx ni las clases cliente relacionadas a partir de una base de datos SQL Azure. Debe conectarse a una instancia local de SQL Server para generar los archivos de asignación y de modelo. Sin embargo, Entity Framework se conectará a Base de datos de SQL Azure cuando se hayan creado los archivos de asignación y de modelo. Además, la casilla **Poner en plural o en singular los nombres de objeto generados** no existe en Visual Studio 2008. Por tanto, debe actualizar Departments como Department y Courses como Course en el ejemplo de código siguiente.

## Crear la aplicación de consola

- Reemplace el código del archivo de programa con el código siguiente y presione F5 para ejecutar la aplicación en su base de datos SQL Azure.

```
Imports System
Imports System.Collections.Generic
Imports System.Linq
Imports System.Text

Module Module1

    Private schoolContext As SchoolEntities

    Sub Main()
```

```

' Initialize theObjectContext.
Using schoolContext = New SchoolEntities()

    ' Define a query that returns all Department
objects
    ' and related Course objects, ordered by
name.
    Dim departmentQuery = _
        From d In
schoolContext.Departments.Include("Courses") _
        Order By d.Name _
        Select d

    Try
        For Each dep As Department In
departmentQuery
            Console.WriteLine(dep.Name)
        Next

        Catch ex As Exception
            Console.WriteLine("An error occurred: " +
ex.Message)
        End Try

    End Using

    Console.WriteLine("Hit enter...")
    Console.Read()

End Sub
End Module

```

## Migrar la aplicación a SQL Azure

- Migre la base de datos School a SQL Azure siguiendo las instrucciones descritas para la migración de una base de datos mediante el Asistente para generar scripts.
- En el proyecto **SchoolEFApplication**, abra el archivo **App.Config**. Cambie la cadena de conexión para que se conecte a su base de datos SQL Azure.

```

<connectionStrings>
  <add name="SchoolEntities"
connectionString="metadata=res://*/SchoolDataModel.csdl|r
es://*/SchoolDataModel.ssdl|res://*/SchoolDataModel.msl;p
rovider=System.Data.SqlClient;provider connection

```

```
string=&quot;;Data
Source=<provideServerName>.database.windows.net;Initial
Catalog=School;Integrated Security=False;User
ID=<provideUserID>;Password=<providePassword>;MultipleAct
iveResultSets=True;Encrypt=True;TrustServerCertificate=Fa
lse&quot;; providerName="System.Data.EntityClient"/>
</connectionStrings>
```

- Presione F5 para ejecutar la aplicación en su base de datos SQL Azure.

# TRANSACT-SQL EN SQL AZURE. TIPOS DE DATOS, FUNCIONES Y OPERADORES

## TIPOS DE DATOS EN BASE DE DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure admite la gramática de Transact-SQL que se puede usar para consultar una base de datos e insertar, actualizar y eliminar datos de tablas de una base de datos. En los apartados de este capítulo se describe la compatibilidad con la gramática de Transact-SQL proporcionada por Base de datos de SQL Azure.

Microsoft Base de datos de SQL Azure proporciona compatibilidad con muchos tipos de datos del sistema de SQL Server.

En la tabla siguiente se muestran las categorías de tipos de datos y también se describe la compatibilidad con los tipos de datos del sistema:

Categoría de tipos de datos	Compatibilidad de SQL Azure
Valores numéricos exactos	Se admiten: <b>bigint, bit, decimal, int, money, numeric, smallint, smallmoney, tinyint.</b>
Valores numéricos aproximados	Se admiten: <b>float, real.</b>

Fecha y hora	Se admiten: <b>date</b> , <b>datetime2</b> , <b>datetime</b> , <b>datetimeoffset</b> , <b>smalldatetime</b> , <b>time</b> .
Cadenas de caracteres	Se admiten: <b>char</b> , <b>varchar</b> , <b>text</b> .
Cadenas de caracteres Unicode	Se admiten: <b>nchar</b> , <b>nvarchar</b> , <b>ntext</b> .
Cadenas binarias	Se admiten: <b>binary</b> , <b>varbinary</b> , <b>image</b> .
Tipos de datos espaciales	Se admiten: <b>geography</b> , <b>geometry</b> .
Otros tipos de datos	Se admiten: <b>cursor</b> , <b>hierarchyid</b> , <b>sql_variant</b> , <b>table</b> , <b>timestamp</b> , <b>uniqueidentifier</b> , <b>xml</b> .

## Compatibilidad con métodos del tipo de datos geography

Base de datos de SQL Azure admite los siguientes métodos OGC en el tipo de datos **geography**:

STArea	STEndPoint	STLength
STAsBinary	STEquals	STNumGeometries
STAsText	STGeometryN	STNumPoints
STBuffer	STGeometryType	STPointN
STDifference	STIntersection	STSrid
STDimension	STIntersects	STStartPoint
STDisjoint	STIsClosed	STSymDifference
STDistance	STIsEmpty	STUnion

Base de datos de SQL Azure admite los siguientes métodos estáticos OGC en el tipo de datos **geography**:

STGeomFromText	STMPolyFromText	STPolyFromWKB
STPointFromText	STGeomCollFromText	STMPointFromWKB
STLineFromText	STGeomCollFromWKB	STMLineFromWKB
STPolyFromText	STGeomFromWKB	STMPolyFromWKB
STMPointFromText	STPointFromWKB	
STMLineFromText	STLineFromWKB	

Base de datos de SQL Azure admite los siguientes métodos extendidos en el tipo de datos **geography**:

AsGml	InstanceOf	Reduce
AsTextZM	IsNull	ToString
BufferWithTolerance	M	Z
Filter	MakeValid	

Base de datos de SQL Azure admite los siguientes métodos estáticos extendidos en el tipo de datos **geography**:

GeomFromGml	Parse	Point
Null		

## Compatibilidad con métodos del tipo de datos geometry

Base de datos de SQL Azure admite los siguientes métodos OGC en el tipo de datos **geometry**:

STArea	STEquals	STNumPoints
STAsBinary	STExteriorRing	STOverlaps
STAsText	STGeometryN	STPointN
STBoundary	STGeometryType	STPointOnSurface
STBuffer	STInteriorRingN	STRelate
STCentroid	STIntersection	STSrid
STContains	STIntersects	STStartPoint
STConvexHull	STIsClosed	STSymDifference
STCrosses	STIsEmpty	STTouches
STDifference	STIsRing	STUnion
STDimension	STIsSimple	STWithin
STDisjoint	STIsValid	STX
STDistance	STLength	STY
STEndPoint	STNumGeometries	
STEnvelope	STNumInteriorRing	

Base de datos de SQL Azure admite los siguientes métodos estáticos OGC en el tipo de datos **geometry**:

STSGeomFromText	STMPolyFromText	STPolyFromWKB
STPointFromText	STGeomCollFromText	STMPointFromWKB

STLineFromText	STGeomCollFromWKB	STMLineFromWKB
STPolyFromText	STGeomFromWKB	STPolyFromWKB
STMPointFromText	STPointFromWKB	
STMLineFromText	STLineFromWKB	

Base de datos de SQL Azure admite los siguientes métodos extendidos en el tipo de datos **geometry**:

AsGml	InstanceOf	Reduce
AsTextZM	IsNull	ToString
BufferWithTolerance	M	Z
Filter	MakeValid	

Base de datos de SQL Azure admite los siguientes métodos estáticos extendidos en el tipo de datos **geometry**:

GeomFromGml	Parse	Point
Null		

## Compatibilidad con métodos del tipo de datos hierarchyid

Base de datos de SQL Azure admite los siguientes métodos en el tipo de datos **hierarchyid**:

GetAncestor	IsDescendantOf	ToString
GetDescendant	Parse	Write
GetLevel	Read	
GetRoot	GetReparentedValue	

## El tipo de datos xml

Base de datos de SQL Azure admite el tipo de datos **xml** que almacena datos XML. Puede almacenar instancias de **xml** en una columna o en una variable del tipo **xml**.

El lenguaje de manipulación de datos XML (XML DML) es una extensión del lenguaje XQuery. XML DML agrega las siguientes palabras clave con distinción entre mayúsculas y minúsculas a XQuery y se admiten en Base de datos de SQL Azure:

- insert (XML DML)

- delete (XML DML)
- replace value of (XML DML)

Puede utilizar los métodos del tipo de datos **xml** para consultar una instancia XML almacenada en una variable o columna del tipo **xml**. Base de datos de SQL Azure admite los siguientes métodos del tipo de datos **xml**:

- query() (método del tipo de datos xml)
- value() (método del tipo de datos xml)
- exist() (método del tipo de datos xml)
- modify() (método del tipo de datos xml)
- nodes() (método del tipo de datos XML)

## FUNCIONES DE BASE DE DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure proporciona compatibilidad con muchas funciones integradas de SQL Server. En este tema se describen las funciones integradas compatibles y no compatibles.

En la tabla siguiente se enumeran las categorías de funciones y también se describe la compatibilidad con funciones integradas:

Categoría de función	Descripción	Compatibilidad con Base de datos de SQL Azure
Funciones de agregado	Operan sobre una colección de valores y devuelven un solo valor de resumen.	Sí
Funciones de categoría	Devuelven un valor de categoría para cada fila de una partición.	Sí
Funciones de conjuntos de filas	Devuelven un objeto que se puede utilizar como las referencias de tabla en una instrucción SQL.	No
Funciones escalares	Operan sobre un solo valor y devuelven otro valor. Las funciones escalares se pueden utilizar donde la expresión sea válida.	Sí pero Base de datos de SQL Azure no admite todas las funciones escalares

Funciones escalares de ODBC	Funciones especificadas por ODBC. Se pueden usar en procedimientos almacenados.	Sí
-----------------------------	---	----

## Funciones de agregado

Las funciones de agregado realizan un cálculo sobre un conjunto de valores y devuelven un solo valor. En la tabla siguiente se enumeran las funciones de agregado compatibles con Base de datos de SQL Azure.

AVG	COUNT_BIG	SUM
BINARY_CHECKSUM	GROUPING	STDEV
CHECKSUM	GROUPING_ID	STDEVP
CHECKSUM_AGG	MAX	VAR
COUNT	MIN	VARP

## Funciones de categoría

Las funciones de categoría devuelven un valor de categoría para cada fila de una partición. En la tabla siguiente se enumeran las funciones de categoría compatibles con Base de datos de SQL Azure.

DENSE_RANK	RANK
NTILE	ROW_NUMBER

## Funciones de conjuntos de filas

Las siguientes funciones de conjunto de filas devuelven un objeto que se puede utilizar, en instrucciones Transact-SQL, en lugar de una referencia de tabla. En la tabla siguiente se enumeran las funciones de conjuntos de filas **no compatibles** con Base de datos de SQL Azure.

CONTAINSTABLE	OPENDATASOURCE	OPENROWSET
FREETEXTTABLE	OPENQUERY	OPENXML

## Funciones escalares

En la tabla siguiente se enumeran las categorías de funciones escalares y también se describe la compatibilidad proporcionada por Base de datos de SQL Azure.

Categoría de función	Descripción	Compatibilidad con Base de datos de SQL Azure
Funciones de configuración	Devuelven información sobre la configuración actual.	Compatibilidad parcial.
Funciones del cursor	Devuelven información sobre los cursores.	Sí
Funciones de fecha y hora	Llevan a cabo operaciones sobre un valor de entrada de fecha y hora, y devuelven un valor numérico, de cadena o de fecha y hora.	Compatibilidad parcial.
Funciones matemáticas	Realizan cálculos basados en valores de entrada proporcionados como parámetros a las funciones y devuelven valores numéricos.	Sí
Funciones de metadatos	Devuelven información sobre la base de datos y los objetos de base de datos.	Compatibilidad parcial.
Funciones de seguridad	Devuelven información sobre usuarios y roles.	Compatibilidad parcial.
Funciones de cadenas	Realizan operaciones en el valor de entrada de una cadena ( <b>char</b> o <b>varchar</b> ) y devuelven una cadena o un valor numérico.	Sí
Funciones del sistema	Realizan operaciones y devuelven información sobre valores, objetos y configuraciones de una base de datos.	Compatibilidad parcial.
Funciones estadísticas del sistema	Devuelven información estadística sobre el sistema.	No
Funciones de texto y de imagen	Realizan operaciones sobre los valores de entrada o columnas de texto o imagen, y devuelven información sobre el valor.	Compatibilidad parcial.

## Funciones de configuración

En la tabla siguiente se enumeran las funciones de configuración compatibles con Base de datos de SQL Azure.

@@DATEFIRST	@@LOCK_TIMEOUT	@@SERVERNAME
@@DBTS	@@MAX_PRECISION	@@SPID
@@LANGID	@@NESTLEVEL	@@TEXTSIZE
@@LANGUAGE	@@OPTIONS	@@VERSION (Base de datos de SQL Azure)

Base de datos de SQL Azure **no admite** las siguientes funciones de configuración: CONNECTIONPROPERTY, @@MAX\_CONNECTIONS, @@REMSERVER, @@SERVICENAME.

## Funciones del cursor

En la tabla siguiente se enumeran las funciones del cursor compatibles con Base de datos de SQL Azure. Estas funciones escalares devuelven información sobre los cursores.

@@CURSOR_ROWS	@@FETCH_STATUS	CURSOR_STATUS
---------------	----------------	---------------

## Funciones de fecha y hora

En la tabla siguiente se enumeran las funciones de fecha y hora compatibles con Base de datos de SQL Azure:

@@DATEFIRST	DATEPART	SWITCHOFFSET
@@LANGUAGE	DAY	SYSDATETIME
CURRENT_TIMESTAMP	GETDATE	SYSDATETIMEOFFSET
DATEADD	GETUTCDATE	SYSUTCDATETIME
DATEDIFF	ISDATE	TODATETIMEOFFSET
DATENAME	MONTH	YEAR

## Funciones matemáticas

Las siguientes funciones escalares realizan un cálculo, normalmente basado en valores de entrada proporcionados como argumentos, y devuelven un valor numérico. En la tabla siguiente se enumeran las funciones matemáticas compatibles con Base de datos de SQL Azure:

ABS	DEGREES	RAND
ACOS	EXP	ROUND
ASIN	FLOOR	SIGN

ATAN	LOG	SIN
ATN2	LOG10	SQRT
CEILING	PI	SQUARE
COS	POWER	TAN
COT	RADIANS	

## Funciones de metadatos

Las siguientes funciones escalares devuelven información sobre la base de datos y los objetos de base de datos. En la tabla siguiente se enumeran las funciones de metadatos compatibles con Base de datos de SQL Azure:

@@PROCID	INDEX_COL	SCHEMA_ID
COL_LENGTH	INDEXPROPERTY	SCHEMA_NAME
COL_NAME	OBJECT_DEFINITION	SQL_VARIANT_PROPERTY
COLUMNPROPERTY	OBJECT_ID	TYPE_ID
DATABASE_PRINCIPAL_ID	OBJECT_NAME	TYPE_NAME
DATABASEPROPERTYEX (Base de datos de SQL Azure)	OBJECT_SCHEMA_NAME	TYPEPROPERTY
DB_ID	OBJECTPROPERTY	
DB_NAME	OBJECTPROPERTYEX	

En la tabla siguiente se enumeran las funciones de metadatos **no compatibles** con Base de datos de SQL Azure:

ASSEMBLYPROPERTY	FILE_NAME	FULLTEXTSERVICEPROPERTY
ASYMKEY_ID	FILEGROUP_ID	INDEXKEY_PROPERTY
ASYMKEYPROPERTY	FILEGROUP_NAME	Key_GUID
Cert_ID	FILEGROUPPROPERTY	Key_ID
DATABASEPROPERTY	FILEPROPERTY	KEY_NAME
FILE_ID	fn_listextendedproperty	SYMKEYPROPERTY
FILE_IDEX	FULLTEXTCATALOGPROPERTY	

## Funciones de seguridad

Las siguientes funciones devuelven información útil para la administración de la seguridad. En la tabla siguiente se enumeran las funciones de seguridad compatibles con Base de datos de SQL Azure:

CURRENT_USER	SESSION_USER	USER_NAME
HAS_PERMS_BY_NAME (Base de datos de SQL Azure)	SUSER_SID (Base de datos de SQL Azure)	SUSER_SNAME (Base de datos de SQL Azure)
IS_MEMBER (Base de datos de SQL Azure)	USER	
IS_SRVROLEMEMBER (Base de datos de SQL Azure)	USER_ID	

En la tabla siguiente se enumeran las funciones de seguridad **no compatibles** con Base de datos de SQL Azure:

fn_Trace_Geteventinfo	fn_my_permissions	SUSER_NAME
fn_Trace_Getfilterinfo	PERMISSIONS	SYSTEM_USER
fn_Trace_Getinfo	SETUSER	sys.fn_builtin_permissions
fn_Trace_Gettable	SUSER_ID	

## Funciones de cadenas

Las siguientes funciones escalares realizan una operación sobre un valor de cadena de entrada y devuelven un valor de cadena o un valor numérico. En la tabla siguiente se enumeran las funciones de cadenas compatibles con Base de datos de SQL Azure:

ASCII	NCHAR	SOUNDEX
CHAR	PATINDEX	SPACE
CHARINDEX	QUOTENAME	STR
DIFFERENCE	REPLACE	STUFF
LEFT	REPLICATE	SUBSTRING
LEN	REVERSE	UNICODE
LOWER	RIGHT	UPPER
LTRIM	RTRIM	

## Funciones del sistema

En la tabla siguiente se enumeran las funciones del sistema compatibles con Base de datos de SQL Azure:

APP_NAME	ERROR_PROCEDURE	PARSENAME
CASE	ERROR_SEVERITY	@@ROWCOUNT
CAST Y CONVERT	ERROR_STATE	ROWCOUNT_BIG
COALESCE	FORMATMESSAGE	SCOPE_IDENTITY
COLLATIONPROPERTY	GETANSINULL	SERVERPROPERTY (Base de datos de SQL Azure)
COLUMNS_UPDATED	IDENT_CURRENT	SESSIONPROPERTY
CONVERT	IDENT_INCR	SESSION_USER
CURRENT_TIMESTAMP	IDENT_SEED	STATS_DATE
CURRENT_USER	@@IDENTITY	@@TRANCOUNT
DATALENGTH	ISDATE	UPDATE()
@@ERROR	ISNULL	USER_NAME
ERROR_LINE	ISNUMERIC	XACT_STATE
ERROR_MESSAGE	NEWID	
ERROR_NUMBER	NULLIF	

Base de datos de SQL Azure **no admite** las siguientes funciones del sistema:

fn_helpcollations	fn_indexinfo	ORIGINAL_LOGIN
fn_serversharedrives	HOST_ID	SYSTEM_USER
fn_virtualfilestats	HOST_NAME	sys.dm_db_index_physical_stats
fn_virtualservernodes	IDENTITY	

## Funciones estadísticas del sistema

Las funciones estadísticas del sistema devuelven información estadística sobre el sistema. Base de datos de SQL Azure **no admite** las siguientes funciones del sistema:

@@CONNECTIONS	@@PACKET_ERRORS	@@TOTAL_ERRORS
@@CPU_BUSY	@@PACK_RECEIVED	@@TOTAL_READ
@@IDLE	@@PACK_SENT	@@TOTAL_WRITE
@@IO_BUSY	@@TIMETICKS	fn_virtualfilestats

## Funciones de texto y de imagen

Las funciones de texto y de imagen realizan una operación sobre un valor o columna de entrada de texto o de imagen y devuelven información sobre dicho valor. En esta categoría, Base de datos de SQL Azure solo admite la función PATINDEX.

Tenga en cuenta que Base de datos de SQL Azure **no admite** las siguientes funciones de texto y de imagen: TEXTPTR, TEXTVALID.

## Funciones escalares de ODBC

En la tabla siguiente se enumeran las categorías de funciones escalares de ODBC y también se describe la compatibilidad proporcionada por Base de datos de SQL Azure.

Categoría de función escalar de ODBC	Compatibilidad con Base de datos de SQL Azure
Funciones de cadenas de ODBC	Sí
Funciones numéricas de ODBC	Sí
Funciones de hora, fecha e intervalo de ODBC	Sí

### Funciones de cadenas de ODBC

BIT_LENGTH( string_exp ) (ODBC 3.0)
CONCAT( string_exp1,string_exp2) (ODBC 1.0)
OCTET_LENGTH( string_exp ) (ODBC 3.0)

### Funciones numéricas de ODBC

TRUNCATE( numeric_exp, integer_exp) (ODBC 2.0)
--

### Funciones de hora, fecha e intervalo de ODBC

CURRENT_DATE( ) (ODBC 3.0)
CURRENT_TIME[( time-precision )] (ODBC 3.0)
DAYNAME( date_exp ) (ODBC 2.0)
DAYOFMONTH( date_exp ) (ODBC 1.0)
DAYOFWEEK( date_exp ) (ODBC 1.0)
HOUR( time_exp ) (ODBC 1.0)
MINUTE( time_exp ) (ODBC 1.0)
MONTHNAME( date_exp ) (ODBC 2.0)
QUARTER( date_exp ) (ODBC 1.0)
WEEK( date_exp ) (ODBC 1.0)

## OPERADORES EN BASE DE DATOS DE SQL AZURE

Un operador es un símbolo que especifica una acción que se realiza en una o más expresiones. En la tabla siguiente se enumeran las categorías de operadores que Base de datos de SQL Azure admite.

Categoría de operador	Compatibilidad con Base de datos de SQL Azure
Operadores aritméticos	Sí
Operadores compuestos	Sí
Operadores bit a bit	Sí
Operadores lógicos	Sí
Operadores de comparación	Sí
Operadores unarios	Sí
Otros operadores misceláneos (operador de concatenación de cadenas y operador de asignación)	Sí
Operador de resolución de ámbito	No

### Operadores aritméticos

Los operadores aritméticos realizan operaciones matemáticas en dos expresiones de uno o más tipos de datos numéricos. En la tabla siguiente se enumeran los operadores aritméticos que Base de datos de SQL Azure admite:

Operador	Significado
+ (sumar)	Suma
- (restar)	Resta
* (multiplicar)	Multiplicación
/ (dividir)	División
% (Módulo)	Devuelve el resto entero de una división. Por ejemplo, $12 \% 5 = 2$ porque el resto de 12 dividido entre 5 es 2.

### Operadores compuestos

Los operadores compuestos ejecutan una operación y establecen un valor original en el resultado de dicha operación. Por ejemplo, si una variable @x es igual a 35, @x += 2 toma el valor original de @x, suma 2 y establece @x en el nuevo valor (37).

En la tabla siguiente se muestran los operadores compuestos que Base de datos de SQL Azure admite:

Operador	Más información	Acción
+=	+= (sumar igual)	Suma la cantidad especificada al valor original y establece este en el resultado de la operación.
-=	-= (restar igual)	Resta la cantidad especificada del valor original y establece este en el resultado de la operación.
*=	*= (multiplicar igual a)	Multiplica por el valor especificado y establece el valor original en el resultado de la operación.
/=	/= (dividir igual)	Divide por el valor especificado y establece el valor original en el resultado de la operación.
%=	%= (módulo igual)	Divide por el valor especificado y establece el valor original en el módulo.
&=	&= (AND bit a bit igual)	Realiza una operación AND bit a bit y establece el valor original en el resultado de la operación.
^=	^= (OR exclusivo bit a bit igual)	Realiza una operación OR exclusiva bit a bit y establece el valor original en el resultado de la operación.
=	= (OR bit a bit igual)	Realiza una operación OR bit a bit y establece el valor original en el resultado de la operación.

## Operadores bit a bit

Los operadores bit a bit realizan manipulaciones de bits entre dos expresiones de cualquiera de los tipos de datos de la categoría de tipo de datos entero. En la tabla siguiente se muestran los operadores bit a bit que Base de datos de SQL Azure admite:

Operador	Significado
& (AND bit a bit)	Operador AND bit a bit (dos operandos).
(OR bit a bit)	Operador OR bit a bit (dos operandos).
^ (OR exclusivo bit a bit)	Operador OR exclusivo bit a bit (dos operandos).

## Operadores lógicos

Los operadores lógicos comprueban la veracidad de alguna condición. Como los operadores de comparación, devuelven el tipo de datos **Boolean** con el valor TRUE, FALSE o UNKNOWN. En la tabla siguiente se enumeran los operadores lógicos que Base de datos de SQL Azure admite:

Operador	Significado
ALL	TRUE si todo el conjunto de comparaciones es TRUE.
AND	TRUE si ambas expresiones <b>Boolean</b> son TRUE.
ANY	TRUE si cualquier miembro del conjunto de comparaciones es TRUE.
BETWEEN	TRUE si el operando está dentro de un intervalo.
EXISTS	TRUE si una subconsulta contiene cualquiera de las filas.
IN	TRUE si el operando es igual a uno de la lista de expresiones.
LIKE	TRUE si el operando coincide con un patrón.
NOT	Invierte el valor de cualquier otro operador <b>Boolean</b> .
OR	TRUE si cualquiera de las expresiones <b>Boolean</b> es TRUE.
SOME	TRUE si alguna de las comparaciones de un conjunto es TRUE.

## Operadores de comparación

Los operadores de comparación comprueban si dos expresiones son iguales. En la siguiente tabla se muestran los operadores de comparación que Base de datos de SQL Azure admite:

Operador	Significado
= (Igual a)	Igual a
> (Mayor que)	Mayor que
< (Menor que)	Menor que
>= (Mayor o igual que)	Mayor o igual que
<= (Menor o igual que)	Menor o igual que
<> (No es igual a)	No es igual a
!= (No es igual a)	No es igual a (no es del estándar ISO)
!< (No menor que)	No es menor que (no es del estándar ISO)
!> (No mayor que)	No es mayor que (no es del estándar ISO)

## Operadores unarios

Los operadores unarios realizan una operación sobre una única expresión de cualquiera de los tipos de datos de la categoría del tipo de datos numérico. En la tabla siguiente se muestran los operadores unarios que Base de datos de SQL Azure admite:

Operador	Significado
+ (Positivo)	El valor numérico es positivo.
- (Negativo)	El valor numérico es negativo.
~ (NOT bit a bit)	Realiza una operación lógica NOT bit a bit para la expresión, tomando un bit cada vez. Si la expresión tiene un valor de 0, los bits del conjunto de resultados se establecen en 1; de lo contrario, el bit del resultado se borra a un valor de 0. Es decir, los unos se cambian a ceros y los ceros se cambian a unos.

## Otros operadores misceláneos

En la tabla siguiente tabla se enumeran los operadores que no encajan en otras categorías y que son compatibles con Base de datos de SQL Azure.

Operador	Significado
+ (Operador de concatenación de cadenas)	El signo de suma (+) es el operador que permite concatenar cadenas.
= (Operador de asignación)	El signo igual (=) es el único operador de asignación de Transact-SQL.

## TABLAS DEL SISTEMA DE BASE DE DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure no admite ninguna de las tablas del sistema proporcionadas por SQL Server. Para obtener más información sobre las tablas del sistema en SQL Server, vea Tablas del sistema (Transact-SQL) en Libros en pantalla de SQL Server.

En la tabla siguiente se proporciona una lista de las tablas del sistema que Base de datos de SQL Azure **no admite**.

Tablas de copias de seguridad y restauración	Tablas de replicación
Tablas de captura de datos modificados	Tablas del Agente SQL Server
Tablas de planes de mantenimiento de bases de datos	sys.sysoledbusers
Tablas de Integration Services	systranschemas
Tablas de trasvase de registros	

## VISTAS DEL SISTEMA DE BASE DE DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure ofrece compatibilidad con algunas vistas del sistema integradas de SQL Server. En este tema se describe la compatibilidad con las vistas del sistema.

En la tabla siguiente se muestran las categorías de vistas del sistema y también se describe si Base de datos de SQL Azure ofrece compatibilidad con cada categoría.

Categoría	Compatibilidad con Base de datos de SQL Azure
Vistas de bases de datos de SQL Azure	Sí
Vistas de catálogo	Parcial
Vistas de compatibilidad	Parcial
Vistas del correo electrónico de base de datos	No
Vistas del recopilador de datos	No
Vistas de la administración basada en directivas	No
Vistas de administración dinámica	Parcial
Vistas del esquema de información	Sí
Vistas de replicación	No

### Vistas de bases de datos de SQL Azure

Varias vistas son específicas de Base de datos de SQL Azure y no se admiten en SQL Server local. Estas vistas se muestran en la tabla siguiente:

sys.bandwidth_usage (Base de datos de SQL Azure)	sys.dm_database_copies (Base de datos de SQL Azure)	sys.firewall_rules (Base de datos de SQL Azure)
sys.database_usage (Base de datos de SQL Azure)	sys.dm_db_objects_impacted_on_version_change (base de datos de SQL Azure)	

## Vistas de catálogo

Las vistas de catálogo devuelven información utilizada por el motor de base de datos de SQL Server subyacente. En la tabla siguiente se enumeran las vistas de catálogo compatibles en Base de datos de SQL Azure.

sys.all_columns	sys.events	sys.stats_columns
sys.all_objects	sys.foreign_key_columns	sys.synonyms
sys.all_parameters	sys.foreign_keys	sys.system_columns
sys.all_sql_modules	sys.identity_columns	sys.system_objects
sys.all_views	sys.index_columns	sys.system_parameters
sys.check_constraints	sys.indexes	sys.system_sql_modules
sys.columns	sys.key_constraints	sys.system_views
sys.computed_columns	sys.objects	sys.table_types
sys.database_permissions	sys.parameters	sys.tables
sys.database_principals	sys.plan_guides	sys.trigger_event_types
sys.database_role_members	sys.procedures	sys.trigger_events
sys.databases (Base de datos de SQL Azure)	sys.schemas	sys.triggers
sys.default_constraints	sys.sql_logins	sys.types
sys.event_notification_event_types	sys.sql_modules	sys.views
sys.event_notifications	sys.stats	sys.xml_schema_collections

Base de datos de SQL Azure **no admite** las siguientes vistas de catálogo del sistema.

sys.allocation_units	sys.fulltext_stopwords	sys.server_role_members
sys.assemblies	sys.fulltext_system_stopwords	sys.server_sql_modules
sys.assembly_files	sys.function_order_columns	sys.server_trigger_events
sys.assembly_modules	sys.http_endpoints	sys.server_triggers
sys.assembly_references	sys.internal_tables	sys.servers
sys.assembly_types	sys.key_encryptions	sys.service_broker_endpoints
sys.asymmetric_keys	sys.linked_logins	sys.service_contract_message_usages
sys.backup_devices	sys.master_files	sys.service_contract_usages
sys.certificates	sys.master_key_passwords	sys.service_contracts
sys.change_tracking_database	sys.message_type_xml_schema_collection_usages	sys.service_message_types
sys.change_tracking_tables	sys.messages	sys.service_queue_usages
sys.column_type_usages	sys.module_assembly_usages	sys.service_queues
sys.column_xml_schema	sys.numbered_procedure	sys.services

collection_usages	parameters	
sys.configurations	sys.numbered_procedures	sys.soap_endpoints
sys.conversation_endpoints	sys.openkeys	sys.spatial_index_tessellations
sys.conversation_groups	sys.parameter_xml_schema_collection_usages	sys.spatial_indexes
sys.conversation_priorities	sys.paramter_type_usages	sys.symmetric_keys
sys.credentials	sys.partition_functions	sys.system_components_surface_area_configuration
sys.crypt_properties	sys.partition_parameters	sys.tcp_endpoints
sys.cryptographic_providers	sys.partition_range_values	sys.trace_categories
sys.data_spaces	sys.partition_schemes	sys.trace_columns
sys.database_audit_specification_details	sys.partitions	sys.trace_event_bindings
sys.database_audit_specifications	sys.remote_logins	sys.trace_events
sys.database_files	sys.remote_service_bindings	sys.trace_subclass_values
sys.database_mirroring	sys.resource_governor_configuration	sys.traces
sys.database_mirroring_endpoints	sys.resource_governor_resource_pools	sys.transmission_queue
sys.database_mirroring_witnesses	sys.resource_governor_workload_groups	sys.type_assembly_usages
sys.database_recovery_status	sys.routes	sys.via_endpoints
sys.destination_data_spaces	sys.securable_classes	sys.xml_indexes
sys.endpoint_webmethods	sys.server_assembly_modules	sys.xml_schema_attributes
sys.endpoints	sys.server_audit_specifications_details	sys.xml_schema_collections
sys.extended_properties	sys.server_audits	sys.xml_schema_component_placements
sys.extended_procedures	sys.server_event_notifications	sys.xml_schema_components
sys.filegroups	sys.server_event_session_fields	sys.xml_schema_elements
sys.fulltext_catalogs	sys.server_event_session_targets	sys.xml_schema_facets
sys.fulltext_document_types	sys.server_event_sessions	sys.xml_schema_model_groups
sys.fulltext_index_catalog_usages	sys.server_event_sessions_actions	sys.xml_schema_namespaces
sys.fulltext_index_columns	sys.server_event_sessions_events	sys.xml_schema_types
sys.fulltext_index_fragments	sys.server_events	sys.xml_schema_wildcard_namespaces

sys.fulltext_indexes	sys.server_file_audits	sys.xml_schema_wildcards
sys.fulltext_languages	sys.server_permissions	
sys.fulltext_stoplists	sys.server_principals	

## Vistas de compatibilidad

Muchas de las tablas del sistema de versiones anteriores de SQL Server se implementan ahora como un conjunto de vistas. Se conocen como vistas de compatibilidad y solo se proporcionan por compatibilidad con versiones anteriores.

Base de datos de SQL Azure solo admite sys.syscharsets, sys.syslanguages y sys.systypes en esta categoría.

## Vistas de administración dinámica

Las vistas de administración dinámica devuelven información sobre el estado del servidor que se puede utilizar para supervisar el estado de una instancia del servidor, diagnosticar problemas y optimizar el rendimiento. En esta versión, Base de datos de SQL Azure proporciona compatibilidad parcial para solo tres categorías de vistas de administración dinámica: Vistas de administración dinámica relacionadas con la base de datos, Vistas de administración dinámica relacionadas con ejecuciones y Vistas de administración dinámica relacionadas con transacciones.

En Base de datos de SQL Azure, para consultar una vista de administración dinámica se necesita el permiso **VIEW DATABASE STATE**.

### Vistas de administración dinámica relacionadas con la base de datos

En la tabla siguiente se enumeran las vistas de administración dinámica relacionadas con la base de datos compatibles en Base de datos de SQL Azure.

sys.dm_db_partition_stats
---------------------------

Base de datos de SQL Azure no admite las siguientes vistas de administración dinámica relacionadas con la base de datos.

sys.dm_db_file_space_usage	sys.dm_db_session_space_usage
sys.dm_db_persisted_sku_features	sys.dm_db_task_space_usage
sys.dm_db_objects_disabled_on_compatibility_level_change	

## Vistas de administración dinámica relacionadas con ejecuciones

En la tabla siguiente se enumeran las vistas de administración dinámica relacionadas con ejecuciones compatibles en Base de datos de SQL Azure.

sys.dm_exec_connections (Base de datos de SQL Azure)
sys.dm_exec_query_plan
sys.dm_exec_query_stats
sys.dm_exec_requests (Base de datos de SQL Azure)
sys.dm_exec_sessions (Base de datos de SQL Azure)
sys.dm_exec_sql_text
sys.dm_exec_text_query_plan

Base de datos de SQL Azure **no admite** las siguientes vistas de administración dinámica relacionadas con ejecuciones.

sys.dm_exec_background_job_queue	sys.dm_exec_plan_attributes
sys.dm_exec_background_job_queue	sys.dm_exec_procedure_stats
sys.dm_exec_background_job_queue_stats	sys.dm_exec_trigger_stats
sys.dm_exec_query_resource_semaphores	sys.dm_exec_query_memory_grants
sys.dm_exec_cached_plans	sys.dm_exec_query_optimizer_info
sys.dm_exec_cached_plan_dependent_objects	sys.dm_exec_xml_handles
sys.dm_exec_cursors	

## Vistas de administración dinámica relacionadas con transacciones

En la tabla siguiente se enumeran las vistas de administración dinámica relacionadas con transacciones compatibles en Base de datos de SQL Azure.

sys.dm_tran_active_transactions (Base de datos de SQL Azure)
sys.dm_tran_database_transactions (Base de datos de SQL Azure)
sys.dm_tran_locks
sys.dm_tran_session_transactions

Base de datos de SQL Azure **no admite** las siguientes vistas de administración dinámica relacionadas con transacciones.

sys.dm_tran_active_snapshot_database_transactions	sys.dm_tran_transactions_snapshot
sys.dm_tran_current_snapshot	sys.dm_tran_version_store
sys.dm_tran_current_transaction	sys.dm_tran_active_snapshot_database_transactions
sys.dm_tran_top_version_generators	

## Vistas del esquema de información

Las vistas del esquema de información proporcionan información sobre metadatos. En la tabla siguiente se enumeran las vistas del esquema de información compatibles en Base de datos de SQL Azure.

CHECK_CONSTRAINTS	DOMAINS	TABLE_CONSTRAINTS
COLUMN_DOMAIN_USAGE	KEY_COLUMN_USAGE	TABLE_PRIVILEGES
COLUMN_PRIVILEGES	PARAMETERS	TABLES
COLUMNS	REFERENTIAL_CONSTRAINTS	VIEW_COLUMN_USAGE
CONSTRAINT_COLUMN_USAGE	ROUTINE_COLUMNS	VIEW_TABLE_USAGE
CONSTRAINT_TABLE_USAGE	ROUTINES	VIEWS
DOMAIN_CONSTRAINTS	SCHEMATA	

# TRANSACT-SQL EN SQL AZURE. INSTRUCCIONES SQL SERVER ADMITIDAS Y PROCEDIMIENTOS

## INSTRUCCIONES TRANSACT-SQL ADMITIDAS EN BASE DE DATOS DE SQL AZURE

En la tabla siguiente se proporciona una lista de las instrucciones de Transact-SQL admitidas por Microsoft Base de datos de SQL Azure. Estas instrucciones de Transact-SQL proporcionan la misma funcionalidad que sus instrucciones correspondientes de Transact-SQL en SQL Server 2008.

ALTER ROLE	DECLARE CURSOR	OPEN
ALTER SCHEMA	DELETE	Cláusula OPTION
ALTER VIEW	DENY permisos de objeto	Cláusula ORDER BY
APPLOCK_MODE	DENY permisos de esquema	Cláusula OUTPUT
APPLOCK_TEST	DROP LOGIN	Cláusula OVER
BEGIN_TRANSACTION	DROP PROCEDURE	PRINT
BEGIN...END	DROP ROLE	RAISERROR
BINARY_CHECKSUM	DROP SCHEMA	RETURN

BREAK	DROP STATISTICS	REVERT
CAST y CONVERT	DROP SYNONYM	REVOKE permisos de objeto
CATCH (TRY...CATCH)	DROP TYPE	REVOKE permisos de esquema
CEILING	DROP USER	ROLLBACK TRANSACTION
CHECKSUM	DROP VIEW	ROLLBACK WORK
CLOSE	END (BEGIN...END)	SAVE TRANSACTION
COALESCE	EXCEPT e INTERSECT	SELECT @local_variable
COLLATE	FETCH	Cláusula SELECT
COLUMNPROPERTY	Cláusula FOR (XML y BROWSE)	SET @local_variable
COMMIT TRANSACTION	FROM	SWITCHOFFSET
COMMIT WORK	GO	TERTIARY_WEIGHTS
COMPUTE	GOTO	TODATETIMEOFFSET
CONTEXT_INFO	GRANT permisos de objeto	TOP
CONTINUE	GRANT permisos de esquema	TRIGGER_NESTLEVEL
CONVERT	GROUP BY	TRUNCATE TABLE
CREATE ROLE	GROUPING_ID	TRY...CATCH
CREATE SCHEMA	HashBytes	UNION
CREATE STATISTICS	HAVING	UPDATE
CREATE VIEW	Sugerencias, como sugerencias de consulta, sugerencias de combinación y sugerencias de tabla	UPDATE STATISTICS
CRYPT_GEN_RANDOM	IDENTITY (propiedad)	USER
CURRENT_REQUEST_ID	IF...ELSE	SWITCHOFFSET
CURSOR_STATUS	INSERT BULK	WAITFOR
DBCC SHOW_STATISTICS	IS [NOT] NULL	WHERE
DEALLOCATE	MERGE	WHILE
DECLARE @local_variable	MIN_ACTIVE_ROWVERSION	WITH common_table_expression

## Compatibilidad con sugerencias

Las sugerencias son opciones o estrategias especificadas para su cumplimiento por el procesador de consultas subyacente en instrucciones **SELECT**, **INSERT**, **UPDATE**, **DELETE** o **MERGE**. Base de datos de SQL Azure admite todas las **Query**, **Table** y **Join Hints**

proporcionadas por SQL Server salvo algunas excepciones. Al ejecutar consultas, Base de datos de SQL Azure siempre establece automáticamente el grado máximo de paralelismo en 1 y omite la sugerencia **MAXDOP** aunque se utilice. Además, SQL Azure no admite la sugerencia de tabla **PAGLOCK**.

## Compatibilidad con instrucciones SET

El lenguaje de programación Transact-SQL ofrece varias instrucciones SET que cambian el tratamiento de información específica por parte de la sesión actual.

En la tabla siguiente se proporciona una lista de las instrucciones SET admitidas por Base de datos de SQL Azure. Tenga en cuenta que estas instrucciones SET proporcionan la misma funcionalidad que sus instrucciones SET correspondientes en SQL Server 2008.

Categoría de instrucción SET	Instrucción
Instrucciones de fecha y hora	SET DATEFIRST SET DATEFORMAT
Instrucciones de bloqueo	SET DEADLOCK_PRIORITY SET LOCK_TIMEOUT
Otras instrucciones	SET CURSOR_CLOSE_ON_COMMIT SET FIPS_FLAGGER SET IDENTITY_INSERT SET LANGUAGE SET QUOTED_IDENTIFIER
Instrucciones de ejecución de consultas	SET ARITHABORT SET ARITHIGNORE SET FMTONLY SET NOCOUNT SET NOEXEC SET NUMERIC_ROUNDABORT SET PARSEONLY SET ROWCOUNT SET TEXTSIZE
Instrucciones de configuración de ISO	SET ANSI_NULL_DFLT_OFF SET ANSI_NULL_DFLT_ON SET ANSI_PADDING_ON SET ANSI_WARNINGS
Instrucciones de estadísticas	SET FORCEPLAN SET SHOWPLAN_ALL

	SET SHOWPLAN_TEXT SET SHOWPLAN_XML SET STATISTICS IO SET STATISTICS PROFILE SET STATISTICS TIME SET STATISTICS XML
Instrucciones de transacciones	SET IMPLICIT_TRANSACTIONS SET TRANSACTION ISOLATION LEVEL SET XACT_ABORT

## INSTRUCCIONES TRANSACT-SQL NO ADMITIDAS EN BASE DE DATOS DE SQL AZURE

Microsoft Base de datos de SQL Azure no admite todas las instrucciones de Transact-SQL que se ofrecen en SQL Server 2008. En la tabla siguiente se muestra una lista de las instrucciones de Transact-SQL que Base de datos de SQL Azure **no admite**.

\$PARTITION	fn_syscollector_get_execution_stats
ADD SIGNATURE	fn_trace_geteventinfo
ALTER RESOURCE GOVERNOR	fn_trace_getinfo
ALTER SERVICE MASTER KEY	fn_trace_gettable
ASSEMBLYPROPERTY	fn_virtualfilestats
ASYMKEY_ID	fn_virtualservernodes
ASYMKEYPROPERTY	FREETEXT
BACKUP	FREETEXTTABLE
BACKUP CERTIFICATE	FULLTEXTCATALOGPROPERTY
BACKUP MASTER KEY	FULLTEXTSERVICEPROPERTY
BACKUP SERVICE MASTER KEY	GET CONVERSATION GROUP
BEGIN CONVERSATION TIMER	GET_FILESTREAM_TRANSACTION_CONTEXT
BEGIN DIALOG CONVERSATION	GET_TRANSMISSION_STATUS
BEGIN DISTRIBUTED TRANSACTION	GRANT/REVOKE/DENY ALL
BULK INSERT	GRANT/REVOKE/DENY ALL PRIVILEGES
cdc.fn_cdc_get_all_changes_<capture_instance>	GRANT/REVOKE/DENY permisos de ensamblado
cdc.fn_cdc_get_net_changes_<capture_instance>	GRANT/REVOKE/DENY permisos de clave asimétrica

CERT_ID	GRANT/REVOKE/DENY permisos de certificado
CERTPROPERTY	GRANT/REVOKE/DENY permisos de extremo
CHANGE_TRACKING_CURRENT_VERSION	GRANT/REVOKE/DENY permisos del catálogo de texto completo
CHANGE_TRACKING_IS_COLUMN_IN_MASK	GRANT/REVOKE/DENY permisos de servidor
CHANGE_TRACKING_MIN_VALID_VERSION	GRANT/REVOKE/DENY permisos de entidad de seguridad de servidor
CHANGETABLE	GRANT/REVOKE/DENY permisos de Service Broker
CHECKPOINT	GRANT/REVOKE/DENY permisos de clave simétrica
CLOSE MASTER KEY	GRANT/REVOKE/DENY permisos de objeto del sistema
CLOSE SYMMETRIC KEY	GRANT/REVOKE/DENY permisos de colección de esquemas XML
CONNECTIONPROPERTY	HAS_DBACCESS
CONTAINS	INDEXKEY_PROPERTY
CONTAINSTABLE	IS_OBJECTSIGNED
CREATE/DROP AGGREGATE	KEY_GUID
CREATE/ALTER/DROP APPLICATION ROLE	KEY_ID
CREATE/ALTER/DROP ASSEMBLY	KEY_NAME
CREATE/ALTER/DROP ASYMMETRIC KEY	KILL
CREATE/ALTER/DROP BROKER PRIORITY	KILL QUERY NOTIFICATION SUBSCRIPTION
CREATE/ALTER/DROP CERTIFICATE	KILL STATS JOB
CREATE/ALTER/DROP CONTRACT	LOGINPROPERTY
CREATE/ALTER/DROP CREDENTIAL	MOVE CONVERSATION
CREATE/ALTER/DROP CRYPTOGRAPHIC PROVIDER	NEWSEQUENTIALID
CREATE/ALTER/DROP DATABASE AUDIT SPECIFICATION	OBJECT_DEFINITION
CREATE/ALTER/DROP DATABASE ENCRYPTION KEY	OPEN MASTER KEY
CREATE/ALTER/DROP DEFAULT	OPEN SYMMETRIC KEY
CREATE/ALTER/DROP ENDPOINT	OPENDATASOURCE
CREATE/ALTER/DROP EVENT NOTIFICATION	OPENQUERY

CREATE/ALTER/DROP EVENT SESSION	OPENROWSET
CREATE/ALTER/DROP FULLTEXT CATALOG	OPENXML
CREATE/ALTER/DROP FULLTEXT INDEX	ORIGINAL_DB_NAME
CREATE/ALTER/DROP FULLTEXT STOPLIST	ORIGINAL_LOGIN
CREATE/ALTER/DROP MASTER KEY	PathName
CREATE/ALTER/DROP MESSAGE TYPE	PUBLISHINGSERVERNAME
CREATE/ALTER/DROP PARTITION FUNCTION	READTEXT
CREATE/ALTER/DROP PARTITION SCHEME	RECEIVE
CREATE/ALTER/DROP QUEUE	RECONFIGURE
CREATE/ALTER/DROP REMOTE SERVICE BINDING	RESTORE
CREATE/ALTER/DROP RESOURCE POOL	RESTORE FILELISTONLY
CREATE/ALTER/DROP ROUTE	RESTORE HEADERONLY
CREATE/ALTER/DROP SERVER AUDIT	RESTORE LABELONLY
CREATE/ALTER/DROP SERVER AUDIT SPECIFICATION	RESTORE MASTER KEY
CREATE/ALTER/DROP SERVICE	RESTORE REWINDONLY
CREATE/ALTER/DROP SYMMETRIC KEY	RESTORE SERVICE MASTER KEY
CREATE/ALTER/DROP WORKLOAD GROUP	RESTORE VERIFYONLY
CREATE/ALTER/DROP XML SCHEMA COLLECTION	SELECT INTO Clause
CREATE/DROP RULE	SEND
CREATE/DROP XML INDEX	SET ANSI_DEFAULTS
DATABASEPROPERTY	SET ANSI_NULLS
DBCC CHECKALLOC	SET CONCAT_NULL_YIELDS_NULL
DBCC CHECKCATALOG	SET ANSI_PADDING_OFF
DBCC CHECKCONSTRAINTS	SET CONTEXT_INFO
DBCC CHECKDB	SET DISABLE_DEF_CNST_CHK
DBCC CHECKFILEGROUP	SET OFFSETS
DBCC CHECKIDENT	SET REMOTE_PROC_TRANSACTIONS
DBCC CHECKTABLE	SETUSER
DBCC CLEANABLE	SHUTDOWN
DBCC DBREINDEX	SIGNBYASYMKEY
DBCC dllname	SIGNBYCERT
DBCC DROPCLEANBUFFERS	snapshots.fn_trace_getdata
DBCC FREEPROCCACHE	SYMKEYPROPERTY
DBCC FREESESSIONCACHE	sys.fn_all_changes_capture_instance

	>
DBCC FREESYSTEMCACHE	sys.fn_builtin_permissions
DBCC HELP	sys.fn_cdc_decrement_lsn
DBCC INDEXDEFRAG	sys.fn_cdc_get_column_ordinal
DBCC INPUTBUFFER	sys.fn_cdc_get_max_lsn
DBCC OPENTRAN	sys.fn_cdc_get_min_lsn
DBCC OUTPUTBUFFER	sys.fn_cdc_has_column_changed
DBCC PROCCACHE	sys.fn_cdc_increment_lsn
DBCC SHOWCONTIG	sys.fn_cdc_is_bit_set
DBCC SHRINKDATABASE	sys.fn_cdc_map_lsn_to_time
DBCC SHRINKFILE	sys.fn_cdc_map_time_to_lsn
DBCC SQLPERF	sys.fn_check_object_signatures
DBCC TRACEOFF	sys.fn_net_changes_<capture_instance>
DBCC TRACEON	sys.fn_translate_permissions
DBCC TRACESTATUS	sys.fn_validate_plan_guide
DBCC UPDATEUSAGE	sys.fn_xe_file_target_read_file
DBCC USEROPTIONS	sys.login_token
DECRYPTBYASYMKEY	sys.user_token
DECRYPTBYCERT	sys.fn_validate_plan_guide
DECRYPTBYKEY	sys.fn_xe_file_target_read_file
DECRYPTBYKEYAUTOASYMKEY	sys.login_token
DECRYPTBYKEYAUTOCERT	sys.numbered_procedure_parameters
DECRYPTBYPASSPHRASE	sys.numbered_procedures
ENCRYPTBYASYMKEY	sys.user_token
ENCRYPTBYCERT	UPDATETEXT
ENCRYPTBYKEY	VERIFYSIGNEDBYASYMKEY
ENCRYPTBYPASSPHRASE	VERIFYSIGNEDBYCERT
END CONVERSATION	WITH CHANGE_TRACKING_CONTEXT
EVENTDATA	WITH XMLNAMESPACES
fn_get_audit_file	WRITETEXT
fn_get_sql	XML_SCHEMA_NAMESPACE
fn_syscollector_get_execution_details	

## PROCEDIMIENTOS ALMACENADOS DEL SISTEMA EN BASE DE DATOS DE SQL AZURE

Algunas actividades administrativas e informativas se pueden realizar mediante procedimientos almacenados del sistema. Microsoft Base de datos de SQL Azure no admite todos los procedimientos almacenados del sistema proporcionados por SQL Server. En este tema se describe la compatibilidad con los procedimientos almacenados del sistema.

En la tabla siguiente se muestran las categorías de procedimientos almacenados del sistema y también se describe si Base de datos de SQL Azure ofrece o no compatibilidad con cada categoría:

Categoría de procedimientos almacenados del sistema	Compatibilidad con Base de datos de SQL Azure
Procedimientos almacenados de base de datos SQL Azure	Sí
Procedimientos almacenados de Active Directory	No
Procedimientos almacenados del catálogo	Parcial
Procedimientos almacenados de captura de datos modificados	No
Procedimientos almacenados de cursor	No
Procedimientos almacenados del motor de base de datos	Parcial
Procedimientos almacenados de Correo electrónico de base de datos y SQL Mail	No
Procedimientos almacenados de planes de mantenimiento de bases de datos	No
Procedimientos almacenados de consultas distribuidas	No
Procedimientos almacenados de búsqueda de texto completo	No
Procedimientos almacenados de trasvase de registros	No
Procedimientos almacenados de automatización	No
Procedimientos almacenados de administración basada en directivas	No
Procedimientos almacenados de replicación	No

Procedimientos almacenados de seguridad	Parcial
Procedimientos almacenados de SQL Server Profiler	No
Procedimientos almacenados del Agente SQL Server	No
Procedimientos almacenados de XML	No
Procedimientos almacenados extendidos generales	No

## Procedimientos almacenados de base de datos SQL Azure

Varios procedimientos almacenados son específicos de Base de datos de SQL Azure y no se admiten en SQL Server local. Estos procedimientos almacenados se muestran en la tabla siguiente:

sp_delete_firewall_rule (Base de datos de SQL Azure)	sp_set_firewall_rule (Base de datos de SQL Azure)
--	---

## Procedimientos almacenados del catálogo

Base de datos de SQL Azure admite los procedimientos almacenados del sistema de catálogo de la tabla siguiente tal cual. Estos procedimientos almacenados implementan funciones del diccionario de datos ODBC y aíslan las aplicaciones ODBC de los cambios en las tablas del sistema subyacentes.

sp_column_privileges	sp_sproc_columns
sp_columns	sp_statistics
sp_fkeys	sp_stored_procedures
sp_pkeys	sp_table_privileges
sp_special_columns	sp_tables

Base de datos de SQL Azure **no admite** los procedimientos almacenados del sistema sp\_server\_info y sp\_databases.

## Procedimientos almacenados del motor de base de datos

Los procedimientos almacenados del motor de base de datos se emplean para el mantenimiento general de una instancia de SQL Server. En la tabla siguiente se muestran los procedimientos almacenados del motor de base de datos que Base de datos de SQL Azure admite:

sp_autostats	sp_helplanguage	sp_releaseapplock
sp_createstats	sp_helpsort	sp_rename
sp_datatype_info	sp_helpstats	sp_settriggerorder (Base de datos de SQL Azure)
sp_executesql	sp_helptext	sp_tableoption (Base de datos de SQL Azure)
sp_getapplock	sp_helptrigger	sp_updatestats
sp_help	sp_recompile	sp_validname
sp_helpconstraint	sp_refreshsqlmodule (Base de datos de SQL Azure)	
sp_helpindex	sp_refreshview	

En la tabla siguiente se muestran los procedimientos almacenados del motor de base de datos que Base de datos de SQL Azure **no admite**:

sp_add_data_file_recover_suspect_db	sp_cycle_errorlog	sp_helpdb
sp_add_log_file_recover_suspect_db	sp_dbcmptlevel	sp_helpdevice
sp_addextendedproc	sp_dbmmonitoraddmonitoring	sp_helpextendedproc
sp_addextendedproperty	sp_dbmmonitorchangealert	sp_helpfile
sp_addmessage	sp_dbmmonitorchangemonitoring	sp_helpfilegroup
sp_addtype	sp_dbmmonitordropalert	sp_helpserver
sp_addumpdevice	sp_dbmmonitordropmonitoring	sp_indexoption
sp_altermessage	sp_dbmmonitorhelpalert	sp_invalidate_textptr
sp_attach_db	sp_dbmmonitorhelpmonitoring	sp_lock
sp_attach_single_file_db	sp_dbmmonitorresults	sp_monitor
sp_bindefault	sp_dboption	sp_procoption
sp_bindrule	sp_dbremove	sp_renamedb
sp_bindsession	sp_delete_backuphistory	sp_resetstatus

sp_certify_removable	sp_depends	sp_serveroption
sp_clean_db_file_free_space	sp_detach_db	sp_setnetname
sp_clean_db_free_space	sp_dropdevice	sp_spaceused
sp_configure	sp_dropextendedproc	sp_unbinddefault
sp_control_plan_guide	sp_dropextendedproperty	sp_unbindrule
sp_create_plan_guide	sp_dropmessage	sp_updateextendedproperty
sp_create_plan_guide_from_handle	sp_droptype	sp_who
sp_create_removable	sp_getbindtoken	

## Procedimientos almacenados de seguridad

Los procedimientos almacenados del sistema de seguridad se usan para administrar la seguridad. Base de datos de SQL Azure admite los siguientes procedimientos almacenados del sistema de seguridad: sp\_addrolemember, sp\_droprolemember y sp\_helprole.

En la tabla siguiente se muestran los procedimientos almacenados de seguridad que Base de datos de SQL Azure **no admite**:

sp_addapprole	sp_dropalias	sp_helpprotect
sp_addlogin	sp_dropapprole	sp_helpsrvrole
sp_addremotelogin	sp_droplogin	sp_helpsrvrolemember
sp_addrole	sp_dropremotelogin	sp_helpuser
sp_addserver	sp_droprole	sp_MShasdbaccess
sp_addsrvrolemember	sp_dropserver	sp_password
sp_adduser	sp_dropsrvrolemember	sp_remoteoption
sp_approlepassword	sp_dropuser	sp_revokedbaccess
sp_change_users_login	sp_grantdbaccess	sp_revokelogin
sp_changedbowner	sp_grantlogin	sp_setapprole
sp_changeobjectowner	sp_helpdbfixedrole	sp_srvrolepermission
sp_control_dbmasterkey_password	sp_helplinkedsvlogin	sp_unsetapprole
sp_dbfixedrolepermission	sp_helplogins	sp_validatelogins
sp_defaultdb	sp_helpntgroup	sp_xp_cmdshell_proxy_account
sp_defaultlanguage	sp_helpremotelogin	
sp_denylogin	sp_helprolemember	

## REFERENCIA DE ERRORES Y EXCEPCIONES EN BASE DE DATOS DE SQL AZURE

A continuación, se proporciona información sobre los errores y excepciones que se pueden producir solo cuando se utiliza Microsoft Base de datos de SQL Azure.

### Errores generales

En la tabla siguiente se muestran todos los errores generales en orden ascendente.

Número de error	Gravedad	Descripción
40014	15	No se pueden usar varias bases de datos en la misma transacción.
40054	15	Las tablas sin un índice clúster no se admiten en esta versión de SQL Server. Cree un índice clúster e inténtelo de nuevo.
40133	16	Esta operación no se admite en esta versión de SQL Server.
40506	15	El SID especificado no es válido para esta versión de SQL Server.
40507	15	'%.*Is' no se puede invocar con parámetros en esta versión de SQL Server.
40508	15	La instrucción USE no se admite para cambiar de una base de datos a otra. Use una conexión nueva para conectarse a otra base de datos.
40512	16	La característica desusada '%Is' no se admite en esta versión de SQL Server.
40516	15	Los objetos temporales globales no se admiten en esta versión de SQL Server.
40520	15	La clase protegible '%S_MSG' no se admite en esta versión de SQL Server.
40521	15	La clase protegible '%S_MSG' no se admite en el ámbito de servidor en esta versión de SQL Server.
40522	15	El tipo de entidad de seguridad de base de datos '%.*Is' no se admite en esta versión de SQL Server.
40523	15	La creación de usuario implícita '%.*Is' no se admite en esta versión de SQL Server. Cree el usuario de forma explícita antes de usarlo.
40524	15	El tipo de datos '%.*Is' no se admite en esta versión de SQL Server.

40525	15	WITH '%.ls' no se admite en esta versión de SQL Server.
40526	15	El proveedor de conjuntos de filas '%.*ls' no se admite en esta versión de SQL Server.
40527	15	Los servidores vinculados no se admiten en esta versión de SQL Server.
40528	15	Los usuarios no se pueden asignar a certificados, claves asimétricas o inicios de sesión de Windows en esta versión de SQL Server.
40532	11	No se puede abrir el servidor "%.*ls" solicitado por el inicio de sesión. Error de inicio de sesión.
40553	20	La sesión ha terminado debido al uso excesivo de la memoria. Intente modificar la consulta para procesar menos filas.
40604	15	No puede utilizar CREATE/ALTER DATABASE porque superaría la cuota del servidor.
40606	15	No se admite adjuntar bases de datos en esta versión de SQL Server.
40607	15	Los inicios de sesión de Windows no se admiten en esta versión de SQL Server.
40611	15	Los servidores pueden tener 128 reglas de firewall definidas como máximo.
40615	14	No se puede abrir el servidor '{0}' solicitado por el inicio de sesión. El cliente con la dirección IP '{1}' no tiene permiso para obtener acceso al servidor. Para habilitar el acceso, use el portal de SQL Azure o ejecute sp_set_firewall_rule en la base de datos maestra para crear una regla de firewall para esta dirección IP o para este intervalo de direcciones. Se puede tardar hasta cinco minutos para que este cambio surta efecto.
40636	16	No se puede usar un nombre de base de datos reservado '%.*ls' en esta operación.

## Errores de copia de base de datos

En la tabla siguiente se incluyen los diversos errores que puede encontrarse al copiar una base de datos en SQL Azure.

Número de error	Gravedad	Descripción
40635	16	El cliente con la dirección IP '%.*s' está deshabilitado temporalmente.
40637	16	La creación de una copia de la base de datos está deshabilitada actualmente.
40561	16	Error al copiar la base de datos. No existe la base de datos de origen o de destino.
40562	16	Error al copiar la base de datos. La base de datos de origen se ha quitado.
40563	16	Error al copiar la base de datos. La base de datos de destino se ha quitado.
40564	16	No se pudo copiar la base de datos debido a un error interno. Quite la base de datos de destino e inténtelo de nuevo.
40565	16	Error al copiar la base de datos. No se permite más de una copia de base de datos simultánea para el mismo origen. Quite la base de datos de destino e inténtelo de nuevo más tarde.
40566	16	No se pudo copiar la base de datos debido a un error interno. Quite la base de datos de destino e inténtelo de nuevo.
40567	16	No se pudo copiar la base de datos debido a un error interno. Quite la base de datos de destino e inténtelo de nuevo.
40568	16	Error al copiar la base de datos. La base de datos de origen ha dejado de estar disponible. Quite la base de datos de destino e inténtelo de nuevo.
40569	16	Error al copiar la base de datos. La base de datos de destino ha dejado de estar disponible. Quite la base de datos de destino e inténtelo de nuevo.
40570	16	No se pudo copiar la base de datos debido a un error interno. Quite la base de datos de destino e inténtelo de nuevo más tarde.
40571	16	No se pudo copiar la base de datos debido a un error interno. Quite la base de datos de destino e inténtelo de nuevo más tarde.

## Errores de pérdida de conexión

En la tabla siguiente se muestran los errores de pérdida de conexión que puede encontrarse al trabajar con Base de datos de SQL Azure.

Número de error	Gravedad	Descripción
40197	17	Error en el servicio al procesar la solicitud. Inténtelo de nuevo. Código de error %d.
40501	15	El servicio está ocupado actualmente. Vuelva a intentar la solicitud después de 10 segundos. Código: %d.
40544	15	La base de datos ha alcanzado su cuota de tamaño. Cree particiones o elimine datos, quite índices o consulte la documentación para obtener soluciones posibles. Código: %d.
40549	20	La sesión terminó porque tiene una transacción de larga duración. Intente reducir la transacción.
40550	20	La sesión ha terminado porque ha adquirido demasiados bloqueos. Intente leer o modificar menos filas en una sola transacción.
40551	20	La sesión ha terminado debido al uso excesivo de <b>TEMPDB</b> . Intente modificar la consulta para reducir el uso de espacio de la tabla temporal.
40552	20	La sesión ha terminado debido al excesivo uso de espacio del registro de transacciones. Intente modificar menos filas en una sola transacción.
40553	20	La sesión ha terminado debido al uso excesivo de la memoria. Intente modificar la consulta para procesar menos filas.
40613	17	La base de datos '%.*Is' en el servidor '%.*Is' no está disponible actualmente. Vuelva a intentar la conexión más tarde. Si el problema continúa, póngase en contacto con el servicio de soporte al cliente y proporcióneles el id. de seguimiento de sesión de '%.*Is'.

## Limitación de SQL Azure y descodificación de códigos de motivo

SQL Azure proporciona un servicio de base de datos de varios inquilinos a gran escala en recursos compartidos desde miles de equipos físicos. Cada equipo físico puede dar servicio a muchas bases de datos. Cuando haya recursos suficientes en un equipo, todas las bases de datos ubicadas en el mismo pueden aprovechar completamente los recursos disponibles. Sin embargo, si los recursos se vuelven insuficientes, SQL Azure terminará de manera selectiva las sesiones de base de datos que usan demasiados recursos para proporcionar un sistema estable e impide que las sesiones monopolicen todos los recursos.

SQL Azure emplea una tecnología de equilibrio de carga integrada para garantizar el uso óptimo de los servidores físicos en los centros de datos para que se minimicen los límites en todos los equipos. Además, SQL Azure emplea un mecanismo integrado de limitación para proteger los recursos de un equipo dado, impidiendo que cualquier base de datos prive de los recursos al resto de los inquilinos del equipo (o que ponga en peligro el propio servidor).

La limitación impide de inmediato el uso excesivo de los recursos en un equipo por parte de un inquilino y está pensada para minimizar el impacto de los picos de uso. Existen dos tipos de limitación: suave y estricta. La limitación suave aparece cuando los recursos del equipo como CPU, E/S, almacenamiento y subprocesos de trabajo superan los umbrales de seguridad predefinidos a pesar del esfuerzo por impedirlo del equilibrador de carga.

SQL Azure selecciona un subconjunto de las bases de datos que consumen la mayor parte de los recursos y limita sus actividades. No todas las bases de datos del equipo experimentan la limitación, solo las que usan la mayor parte de los recursos. Esto significa que aquellas bases de datos colindantes están protegidas del uso excesivo de sus vecinos. El uso por debajo del umbral predefinido indica que hay muchos recursos para todas las bases de datos del servidor. La limitación estricta tiene lugar cuando el equipo se queda sin recursos, por ejemplo, sin espacio de almacenamiento. Con este tipo de limitación, no se permiten más conexiones nuevas a las bases de datos hospedadas en el equipo hasta que se liberen recursos. SQL Azure devuelve mensajes de error cuando se intentan nuevas conexiones que indican el recurso que ha superado la limitación.

## Descodificar códigos de motivo

En esta sección se describe cómo descodificar los códigos de motivo que devuelven el código de error 40501 "El servicio está ocupado actualmente. Vuelva a intentar la solicitud después de 10 segundos. Código: %d.". El código de motivo (Code: %d) es un número decimal que contiene el modo límite y los tipos de recurso que se han superado. El modo límite enumera los tipos de instrucción rechazados. El tipo de recurso especifica los recursos superados. La limitación puede ocurrir en varios tipos de recursos al mismo tiempo, como CPU y E/S.

En la tabla siguiente se proporciona una lista de los modos límite.

Código del modo límite	Descripción	Tipos de instrucción rechazados	Instrucciones que todavía se pueden procesar
0	No hay limitación	Ninguno	Todas
1	Rechazar la actualización / inserción	INSERT, UPDATE, CREATE TABLE   INDEX	DELETE, DROP TABLE   INDEX, TRUNCATE
2	Rechazar escrituras	INSERT, UPDATE, DELETE, CREATE, DROP	SELECT
3	Rechazar todo	Todos	Ninguna

Como ejemplo, use 131075 como código de motivo. Para obtener el modo de limitación, aplique el módulo 4 al código de motivo.  $131075 \% 4 = 3$ . El resultado 3 significa que el modo de limitación es "Rechazar todo".

Para obtener el tipo y el recurso de limitación, divida el código de motivo entre 256. Después, convierta el cociente del resultado a su equivalente binario.  $131075 / 256 = 512$  (decimal) y  $512$  (decimal) = 10 00 00 00 00 (binario). Esto significa que la base de datos se limitó por la CPU (tipo de recurso 4) y por una limitación estricta (10).

## Excepciones de SMO en base de datos de SQL Azure

Esta página contiene información sobre las excepciones que se pueden producir al programar con Objetos de administración de SQL Server (SMO) en Microsoft Base de datos de SQL Azure.

Descripción (texto del mensaje)	Causa del mensaje de error
El objeto o una de sus propiedades no se admiten en el tipo de motor de destino.	Se está incluyendo en el script un objeto no admitido en SQL Azure.
Los inicios de sesión no se pueden incluir en el script mediante el identificador de seguridad (SID) desde la base de datos de SQL Azure a una instancia local de SQL Server.	Se está intentando iniciar sesión en SQL Azure con autenticación de Windows. Utilice autenticación de SQL Server en su lugar.
La opción Replication no se admite en la base de datos de SQL Azure. Para generar el script, intente establecer la propiedad TextMode en false.	Se están incluyendo en el script objetos de modo de texto (como procedimientos almacenados, desencadenadores o vistas) con la propiedad Replication.
La versión 10.0 no admite la operación.	Se hace referencia a un objeto que no se admite en SQL Azure.

# TRANSACT-SQL EN SQL AZURE. LENGUAJE DE DEFINICIÓN DE DATOS DDL

## INSTRUCCIONES TRANSACT-SQL ADMITIDAS DE FORMA PARCIAL EN BASE DE DATOS DE SQL AZURE

En Microsoft Base de datos de SQL Azure, algunas instrucciones de Transact-SQL no admiten algunos de los argumentos y opciones que existen en sus instrucciones de Transact-SQL correspondientes en SQL Server 2008. En la tabla siguiente se proporciona una lista de esas instrucciones de Transact-SQL.

ALTER AUTHORIZATION (Base de datos de SQL Azure)	CREATE PROCEDURE (Base de datos de SQL Azure)	DROP TRIGGER (Base de datos de SQL Azure)
ALTER DATABASE (Base de datos SQL Azure)	CREATE SPATIAL INDEX (Base de datos de SQL Azure)	DISABLE TRIGGER (Base de datos de SQL Azure)
ALTER FUNCTION (Base de datos de SQL Azure)	CREATE SYNONYM (Base de datos de SQL Azure)	ENABLE TRIGGER (Base de datos de SQL Azure)
ALTER INDEX (Base de datos de SQL Azure)	CREATE TABLE (Base de datos SQL Azure)	EXECUTE (Base de datos de SQL Azure)

ALTER LOGIN (Base de datos de SQL Azure)	CREATE TRIGGER (Base de datos de SQL Azure)	EXECUTE AS (Base de datos de SQL Azure)
ALTER PROCEDURE (Base de datos de SQL Azure)	CREATE TYPE (Base de datos de SQL Azure)	EXECUTE AS Clause (Base de datos de SQL Azure)
ALTER TABLE (Base de datos SQL Azure)	CREATE USER (Base de datos de SQL Azure)	GRANT (permisos de base de datos de Base de datos de SQL Azure)
ALTER TRIGGER (Base de datos de SQL Azure)	CREATE VIEW (Base de datos de SQL Azure)	GRANT (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)
ALTER USER (Base de datos de SQL Azure)	DENY (permisos de base de datos de Base de datos de SQL Azure)	GRANT (permisos de tipo de Base de datos de SQL Azure)
ALTER VIEW (Base de datos de SQL Azure)	DENY (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)	INSERT (Base de datos de SQL Azure)
CREATE DATABASE (Base de datos SQL Azure)	DENY (permisos de tipo de Base de datos de SQL Azure)	REVOKE (permisos de base de datos de Base de datos de SQL Azure)
CREATE FUNCTION (Base de datos de SQL Azure)	DROP DATABASE (Base de datos de SQL Azure)	REVOKE (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)
CREATE INDEX (Base de datos de SQL Azure)	DROP INDEX (Base de datos de SQL Azure)	REVOKE (permisos de tipo de Base de datos de SQL Azure)
CREATE LOGIN (Base de datos de SQL Azure)	DROP TABLE (Base de datos de SQL Azure)	USE (Base de datos de SQL Azure) Nota: en Microsoft Base de datos de SQL Azure, la instrucción USE no cambia entre bases de datos. Para cambiar las bases de datos, debe conectarse directamente a la base de datos.

## LENGUAJE DDL. INSTRUCCIONES ALTER EN BASE DE DATOS DE SQL AZURE

Dentro del lenguaje de modificación de datos (DML) de Transact-SQL destacan las instrucciones ALTER cuya finalidad general es la modificación de los objetos, propiedades y elementos de la base de datos. En los párrafos siguientes se analizan las diferentes instrucciones ALTER en Base de datos de SQL Azure.

### ALTER AUTHORIZATION

Esta instrucción cambia la propiedad de un elemento protegible. Su sintaxis es la siguiente:

```
ALTER AUTHORIZATION
    ON [ <entity_type> :: ] entity_name
    TO { SCHEMA OWNER | principal_name }

<entity_type> ::=
    {
        OBJECT | TYPE | ROLE | SCHEMA    }
```

Base de datos de SQL Azure **no admite** los siguientes tipos de entidad con la instrucción ALTER AUTHORIZATION:

- ASSEMBLY
- ASYMMETRIC KEY
- CERTIFICATE
- CONTRACT
- DATABASE
- ENDPONT
- FULLTEXT CATALOG
- FULLTEXT STOPLIST
- MESSAGE TYPE
- REMOTE SERVICE BINDING
- ROUTE
- SERVICE
- SYMMETRIC KEY
- XML SCHEMA COLLECTION

## ALTER DATABASE

Esta instrucción modifica una base de datos. Su sintaxis es la siguiente:

```
ALTER DATABASE database_name
{
    MODIFY NAME =new_database_name
    |MODIFY (<edition_options> [, ..n])
}

<edition_options> ::=
{
    (MAXSIZE = {1 | 5 | 10 | 20 | 30 | 40 | 50} GB)
    | (EDITION = {'web' | 'business'})
}
[;]
```

### Argumentos

*database\_name*. Nombre de la base de datos que se va a modificar.

**MODIFY NAME =*new\_database\_name***. Cambia el nombre de la base de datos por el nombre especificado como *new\_database\_name*.

**MODIFY (MAXSIZE = [1 | 5 | 10 | 20 | 30 | 40 | 50] GB)**. Especifica el tamaño máximo de la base de datos. El tamaño máximo debe cumplir el conjunto válido de valores para la propiedad EDITION de la base de datos. En el caso de Web Edition, los valores válidos de MAXSIZE son: 1 GB o 5 GB. Para Business Edition, los valores válidos son: 10 GB, 20 GB, 30 GB, 40 GB o 50 GB. El cambio del tamaño máximo de la base de datos puede hacer que también se cambie la edición de la base de datos. Si MAXSIZE está establecido en un valor de 1GB o 5GB y no se especifica EDITION, la edición de la base de datos se establecerá automáticamente en Web Edition. Si MAXSIZE está establecido en un valor de 10 GB, 20 GB, 30 GB, 40 GB o 50 GB y no se especifica EDITION, la edición de la base de datos se establecerá automáticamente en Business Edition. Si no se especifica MAXSIZE ni EDITION, se creará una base de datos Web Edition con un tamaño de 1GB.

**MODIFY (EDITION = ['web' | 'business'])**. Cambia la edición de la base de datos. En esta versión, Microsoft Base de datos de SQL Azure proporciona dos ediciones de base de datos: Web Edition y Business Edition. Se producirá un error en el cambio de edición si la propiedad MAXSIZE de la base de datos está establecida en un valor

fuera del intervalo válido admitido por esa edición. Cuando se especifica EDITION pero no se especifica MAXSIZE, MAXSIZE se establecerá en el tamaño más restrictivo que admita la edición (1 GB para Web Edition y 10 GB para Business Edition).

Base de datos de SQL Azure **no admite** las siguientes opciones con la instrucción ALTER DATABASE:

- COLLATE
- <file\_and\_filegroup\_options>
  - <add\_or\_modify\_files>
  - <filespec>
  - <add\_or\_modify\_filegroups>
  - <filegroup\_updatability\_option>
- <set\_database\_options>
  - <optionspec>
  - <auto\_option>
  - <change\_tracking\_option>
  - <cursor\_option>
  - <database\_mirroring\_option>
  - <date\_correlation\_optimization\_option>
  - <db\_encryption\_option>
  - <db\_state\_option>
  - <db\_update\_option>
  - <db\_user\_access\_option>

<external\_access\_option>

<parameterization\_option>

<recovery\_option>

<service\_broker\_option>

<snapshot\_option>

<sql\_option>

## Comentarios

Puede emplear la instrucción ALTER DATABASE para cambiar la edición, el nombre y el tamaño máximo de la base de datos después de su creación.

MAXSIZE ofrece la posibilidad de limitar el tamaño de la base de datos. Si el tamaño de la base de datos alcanza el valor de MAXSIZE, recibirá un código de error 40544. Cuando esto ocurre, no puede insertar o actualizar datos ni crear objetos nuevos (como tablas, procedimientos almacenados, vistas y funciones). Sin embargo, todavía puede leer y eliminar datos, truncar tablas, quitar tablas e índices y volver a generar índices. Después, puede actualizar MAXSIZE a un valor mayor que el tamaño actual de la base de datos o eliminar algunos datos para liberar espacio de almacenamiento. Pueden transcurrir hasta quince minutos antes de que se puedan insertar datos nuevos.

El cambio de la edición o el tamaño máximo de la base de datos es una operación sincrónica y sin conexión, y hace que se desconecten las conexiones existentes. Base de datos de SQL Azure está disponible en dos ediciones: Web Edition y Business Edition. Web Edition admite una base de datos de hasta 5 GB de datos. Business Edition admite una base de datos de hasta 50 GB de datos.

Para cambiar la edición de la base de datos después de su creación, use la cláusula MODIFY de la instrucción ALTER DATABASE para cambiar el tamaño máximo y la edición. Para obtener más información sobre las ediciones y los detalles de facturación, vea Cuentas y facturación en SQL Azure. Del mismo modo, puede cambiar el nombre de la base de datos usando la cláusula MODIFY NAME de la instrucción ALTER DATABASE después de su creación. Durante el proceso de cambio de nombre, se pueden cerrar las conexiones a la base de datos.

## ALTER FUNCTION

Esta sentencia modifica una función de Transact-SQL existente creada anteriormente mediante la ejecución de la instrucción CREATE FUNCTION, sin cambiar los permisos y sin que afecte a ninguna otra función, procedimiento almacenado o desencadenador dependiente. Su sintaxis es la siguiente:

### Scalar Functions

```
ALTER FUNCTION [ schema_name. ] function_name ( [ {
@parameter_name [ AS ] [ type_schema_name. ] parameter_data_type
    [ =default ] }
    [ ,...n ]
]
)
RETURNS return_data_type
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
BEGIN
    function_body
    RETURN scalar_expression
END
[ ; ]
```

### Inline Table-valued Functions

```
ALTER FUNCTION [ schema_name. ] function_name ( [ {
@parameter_name [ AS ] [ type_schema_name. ]
parameter_data_type
    [ =default ] }
    [ ,...n ]
]
)
RETURNS TABLE
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
RETURN [ ( ) select_stmt [ ) ]
[ ; ]
```

### Multistatement Table-valued Functions

```
ALTER FUNCTION [ schema_name. ] function_name ( [ {
@parameter_name [ AS ] [ type_schema_name. ]
parameter_data_type
    [ =default ] }
    [ ,...n ]
]
)

```

```

RETURNS @return_variable TABLE <table_type_definition>
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
BEGIN
    function_body
    RETURN
END
[ ; ]

```

```

Function Options<function_option>::=
{
    [ ENCRYPTION ]
    | [ SCHEMABINDING ]
    | [ RETURNS NULL ON NULL INPUT | CALLED ON NULL INPUT ]
    | [ EXECUTE_AS_Clause ]
}

```

```

Table Type Definitions<table_type_definition>:: = ( {
<column_definition> <column_constraint>
    | <computed_column_definition> }
    [ <table_constraint> ] [ ,...n ]
)<column_definition>::=
{
    { column_name data_type }
    [ [ DEFAULT constant_expression ]
      [ COLLATE collation_name ] | [ ROWGUIDCOL ]
    ]
    | [ IDENTITY [ (seed ,increment ) ] ]
    [ <column_constraint> [ ...n ] ]
}
<column_constraint>::=
{
    [ NULL | NOT NULL ]
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [ WITH FILLFACTOR = fillfactor
      | WITH ( < index_option > [ , ...n ] )
    ]
    [ ON { filegroup | "default" } ]
    | [ CHECK (logical_expression ) ] [ ,...n ]
}

```

```

<computed_column_definition>::=column_name AS
computed_column_expression<table_constraint>::=
{
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    (column_name [ ASC | DESC ] [ ,...n ] )
}

```

```

    [ WITH FILLFACTOR = fillfactor
      | WITH ( <index_option> [ , ...n ] )
    | [ CHECK (logical_expression) ] [ ,...n ]
  }

<index_option>::=
{
  PAD_INDEX = { ON | OFF }
  | FILLFACTOR = fillfactor
  | IGNORE_DUP_KEY = { ON | OFF }
  | STATISTICS_NORECOMPUTE = { ON | OFF }
  | ALLOW_ROW_LOCKS = { ON | OFF }
  | ALLOW_PAGE_LOCKS = { ON | OFF }
}

```

Base de datos de SQL Azure **no admite** los siguientes tipos de función con la instrucción ALTER FUNCTION:

- Funciones escalares CLR.
- Funciones con valores de tabla de CLR.

Base de datos de SQL Azure admite funciones escalares, funciones con valores de tabla insertados y funciones con valores de tabla de múltiples instrucciones con la instrucción ALTER FUNCTION tal cual. Para esas funciones, *function\_body* está sujeto a las limitaciones generales del lenguaje de manipulación de datos (DML) de Transact-SQL de Base de datos de SQL Azure.

Para las funciones con valores de tabla de múltiples instrucciones, los siguientes lenguajes de definición de datos (DDL) están sujetos a las limitaciones generales de DDL de Transact-SQL de Base de datos de SQL Azure:

- table\_type\_definition
- column\_definition
- column\_constraint
- computed\_column\_definition
- index\_option

## ALTER INDEX

Esta instrucción modifica un índice existente de una tabla o una vista (relacional o XML); para ello, deshabilita, vuelve a generar o reorganiza el índice, o establece opciones del índice. Su sintaxis es la siguiente:

```
ALTER INDEX { index_name | ALL }
    ON <object>
    { REBUILD
      [
        [ WITH ( <rebuild_index_option> [ ,...n ] ) ]
      ]
    | DISABLE

    | SET ( <set_index_option> [ ,...n ] )
  }
[ ; ]
```

```
<object> ::=
{
  [ database_name. [ schema_name ] . | schema_name. ]
  table_or_view_name
}
```

```
<rebuild_index_option > ::=
{
  IGNORE_DUP_KEY = { ON | OFF }
  | STATISTICS_NORECOMPUTE = { ON | OFF }
  | ONLINE = { ON | OFF }
}
```

```
<set_index_option>::=
{
  IGNORE_DUP_KEY = { ON | OFF }
  | STATISTICS_NORECOMPUTE = { ON | OFF }
}
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción ALTER INDEX:

- REBUILD PARTITION
- REORGANIZE PARTITION
- rebuild\_index\_option:
  - PAD\_INDEX

- FILLFACTOR
- SORT\_IN\_TEMPDB
- ALLOW\_ROW\_LOCKS
- ALLOW\_PAGE\_LOCKS
- MAXDOP
- DATA\_COMPRESSION
- opción range
- single\_partition\_rebuild\_index\_option
- set\_index\_option:
  - ALLOW\_ROW\_LOCKS
  - ALLOW\_PAGE\_LOCKS

## ALTER LOGIN

Esta instrucción cambia las propiedades de una cuenta de inicio de sesión de base de datos en Microsoft Base de datos de SQL Azure. Su sintaxis es la siguiente:

```
ALTER LOGIN login_name
{
  <status_option>
  | WITH <set_option> [ ,... ]
}

<status_option> ::=ENABLE | DISABLE

<set_option> ::=
  PASSWORD = 'password'
  [
    OLD_PASSWORD = 'oldpassword'
  ]
  | NAME = login_name
```

Este diagrama de sintaxis muestra los argumentos y las opciones compatibles en Base de datos de SQL Azure.

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción ALTER LOGIN:

- Opciones de <set\_option>:
  - DEFAULT\_DATABASE
  - DEFAULT\_LANGUAGE
  - CHECK\_POLICY

- CHECK\_EXPIRATION
- CREDENTIAL
- NO CREDENTIAL
- hashed\_password
- Opciones de <password\_option>:
  - MUST\_CHANGE
  - UNLOCK
- Opciones de <cryptographic\_credentials\_option>:
  - ADD CREDENTIAL
  - DROP CREDENTIAL

Con algunos métodos de conexión a Base de datos de SQL Azure, como `sqlcmd`, debe anexar el nombre del servidor SQL Azure al nombre de inicio de sesión en la cadena de conexión usando la notación `the <inicioDeSesión>@<servidor>`. Por ejemplo, si su inicio de sesión es `inicioDeSesión1` y el nombre completo del servidor SQL Azure es `nombreDeServidor.baseDeDatos.windows.net`, el parámetro `username` de la cadena de conexión debe ser `inicioDeSesión1@nombreDeServidor`.

Como la longitud total del parámetro `username` es 128 caracteres, `loginName` está limitado a 127 caracteres menos la longitud del nombre del servidor. En el ejemplo, `loginName` solo puede tener 111 caracteres de longitud porque `nombreDeServidor` tiene 16 caracteres.

## ALTER PROCEDURE

Esta instrucción modifica un procedimiento creado anteriormente. Su sintaxis es la siguiente:

```
ALTER { PROC | PROCEDURE } [schema_name.] procedure_name
    [ { @parameter [ type_schema_name. ] data_type }
      [ VARYING ] [ =default ] [ [ OUT [ PUT ]
        ] [ ,...n ]
      ]
    [ WITH <procedure_option> [ ,...n ] ]
AS
    { <sql_statement> [ ...n ] }

<procedure_option> ::=
    [ RECOMPILE ]
    [ EXECUTE_AS-Clause ]

<sql_statement> ::=
    { [ BEGIN ] statements [ END ] }
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción ALTER PROCEDURE:

- FOR REPLICATION
- Métodos externos
- Opción ENCRYPTION

## ALTER TABLE

Esta instrucción modifica una definición de tabla al alterar, agregar o quitar columnas y restricciones, reasignar particiones, o deshabilitar o habilitar restricciones y desencadenadores.

```
ALTER TABLE [database_name. [ schema_name ] . |schema_name . ]
table_name
{
    ALTER COLUMN column_name
    {
        [ type_schema_name. ] type_name [ ( { precision [
, scale ]
        | max } ) ]
        [ COLLATE collation_name ]
        [NULL | NOT NULL ]
    | {ADD | DROP }
      { PERSISTED }
    }
    | [ WITH { CHECK | NOCHECK } ]

    | ADD
    {
        <column_definition>
        | <computed_column_definition>
        | <table_constraint>
    } [ ,...n ]

    | DROP
    {
        [ CONSTRAINT ] constraint_name
        [ WITH ( <drop_clustered_constraint_option> [ ,...n ] )
    ]

    | COLUMN column_name
    } [ ,...n ]
}
```

```

    | [ WITH { CHECK | NOCHECK } ] { CHECK | NOCHECK }
CONSTRAINT
    { ALL | constraint_name [ ,...n ] }
    | { ENABLE | DISABLE } TRIGGER
    { ALL | trigger_name [ ,...n ] }
    | REBUILD
    [ WITH ( <rebuild_option> [ ,...n ] ) ]
]
}
[ ; ]

```

```

<drop_clustered_constraint_option> ::=
{
    ONLINE = { ON | OFF }
}

```

```

<rebuild_option> ::=
{
    IGNORE_DUP_KEY = { ON | OFF }
    | STATISTICS_NORECOMPUTE = { ON | OFF }
    | ONLINE = { ON | OFF }
}

```

```

<column_definition> ::=
column_name <data_type>
    [ COLLATE collation_name ]
    [ NULL | NOT NULL ]
    [
        [ CONSTRAINT constraint_name ] DEFAULT
        constant_expression
        | [ IDENTITY [ ( seed , increment ) ]
    ]
    [<column_constraint> [ ...n ] ]

```

```

<data type> ::=
[ type_schema_name . ] type_name
    [ ( precision [ , scale ] | max ) ]

```

```

<column_constraint> ::=
[ CONSTRAINT constraint_name ]
{
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [
        | WITH ( < index_option > [ , ...n ] )
    ]
}

```

```

    | [ FOREIGN KEY ]
      REFERENCES [ schema_name . ] referenced_table_name [ (
ref_column ) ]
      [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
      [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
    | CHECK ( logical_expression )
  }

<computed_column_definition> ::=
column_name AS computed_column_expression
[ PERSISTED [ NOT NULL ] ]
[
  [ CONSTRAINT constraint_name ]
  { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  [
    | WITH ( <index_option> [ , ...n ] )
  ]
  | [ FOREIGN KEY ]
    REFERENCES referenced_table_name [ ( ref_column ) ]
    [ ON DELETE { NO ACTION | CASCADE } ]
    [ ON UPDATE { NO ACTION } ]
  | CHECK ( logical_expression )
]

< table_constraint > ::=
[ CONSTRAINT constraint_name ]
{
  { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  (column [ ASC | DESC ] [ ,...n ] )
  [
    | WITH ( <index_option> [ , ...n ] )
  ]
  | FOREIGN KEY
    ( column [ ,...n ] )
    REFERENCES referenced_table_name [ ( ref_column [ ,...n
] ) ]
  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
  | CHECK ( logical_expression )
}

```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción ALTER TABLE:

- xml\_schema\_collection
- Opción de columna SPARSE
- Opción ROWGUIDCOL
- NOT FOR REPLICATION
- Enable/Disable CHANGE\_TRACKING
- SWITCH [PARTITION] TO target\_table [PARTITION]
- SET FILESTREAM\_ON
- Partición concreta REBUILD (sin compatibilidad con particiones)
- Opción REBUILD en tempdb
- Opción REBUILD con compresión
- <column\_set\_definition>
- REBUILD ON
  - filegroup
  - default
  - partition\_schema\_name

Base de datos de SQL Azure no admite tablas sin índices clúster. Si se crea una tabla sin una restricción clúster, se debe crear un índice clúster antes de que se permita una operación de inserción en la tabla.

Al utilizar el argumento <index\_option>, solo puede usar las opciones de índice que están habilitadas para Base de datos de SQL Azure.

## ALTER TRIGGER

Esta instrucción cambia la definición de un desencadenador del lenguaje de manipulación de datos (DML) o del lenguaje de definición de datos (DDL). Su sintaxis es la siguiente:

```
Trigger on an INSERT, UPDATE, or DELETE statement to a table or
view (DML Trigger)
ALTER TRIGGER schema_name .trigger_name
ON ( table | view )
[ WITH <dml_trigger_option> [ ,...n ] ]
( FOR | AFTER | INSTEAD OF )
{ [ DELETE ] [ , ] [ INSERT ] [ , ] [ UPDATE ] }
AS { sql_statement [ ; ] [ ...n ] }
```

```
<dml_trigger_option> ::=
    [ <EXECUTE AS Clause> ]
```

Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or UPDATE statement (DDL Trigger)

```
ALTER TRIGGER trigger_name
ON { DATABASE }
[ WITH <ddl_trigger_option> [ ,...n ] ]
{ FOR | AFTER } { event_type [ ,...n ] | event_group }
AS { sql_statement [ ; ] }
}
```

```
<ddl_trigger_option> ::=
    [ <EXECUTE AS Clause> ]
```

Base de datos de SQL Azure **no admite** desencadenadores en eventos de inicio de sesión. Por tanto, no puede modificar la definición de los desencadenadores de inicio de sesión.

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción ALTER TRIGGER:

- ALL SERVER para desencadenadores DDL
- ENCRYPTION
- EXECUTE AS LOGIN
- EXTERNAL NAME
- NOT FOR REPLICATION

## ALTER USER

Esta instrucción cambia el nombre de un usuario de base de datos o cambia su esquema predeterminado. Su sintaxis es la siguiente:

```
ALTER USER userName
    WITH <set_item> [ ,...n ]

<set_item> ::=
    NAME =newUserName
    | DEFAULT_SCHEMA =schemaName
    | LOGIN = loginName
```

Si la instrucción ALTER USER es la única instrucción en un lote SQL, Base de datos de SQL Azure admitirá la cláusula **WITH LOGIN**. Si la instrucción ALTER USER no es la única instrucción en un lote SQL o se ejecuta en SQL dinámico, no se admitirá la cláusula **WITH LOGIN**.

## ALTER VIEW

Modifica una vista creada anteriormente. Esto incluye una vista indizada. ALTER VIEW no afecta a los desencadenadores ni procedimientos almacenados dependientes y no cambia los permisos.

```
ALTER VIEW [ schema_name . ] view_name [ ( column [ ,...n ] ) ]
[ WITH <view_attribute> [ ,...n ] ]
AS select_statement
[ WITH CHECK OPTION ] [ ; ]

<view_attribute> ::=
{
    [ SCHEMABINDING ]
    [ VIEW_METADATA ]
}
```

Base de datos de SQL Azure **no admite** la opción ENCRYPTION con la instrucción ALTER VIEW.

## LENGUAJE DDL. INSTRUCCIONES CREATE EN BASE DE DATOS DE SQL AZURE

Dentro del lenguaje de modificación de datos (DML) de Transact-SQL destacan las instrucciones CREATE cuya finalidad general es la creación de los objetos, propiedades y elementos de la base de datos. En los párrafos siguientes se analizan las diferentes instrucciones ALTER en Base de datos de SQL Azure.

## CREATE DATABASE

Esta instrucción crea una nueva base de datos. Su sintaxis es la siguiente:

```
CREATE DATABASE database_name
{
    (<edition_options> [, ...n])
}
```

```
<edition_options> ::=
{
    (MAXSIZE = {1 | 5 | 10 | 20 | 30 | 40 | 50} GB)
    | (EDITION = {'web' | 'business'})
}
[;]
```

Copiar

To copy a database:

```
CREATE DATABASE destination_database_name
    AS COPY OF [source_server_name.] source_database_name
[;]
```

## Argumentos

*database\_name*. Nombre de la nueva base de datos. Este nombre debe ser único en el servidor SQL Azure y cumplir las reglas de SQL Server para los identificadores.

*EDITION*. Especifica la edición de la base de datos: 'business' para Business Edition o 'web' para Web Edition. Cuando se especifica EDITION pero no se especifica MAXSIZE, MAXSIZE se establecerá en el tamaño más restrictivo que admita la edición (1 GB para Web Edition y 10 GB para Business Edition).

*MAXSIZE*. Especifica el tamaño máximo de la base de datos. MAXSIZE debe ser válido para el valor de EDITION especificado. En el caso de Web Edition, los valores válidos de MAXSIZE son: 1 GB o 5 GB. Para Business Edition, los valores válidos son: 10 GB, 20 GB, 30 GB, 40 GB o 50 GB. Si MAXSIZE está establecido en un valor de 1GB o 5GB y no se especifica EDITION, la edición de la base de datos se establecerá automáticamente en Web Edition. Si MAXSIZE está establecido en un valor de 10 GB, 20 GB, 30 GB, 40 GB o 50 GB y no se especifica EDITION, la edición de la base de datos se establecerá automáticamente en Business Edition. Si no se especifica MAXSIZE ni EDITION, se creará una base de datos Web Edition con un tamaño de 1GB.

*destination\_database\_name*. Nombre de la base de datos creada por la copia de la base de datos. Este nombre debe ser único en el servidor SQL Azure (destino) y cumplir las reglas de SQL Server para los identificadores.

*AS COPY OF [source\_server\_name.]source\_database\_name*. Para copiar una base de datos al mismo servidor o a un servidor SQL Azure diferente. No se puede usar AS COPY OF con ningún otro argumento de CREATE DATABASE.

*source\_server\_name*. Nombre del servidor SQL Azure donde se encuentra la base de datos de origen. Este parámetro es opcional cuando la base de datos de origen y la base de datos de destino van a estar en el mismo servidor SQL Azure. El argumento AS COPY OF no admite nombres completos de dominio únicos. Es decir, si el nombre de dominio completo de su servidor es `serverName.database.windows.net`, utilice solo `serverName` durante la copia de la base de datos.

*source\_database\_name*. Nombre de la base de datos que se va a copiar.

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción CREATE DATABASE:

- Parámetros relacionados con la colocación física de los archivos, como `<filespec>` y `<filegroup>`.
- COLLATE en una base de datos de usuario.
- Opciones de acceso externo, como DB\_CHAINING y TRUSTWORTHY.
- Adjuntar una base de datos.
- Opciones de Service Broker, como ENABLE\_BROKER, NEW\_BROKER y ERROR\_BROKER\_CONVERSATIONS.
- Instantánea de base de datos.

## Comentarios

Las bases de datos en Base de datos de SQL Azure tienen varias configuraciones predeterminadas que se establecen cuando se crea la base de datos.

MAXSIZE ofrece la posibilidad de limitar el tamaño de la base de datos. Si el tamaño de la base de datos alcanza el valor de MAXSIZE, recibirá un código de error 40544. Cuando esto ocurre, no puede insertar o actualizar datos ni crear objetos nuevos (como tablas, procedimientos almacenados, vistas y funciones). Sin embargo, todavía puede leer y eliminar datos, truncar tablas, quitar tablas e índices y volver a generar índices. Después, puede actualizar MAXSIZE a un valor mayor que el tamaño actual de la base de datos o eliminar algunos datos para liberar espacio de almacenamiento. Pueden transcurrir hasta quince minutos antes de que se puedan insertar datos nuevos.

CREATE DATABASE debe ser la única instrucción en un lote de Transact-SQL. Debe estar conectado a la base de datos maestra (**master**) al ejecutar la instrucción CREATE DATABASE.

## Copias de la base de datos

La copia de una base de datos mediante la instrucción CREATE DATABASE es una operación asincrónica. Por tanto, no se necesita una conexión con el servidor SQL Azure mientras dura todo el proceso de copia. La instrucción CREATE DATABASE devolverá el control al usuario antes de que se complete la operación de copia de la base de datos. Es decir, la instrucción CREATE DATABASE vuelve correctamente cuando la copia de la base de datos todavía está en curso. Puede supervisar el proceso de copia con las vistas `sys.dm_database_copies` y `sys.databases`. Una vez que el proceso de copia se completa correctamente, la base de datos de destino es transaccionalmente coherente con la base de datos de origen.

Se aplican las siguientes reglas de sintaxis y semánticas a su uso del argumento AS COPY OF:

- El nombre del servidor de origen y el nombre del servidor para el destino de la copia pueden ser iguales o diferentes. Cuando son iguales, este parámetro es opcional y se usará de forma predeterminada el contexto de servidor de la sesión actual.
- Se deben especificar los nombres de las bases de datos de origen y de destino, que deben ser únicos y cumplir las reglas de SQL Server para los identificadores. Para obtener más información, vea [Identificadores](#).
- La instrucción CREATE DATABASE se debe ejecutar dentro del contexto de la base de datos maestra del servidor SQL Azure donde se creará la nueva base de datos.
- Después de completarse la copia, la base de datos de destino se debe administrar como una base de datos independiente. Puede ejecutar las instrucciones ALTER DATABASE y DROP DATABASE en la nueva base de datos independientemente de la base de datos de origen. También puede copiar la nueva base de datos a otra base de datos nueva.
- No se puede tener acceso a la base de datos de destino hasta que el proceso de copia se haya completado. Puede comprobar el estado del proceso de copia consultando la columna `state` de la vista `sys.databases` o la columna `percentage_complete` de la vista `sys.dm_database_copies` en el servidor SQL Azure de destino.

Durante el proceso de copia, la columna state de la vista sys.databases muestra Copying en el servidor SQL Azure de destino. Además, la columna percentange\_complete de sys.dm\_database\_copies muestra el porcentaje de bytes copiados al servidor de destino.

- Se puede seguir teniendo acceso a la base de datos de origen mientras la copia de la base de datos está en curso.

## CREATE FUNCTION

Esta sentencia crea una función definida por el usuario. Es una rutina Transact-SQL guardada que devuelve un valor. Las funciones definidas por el usuario no se pueden usar para realizar acciones que modifican el estado de la base de datos. Las funciones definidas por el usuario, como las funciones de sistema, se pueden invocar desde una consulta. Las funciones escalares se pueden ejecutar mediante una instrucción EXECUTE, igual que los procedimientos almacenados.

Su sintaxis es la siguiente:

### Scalar Functions

```
CREATE FUNCTION [ schema_name. ] function_name( [ {
@parameter_name [ AS ] [ type_schema_name. ] parameter_data_type
    [ =default ] [ READONLY ] }
    [ ,...n ]
]
)
RETURNS return_data_type
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
BEGIN
    function_body
    RETURN scalar_expression
END
[ ; ]
```

### Inline Table-Valued Functions

```
CREATE FUNCTION [ schema_name. ] function_name ( [ {
@parameter_name [ AS ] [ type_schema_name. ]
parameter_data_type
    [ =default ] [ READONLY ] }
    [ ,...n ]
]
)
RETURNS TABLE
    [ WITH <function_option> [ ,...n ] ]
```

```

    [ AS ]
    RETURN [ ( ) select_stmt [ ) ]
[ ; ]

```

#### Multistatement Table-valued Functions

```

CREATE FUNCTION [ schema_name. ] function_name ( [ {
@parameter_name [ AS ] [ type_schema_name. ]
parameter_data_type
    [ =default ] [ READONLY ] }
    [ ,...n ]
]
)
RETURNS @return_variable TABLE <table_type_definition>
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
BEGIN
    function_body
    RETURN
END
[ ; ]

```

#### Function Options<function\_option>::=

```

{
    [ ENCRYPTION ]
    | [ SCHEMABINDING ]
    | [ RETURNS NULL ON NULL INPUT | CALLED ON NULL INPUT ]
    | [ EXECUTE_AS_Clause ]
}

```

#### Table Type Definitions<table\_type\_definition>::= ( {

```

<column_definition> <column_constraint>
    | <computed_column_definition> }
    [ <table_constraint> ] [ ,...n ]
)<column_definition>::=
{
    { column_name data_type }
    [ [ DEFAULT constant_expression ]
      [ COLLATE collation_name ] | [ ROWGUIDCOL ]
    ]
    | [ IDENTITY [ (seed , increment ) ] ]
    [ <column_constraint> [ ...n ] ]
}
<column_constraint>::=
{
    [ NULL | NOT NULL ]
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
}

```

```

        [ WITH FILLFACTOR =fillfactor
          | WITH ( < index_option > [ , ...n ] )
          [ ON { filegroup | "default" } ]
        | [ CHECK (logical_expression) ] [ ,...n ]
      }

<computed_column_definition>::=column_name AS
computed_column_expression<table_constraint>::=
{
  { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  (column_name [ ASC | DESC ] [ ,...n ] )
  [ WITH FILLFACTOR = fillfactor
    | WITH ( <index_option> [ , ...n ] )
  | [ CHECK (logical_expression) ] [ ,...n ]
}

<index_option>::=
{
  PAD_INDEX = { ON | OFF }
  | FILLFACTOR = fillfactor
  | IGNORE_DUP_KEY = { ON | OFF }
  | STATISTICS_NORECOMPUTE = { ON | OFF }
  | ALLOW_ROW_LOCKS = { ON | OFF }
  | ALLOW_PAGE_LOCKS = { ON | OFF }
}

```

Base de datos de SQL Azure **no admite** crear funciones escalares de CLR y funciones con valores de tabla de CLR mediante la instrucción CREATE FUNCTION. Base de datos de SQL Azure admite crear funciones escalares, funciones con valores de tabla insertadas y funciones con valores de tabla de múltiples instrucciones sin realizar ninguna modificación.

Para las funciones escalares y las funciones con valores de tabla de múltiples instrucciones, *function\_body* está sujeto a las limitaciones generales del lenguaje de manipulación de datos (DML) de Transact-SQL de Base de datos de SQL Azure. En la lista siguiente se muestran los DDL, que están sujetos a las limitaciones generales del lenguaje de definición de datos (DDL) de Transact-SQL de Base de datos de SQL Azure:

- table\_type\_definition
- column\_definition
- column\_constraint
- computed\_column\_definition
- index\_option

## CREATE INDEX

Esta instrucción crea un índice relacional en una tabla especificada o en una vista de una tabla especificada. Se puede crear un índice antes de que la tabla posea datos. Los índices relacionales se pueden crear en tablas o en vistas de otra base de datos especificando un nombre completo de base de datos.

Puesto que Microsoft Base de datos de SQL Azure no admite tablas de montón, una tabla debe tener un índice clúster. Si se crea una tabla sin una restricción clúster, se deberá crear un índice clúster antes de que se permita una operación de inserción en la tabla.

Su sintaxis es la siguiente:

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
    ON <object> (column [ ASC | DESC ] [ ,...n ] )
    [ INCLUDE (column_name [ ,...n ] ) ]
    [ WHERE <filter_predicate> ]
    [ WITH ( <relational_index_option> [ ,...n ] ) ]
```

```
[ ; ]
```

```
<object> ::=
```

```
{
    [ database_name. [ schema_name ] . | schema_name. ]
    table_or_view_name
}
```

```
<relational_index_option> ::=
```

```
{
    | IGNORE_DUP_KEY = { ON | OFF }
    | STATISTICS_NORECOMPUTE = { ON | OFF }
    | DROP_EXISTING = { ON | OFF }
    | ONLINE = { ON | OFF }
}
```

```
<filter_predicate> ::=
```

```
<conjunct> [ AND <conjunct> ]
```

```
<conjunct> ::=
```

```
<disjunct> | <comparison>
```

```
<disjunct> ::=column_name IN (constant ,...)
```

```
<comparison> ::=column_name <comparison_op>
```

```
constant<comparison_op> ::=
```

{ IS | IS NOT | = | <> | != | > | >= | !> | < | <= | !< }

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción CREATE INDEX:

- Palabra clave ON para
  - partition\_schema\_name
  - filegroup name
  - default
- FILESTREAM\_ON
- relational\_index\_option:
  - PAD\_INDEX
  - FILLFACTOR
  - SORT\_IN\_TEMPDB
  - ALLOW\_ROW\_LOCKS
  - ALLOW\_PAGE\_LOCKS
  - MAXDOP
  - DATA\_COMPRESSION
- Índice relacional compatible con versiones anteriores.

Base de datos de SQL Azure admite el formato de nombre de tres partes *database\_name.[schema\_name].object\_name* cuando *database\_name* es la base de datos actual o *database\_name* es tempdb y *object\_name* empieza por #. Base de datos de SQL Azure **no admite** nombres de cuatro partes.

## CREATE LOGIN

Esta sentencia crea un nuevo inicio de sesión de base de datos en Microsoft Base de datos de SQL Azure. Su sintaxis es la siguiente:

```
CREATE LOGIN loginName { WITH <option_list1> }
<option_list1> ::=
    PASSWORD = { 'password' }
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción CREATE LOGIN:

- hashed\_password
- MUST\_CHANGE option
- Opciones de <option\_list2>:

- SID
- DEFAULT\_DATABASE
- DEFAULT\_LANGUAGE
- CHECK\_EXPIRATION
- CHECK\_POLICY and
- CREDENTIAL

En algunos métodos de conexión a Base de datos de SQL Azure, como sqlcmd, debe anexar el nombre del servidor SQL Azure al nombre de inicio de sesión en la cadena de conexión usando la notación the *<inicioDeSesión>@<servidor>*. Por ejemplo, si su inicio de sesión es *inicioDeSesión1* y el nombre completo del servidor SQL Azure es *nombreDeServidor.baseDeDatos.windows.net*, el parámetro *username* de la cadena de conexión debe ser *inicioDeSesión1@nombreDeServidor*.

Como la longitud total del parámetro *username* es 128 caracteres, *loginName* está limitado a 127 caracteres menos la longitud del nombre del servidor. En el ejemplo, *loginName* solo puede tener 111 caracteres de longitud porque *nombreDeServidor* tiene 16 caracteres.

CREATE LOGIN debe ser la única instrucción en un lote de SQL.

## CREATE PROCEDURE

Esta instrucción crea un procedimiento almacenado. Su sintaxis es la siguiente:

```
CREATE { PROC | PROCEDURE } [schema_name.] procedure_name
    [ { @parameter [ type_schema_name. ] data_type }
    [ VARYING ] [ =default ] [ OUT | OUTPUT ] [READONLY]
    ] [ ,...n ]
[ WITH <procedure_option> [ ,...n ] ]
AS { <sql_statement> [;][ ...n ] }
[;]
<procedure_option> ::=
    [ RECOMPILE ]
    [ EXECUTE AS Clause ]
<sql_statement> ::=
{ [ BEGIN ] statements [ END ] }
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción CREATE PROCEDURE:

- Opción FOR REPLICATION
- EXTERNAL NAME (métodos externos)
- Opción ENCRYPTION

## CREATE SPATIAL INDEX

Esta instrucción crea un índice espacial en una tabla y columna especificadas de la base de datos actual. Se puede crear un índice antes de que la tabla posea datos. Su sintaxis es la siguiente:

```
CREATE SPATIAL INDEX index_name
  ON <object> (spatial_column_name )
  {
    [ USING <geometry_grid_tessellation> ]
    WITH ( <bounding_box>
          [ [,] <tessellation_parameters> [,... n ] ]
          [ [,] <spatial_index_option> [,... n ] ] )
    | [ USING <geography_grid_tessellation> ]
      [ WITH ( [ <tessellation_parameters> [,... n ] ]
              [ [,] <spatial_index_option> [,... n ] ] ) ]
  }
```

```
[ ; ]
```

```
<object> ::=
{
  [database_name. [schema_name ] . | schema_name. ]
  table_name<geometry_grid_tessellation> ::=
{ GEOMETRY_GRID }
```

```
<bounding_box> ::=
BOUNDING_BOX = ( {
  xmin, ymin, xmax, ymax
  | <named_bb_coordinate>, <named_bb_coordinate>,
  <named_bb_coordinate>, <named_bb_coordinate>
  } ) <named_bb_coordinate> ::= { XMIN = xmin | YMIN = ymin |
XMAX = xmax | YMAX=ymax }
```

```
<tessellation_parameters> ::=
{
  GRIDS = ( { <grid_density> [ ,... n ] | <density>,
  <density>, <density>, <density> } )
  | CELLS_PER_OBJECT = n
}
```

```
<grid_density> ::=
{
  LEVEL_1 = <density>
  | LEVEL_2 = <density>
```

```

    | LEVEL_3 = <density>
    | LEVEL_4 = <density>
}

<density> ::= { LOW | MEDIUM | HIGH }

<geography_grid_tessellation> ::=
{ GEOGRAPHY_GRID }

<spatial_index_option> ::=
{
    IGNORE_DUP_KEY = OFF
    | STATISTICS_NORECOMPUTE = { ON | OFF }
    | DROP_EXISTING = { ON | OFF }
    | ONLINE = OFF
}

```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción CREATE SPATIAL INDEX:

- Palabra clave ON para
  - partition\_schema\_name
  - filegroup name
  - default
- relational\_index\_option:
  - PAD\_INDEX
  - FILLFACTOR
  - SORT\_IN\_TEMPDB
  - ALLOW\_ROW\_LOCKS
  - ALLOW\_PAGE\_LOCKS
  - MAXDOP
- Índice relacional compatible con versiones anteriores.

## CREATE SYNONYM

Esta sentencia crea un sinónimo, que es un nombre alternativo para una tabla, una vista u otro objeto de base de datos.

```

CREATE SYNONYM [ schema_name_1. ] synonym_name FOR < object ><
object > ::=
{
    [database_name. [ schema_name_2 ].| schema_name_2. ]
    object_name
}

```

## CREATE TABLE

Esta sentencia crea una nueva tabla. Su sintaxis es la siguiente:

```
CREATE TABLE
    [ database_name . [ schema_name ] . | schema_name . ]
table_name
    ( { <column_definition> | <computed_column_definition>
      [ <table_constraint> ] [ ,...n ] }

[ ; ]

<column_definition> ::=column_name <data_type>
    [ COLLATE collation_name ]

    [ NULL | NOT NULL ]
    [
        [ CONSTRAINT constraint_name ] DEFAULT
constant_expression ]
    | [ IDENTITY [ ( seed ,increment ) ]
    ]
    [ <column_constraint> [ ...n ] ]

<data type> ::=
[ type_schema_name . ] type_name
    [ (precision [ ,scale ] | max
]

<column_constraint> ::=
[ CONSTRAINT constraint_name ]
{
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [
        WITH ( < index_option > [ , ...n ] )
    ]
| [ FOREIGN KEY ]
    REFERENCES [ schema_name . ] referenced_table_name [ (
ref_column ) ]
    [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
    [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
| CHECK (logical_expression )
}
```

```

<computed_column_definition> ::=column_name AS
computed_column_expression
[ PERSISTED [ NOT NULL ] ]
[
  [ CONSTRAINT constraint_name ]
  { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  [
    WITH ( <index_option> [ , ...n ] )
  ]
  | [ FOREIGN KEY ]
  REFERENCES referenced_table_name [ (ref_column ) ]
  [ ON DELETE { NO ACTION | CASCADE } ]
  [ ON UPDATE { NO ACTION } ]
  [ NOT FOR REPLICATION ]
  | CHECK (logical_expression )
]

< table_constraint > ::=
[ CONSTRAINT constraint_name ]
{
  { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  (column [ ASC | DESC ] [ ,...n ] )
  [
    WITH ( <index_option> [ , ...n ] )
  ]

  | FOREIGN KEY
  ( column [ ,...n ] )
  REFERENCES referenced_table_name [ (ref_column [ ,...n
] ) ]
  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]
  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET
DEFAULT } ]

  | CHECK (logical_expression )
}

<index_option> ::=
{
  IGNORE_DUP_KEY = { ON | OFF }
  | STATISTICS_NORECOMPUTE = { ON | OFF }
}

```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción CREATE TABLE:

- Palabra clave ON { *partition\_schema\_name* | *filegroup* | "default" }
- TEXTIMAGE\_ON
- FILESTREAM\_ON
- <column\_definition>:
  - FILESTREAM
  - NOT FOR REPLICATION
  - ROWGUIDCOL
  - SPARSE
- <data type>:
  - CONTENT
  - DOCUMENT
  - xml\_schema\_collection
- <column\_constraint>:
  - FILLFACTOR
  - ON
  - NOT FOR REPLICATION
- <computed\_column\_definition>:
  - FILLFACTOR
  - NOT FOR REPLICATION
  - ON
- <column\_set\_definition>
- <table\_constraint>:
  - FILLFACTOR
  - ON
  - NOT FOR REPLICATION
- <table\_option>
- <index\_option>:
  - PAD\_INDEX
  - FILLFACTOR
  - ALLOW\_ROW\_LOCKS
  - ALLOW\_PAGE\_LOCKS
  - DATA\_COMPRESSION
  - ON PARTITIONS

Puesto que Base de datos de SQL Azure no admite tablas de montón, una tabla debe tener un índice clúster. Si se crea una tabla sin una restricción clúster, se deberá crear un índice clúster antes de que se permita una operación de inserción en la tabla.

## CREATE TRIGGER

Esta instrucción crea un desencadenador del lenguaje de manipulación de datos (DML) o del lenguaje de definición de datos (DDL). Un desencadenador es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos.

Su sintaxis es la siguiente:

Trigger on an INSERT, UPDATE, or DELETE statement to a table or view (DML Trigger)

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS { sql_statement [ ; ] [ ,...n ] [ ; ] > }
```

```
<dml_trigger_option> ::=
    [ EXECUTE AS Clause ]
```

Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or UPDATE STATISTICS statement (DDL Trigger)

```
CREATE TRIGGER trigger_name
ON { DATABASE }
[ WITH <ddl_trigger_option> [ ,...n ] ]
{ FOR | AFTER } { event_type | event_group } [ ,...n ]
AS { sql_statement [ ; ] [ ,...n ] [ ; ] }
```

```
<ddl_trigger_option> ::=
    [ EXECUTE AS Clause ]
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones con la instrucción CREATE TRIGGER:

- ENCRYPTION.
- WITH APPEND.
- NOT FOR REPLICATION.
- Opción EXTERNAL NAME (no hay compatibilidad con ningún método externo).
- Opción ALL SERVER (desencadenador DDL).
- Desencadenar en un evento LOGON (desencadenador de inicio de sesión).

## CREATE TYPE

Esta sentencia crea un tipo de datos de alias o un tipo definido por el usuario en la base de datos actual. Su sintaxis es la siguiente:

```
CREATE TYPE [ schema_name. ] type_name
{
    FROM base_type
    [ ( precision [ , scale ] ) ]
    [ NULL | NOT NULL ]
    | AS TABLE ( { <column_definition> |
<computed_column_definition> }
                [ <table_constraint> ] [ ,...n ] )
} [ ; ]

<column_definition> ::=column_name <data_type>
    [ COLLATE collation_name ]
    [ NULL | NOT NULL ]
    [
        DEFAULT constant_expression ]
    | [ IDENTITY [ ( seed ,increment ) ]
    ]
    [ <column_constraint> [ ...n ] ]

<data type> ::=
[ type_schema_name. ] type_name
    [ ( precision [ , scale ] | max ) ]

<column_constraint> ::=
{
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [
        WITH ( <index_option> [ ,...n ] )
    ]
    | CHECK (logical_expression)
}

<computed_column_definition> ::=column_name AS
computed_column_expression
[ PERSISTED [ NOT NULL ] ]
[
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [
        WITH ( <index_option> [ ,...n ] )
    ]
]
```

```

    | CHECK (logical_expression)
]
<table_constraint> ::=
{
    { PRIMARY KEY | UNIQUE }
      [ CLUSTERED | NONCLUSTERED ]
      ( column [ ASC | DESC ] [ ,...n ]
    )
    [
      WITH ( <index_option> [ ,...n ] )
    ]
    | CHECK ( logical_expression )
}
<index_option> ::=
{
    IGNORE_DUP_KEY = { ON | OFF }
}

```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción CREATE TYPE:

- EXTERNAL NAME
- assembly\_name
- class\_name
- <column\_definition>:
  - Opción ROWGUIDCOL
- <data\_type>:
  - Opción CONTENT
  - Opción DOCUMENT
  - xml\_schema\_collection

## CREATE USER

Esta sentencia agrega un usuario a la base de datos actual.

```

CREATE USER user_name
  [ { { FOR | FROM }
    {
      LOGIN login_name
    }
    | WITHOUT LOGIN
  }
]
  [ WITH DEFAULT_SCHEMA = schema_name ]

```

Base de datos de SQL Azure no admite los siguientes argumentos cuando se utiliza la instrucción CREATE USER.

- CERTIFICATE
- ASYMMETRIC\_KEY

Si la instrucción CREATE USER es la única instrucción en un lote SQL, Base de datos de SQL Azure admite la cláusula **FOR | FROM LOGIN**. Si la instrucción CREATE USER no es la única instrucción en un lote SQL o se ejecuta en SQL dinámico, no se admitirá la cláusula **FOR | FROM LOGIN**.

## CREATE VIEW

Esta instrucción crea una tabla virtual que representa los datos de una o más tablas de una forma alternativa. CREATE VIEW debe ser la primera instrucción en un lote de consultas. Su sintaxis es la siguiente:

```
CREATE VIEW [ schema_name . ] view_name [ ( column [ ,...n ] ) ]  
[ WITH <view_attribute> [ ,...n ] ]  
AS select_statement  
[ WITH CHECK OPTION ] [ ; ]
```

```
<view_attribute> ::=  
{  
    [ SCHEMABINDING ]  
    [ VIEW_METADATA ] }
```

Base de datos de SQL Azure **no admite** el argumento ENCRYPTION con la instrucción CREATE VIEW.

## LENGUAJE DML. INSTRUCCIONES DROP EN BASE DE DATOS DE SQL AZURE

Dentro del lenguaje de modificación de datos (DML) de Transact-SQL destacan las instrucciones DROP cuya finalidad general es el borrado de los objetos, propiedades y elementos de la base de datos. En los párrafos siguientes se analizarán las diferentes instrucciones DROP en Base de datos de SQL Azure.

## DROP DATABASE

Esta instrucción quita una base de datos. Su sintaxis es la siguiente:

```
DROP DATABASE database_name
[;]
```

Base de datos de SQL Azure **no admite** el argumento *database\_snapshot\_name* cuando se utiliza la instrucción DROP DATABASE. DROP DATABASE debe ser la única instrucción en un lote de SQL y solo puede quitar una base de datos a la vez.

## DROP INDEX

Esta instrucción quita uno o más índices XML, filtrados, espaciales o relacionales de la base de datos actual. Su sintaxis es la siguiente:

```
DROP INDEX
{ <drop_relational_or_xml_or_spatial_index> [ ,...n ]
}

<drop_relational_or_xml_or_spatial_index> ::=
    index_name ON <object>

<object> ::=
{
    [ database_name. [ schema_name ] . | schema_name. ]
    table_or_view_name
}
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción DROP INDEX:

```
database_name
<drop_backward_compatible_index>
<drop_clustered_index_option>
```

Base de datos de SQL Azure admite el formato de nombre de tres partes *database\_name.[schema\_name].object\_name* cuando *database\_name* es la base de datos actual o *database\_name* es *tempdb* y *object\_name* empieza por #.

Base de datos de SQL Azure **no admite** nombres de cuatro partes.

## DROP TABLE

Esta instrucción quita una o varias definiciones de tabla y todos los datos, índices, desencadenadores, restricciones y especificaciones de permisos de las tablas quitadas. Todas las vistas o los procedimientos almacenados que hagan referencia a la tabla quitada se deben quitar explícitamente mediante DROP VIEW o DROP PROCEDURE. Su sintaxis es la siguiente:

```
DROP TABLE [ database_name . [ schema_name ] . | schema_name .
]
           table_name [ ,...n ] [ ; ]
```

Base de datos de SQL Azure admite el formato de nombre de tres partes *database\_name.[schema\_name].object\_name* cuando *database\_name* es la base de datos actual o *database\_name* es tempdb y *object\_name* empieza por #. Base de datos de SQL Azure **no admite** nombres de cuatro partes.

## DROP TRIGGER

Esta instrucción quita uno o más desencadenadores del lenguaje de manipulación de datos (DML) o el lenguaje de definición de datos (DDL) de la base de datos actual.

Su sintaxis es la siguiente para instrucciones INSERT, UPDATE, o DELETE para tabla o vista (DML Trigger) es la siguiente:

```
DROP TRIGGER [schema_name.]trigger_name [ ,...n ] [ ; ]
```

Su sintaxis para instrucciones CREATE, ALTER, DROP, GRANT, DENY, REVOKE o UPDATE (DDL Trigger) es la siguiente:

```
DROP TRIGGER trigger_name [ ,...n ]
ON { DATABASE }
[ ; ]
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción DROP TRIGGER:

- Opción ALL SERVER para desencadenadores DDL.
- Desencadenadores logon.

# TRANSACT-SQL EN SQL AZURE. LENGUAJES DE MANIPULACIÓN Y CONTROL DE DATOS DML Y DCL

## LENGUAJE DE MANIPULACIÓN DE DATOS DML. INSTRUCCIÓN INSERT EN BASE DATOS DE SQL AZURE

Dentro del lenguaje de modificación de datos (DML) de Transact-SQL destaca la única instrucción que tiene acotaciones en SQL Azure es INSERT. El resto (UPDATE, DELETE, etc.) mantienen la misma sintaxis que en Transact-SQL.

La instrucción INSERT agrega una o varias filas nuevas a una tabla o una vista.

Su sintaxis es la siguiente:

```
[ WITH <common_table_expression> [ ,...n ] ]  
INSERT  
    [ TOP ( expression ) [ PERCENT ] ]  
    [ INTO ]  
    { <object>
```

```

    [ WITH ( <Table_Hint_Limited> [ ...n ] ) ]
  }
{
  [ ( column_list ) ]
  [ <OUTPUT Clause> ]
  { VALUES ( { DEFAULT | NULL | expression } [ ,...n ] ) [
, ...n ]
  | derived_table
  | execute_statement
    | <dml_table_source>
  | DEFAULT VALUES
  }
}
[; ]

<object> ::=
{
  [schema_name. ]
  table_or_view_name
}

```

```

<dml_table_source> ::=
  SELECT <select_list>
  FROM ( <dml_statement_with_output_clause> )
  [AS] table_alias [ ( column_alias [ ,...n ] ) ]
  [ WHERE <search_condition> ]
  [ OPTION ( <query_hint> [ ,...n ] ) ]

```

Base de datos de SQL Azure no admite los siguientes argumentos y opciones cuando se utiliza la instrucción INSERT:

- Con nombres de tres y cuatro partes.
- Con las opciones OPENQUERY u OPENROWSET.

## LENGUAJE DE CONTROL DE DATOS DCL EN BASE DATOS DE SQL AZURE

Dentro del lenguaje de control de datos DCL existen varias instrucciones de Transact-SQL con sintaxis parcialmente admitidas en SQL Azure. En los párrafos siguientes se abordarán estas sentencias.

También se abordan instrucciones como USE, EXECUTE, ENABLE y DISABLE, cuya sintaxis en Transact-SQL solo se admite parcialmente en SQL Azure.

## GRANT (permisos de base de datos de Base de Datos de SQL Azure)

Esta instrucción concede permisos en una base de datos. Su sintaxis es la siguiente:

```
GRANT <permission> [ ,...n ]
    TO <database_principal> [ ,...n ] [ WITH GRANT OPTION ]
    [ AS <database_principal> ]
```

```
<permission> ::= permission | ALL [ PRIVILEGES ]
```

```
<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** las siguientes opciones con esta instrucción GRANT:

- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## GRANT (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)

Esta instrucción concede permisos en un usuario de base de datos, un rol de base de datos o un rol de aplicación. Su sintaxis es la siguiente:

```
GRANT permission [ ,...n ]
    ON
    { [ USER ::database_user ]
      | [ ROLE ::database_role ]
    }
    TO <database_principal> [ ,...n ]
```

```

    [ WITH GRANT OPTION ]
    [ AS <database_principal> ]

<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login

```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza esta instrucción GRANT:

- GRANT ON APPLICATION ROLE
- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## GRANT (permisos de tipo de Base de datos de SQL Azure)

Esta instrucción concede permisos para un tipo. Su sintaxis es la siguiente:

```

GRANT permission [ ,...n ] ON TYPE :: [ schema_name . ]
type_name
    TO <database_principal> [ ,...n ]
    [ WITH GRANT OPTION ]
    [ AS <database_principal> ]

<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login

```

Base de datos de SQL Azure **no admite** las siguientes opciones cuando se usa esta instrucción GRANT:

- Opciones de <database\_principal>:
  - Application\_role

- Database\_user\_mapped\_to\_Windows\_User
- Database\_user\_mapped\_to\_Windows\_Group
- Database\_user\_mapped\_to\_certificate
- Database\_user\_mapped\_to\_asymmetric\_key

## REVOKE (permisos de base de datos de Base de datos de SQL Azure)

Esta instrucción revoca los permisos concedidos y denegados en una base de datos. Su sintaxis es la siguiente:

```
REVOKE [ GRANT OPTION FOR ] <permission> [ ,...n ]
    { TO | FROM } <database_principal> [ ,...n ]
    [ CASCADE ]
    [ AS <database_principal> ]
```

```
<permission> ::= permission | ALL [ PRIVILEGES ]
```

```
<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** las siguientes opciones cuando se usa esta instrucción REVOKE:

- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## REVOKE (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)

Esta instrucción revoca permisos concedidos o denegados para un usuario de base de datos o un rol de base de datos. Su sintaxis es la siguiente:

```
REVOKE [ GRANT OPTION FOR ] permission [ ,...n ]
      ON
      { [ USER ::database_user ]
        | [ ROLE ::database_role ]
      }
      { FROM | TO } <database_principal> [ ,...n ]
      [ CASCADE ]
      [ AS <database_principal> ]

<database_principal> ::=Database_user
      | Database_role
      | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza esta instrucción REVOKE:

- REVOKE ON APPLICATION ROLE
- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## REVOKE (permisos de tipo de Base de datos de SQL Azure)

Esta instrucción revoca permisos en un tipo. Su sintaxis es la siguiente:

```
REVOKE [ GRANT OPTION FOR ] permission [ ,...n ]
      ON TYPE :: [ schema_name ].type_name
      { FROM | TO } <database_principal> [ ,...n ]
      [ CASCADE ]
      [ AS <database_principal> ]
```

```
<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** las siguientes opciones con esta instrucción REVOKE:

- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## DENY (permisos de base de datos de Base de datos de SQL Azure)

Esta instrucción deniega permisos en una base de datos. Su sintaxis es la siguiente:

```
DENY <permission> [ ,...n ]
    TO <database_principal> [ ,...n ] [ CASCADE ]
    [ AS <database_principal> ]

<permission> ::= permission | ALL [ PRIVILEGES ]

<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** las siguientes opciones con la instrucción DENY de permisos de base de datos:

- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group

- Database\_user\_mapped\_to\_certificate
- Database\_user\_mapped\_to\_asymmetric\_key

## DENY (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)

Esta instrucción deniega permisos concedidos para un usuario de base de datos o un rol de base de datos. Su sintaxis es la siguiente:

```
DENY permission [ ,...n ]
    ON
    { [ USER ::database_user ]
      | [ ROLE ::database_role ]
    }
    TO <database_principal> [ ,...n ]
    [ CASCADE ]
    [ AS <database_principal> ]

<database_principal> ::=Database_user
    | Database_role
    | Database_user_with_no_login
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza esta instrucción DENY:

- DENY ON APPLICATION ROLE
- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## DENY (permisos de tipo de Base de datos de SQL Azure)

Esta instrucción deniega permisos en un tipo. Su sintaxis es la siguiente:

```

DENY permission [ ,...n ] ON TYPE :: [ schema_name . ]
type_name
    TO <database_principal> [ ,...n ]
    [ CASCADE ]
    [ AS <database_principal> ]

<database_principal> ::= Database_user
    | Database_role
    | Database_user_with_no_login

```

Base de datos de SQL Azure **no admite** las siguientes opciones con esta instrucción DENY:

- Opciones de <database\_principal>:
  - Application\_role
  - Database\_user\_mapped\_to\_Windows\_User
  - Database\_user\_mapped\_to\_Windows\_Group
  - Database\_user\_mapped\_to\_certificate
  - Database\_user\_mapped\_to\_asymmetric\_key

## EXECUTE

Ejecuta una cadena de comandos o una cadena de caracteres dentro de un proceso por lotes de Transact-SQL, o uno de los siguientes módulos: procedimiento almacenado del sistema, procedimiento almacenado definido por el usuario, función escalar definida por el usuario o procedimiento almacenado extendido.

```

Execute a stored procedure or function
[ { EXEC | EXECUTE } ]
{
    [ @return_status= ]
    { module_name | @module_name_var }
    [ [ @parameter= ] { value
        | @variable [ OUTPUT ]
        | [ DEFAULT ]
    }
    ]
    [ ,...n ]
    [ WITH RECOMPILE ]
}

```

```
[;]
```

Execute a character string

```
{ EXEC | EXECUTE }
  ( { @string_variable | [ N ] 'tsql_string' } [ + ...n ] )
  [ AS { USER } = 'name' ]
[;]
```

Base de datos de SQL Azure **no admite** los siguientes argumentos y opciones cuando se utiliza la instrucción EXECUTE:

- Opción **Number**.
- **EXECUTE AS LOGIN**.
- Ejecuta un comando de paso a través en un servidor vinculado.

## EXECUTE AS

Esta instrucción establece el contexto de ejecución de una sesión. Su sintaxis es la siguiente:

```
{ EXEC | EXECUTE } AS <context_specification>
[;]

<context_specification> ::=
{ USER } = 'name'
  [ WITH { NO REVERT | COOKIE INTO @varbinary_variable } ]
| CALLER
```

Base de datos de SQL Azure **no admite** la opción EXECUTE AS LOGIN cuando se utiliza la instrucción EXECUTE AS.

## EXECUTE AS Clause

En Microsoft Base de datos de SQL Azure puede definir el contexto de ejecución de los siguientes módulos definidos por el usuario: funciones (excepto funciones con valores de tabla insertados), procedimientos y desencadenadores. La sintaxis es la siguiente:

```
Functions (except inline table-valued functions), Stored
Procedures, and DML Triggers
```

```
{ EXEC | EXECUTE } AS { CALLER | SELF | OWNER | 'user_name' }
```

DDL Triggers with Database Scope

```
{ EXEC | EXECUTE } AS { CALLER | SELF | 'user_name' }
```

Cuando se usa la instrucción EXECUTE AS Clause, Base de datos de SQL Azure **no admite**:

- Ejecutar desencadenadores DDL con ámbito de servidor y desencadenadores de inicio de sesión.
- Ejecutar colas.

## ENABLE TRIGGER

Esta instrucción habilita un desencadenador del lenguaje de manipulación de datos (DML), un desencadenador del lenguaje de definición de datos (DDL) o un desencadenador de inicio de sesión. Su sintaxis es la siguiente:

```
ENABLE TRIGGER { [ schema_name . ] trigger_name [ ,...n ] | ALL }
ON { object_name | DATABASE } [ ; ]
```

Base de datos de SQL Azure **no admite** el argumento ALL SERVER con la instrucción ENABLE TRIGGER.

## DISABLE TRIGGER

Esta instrucción deshabilita un desencadenador. Su sintaxis es la siguiente:

```
DISABLE TRIGGER { [ schema_name . ] trigger_name [ ,...n ] | ALL }
ON { object_name | DATABASE } [ ; ]
```

Base de datos de SQL Azure **no admite** el argumento ALL SERVER cuando se utiliza la instrucción DISABLE TRIGGER.

## USE (Base de datos de SQL Azure)

Esta instrucción mantiene el contexto de base de datos como la base de datos actual. Su sintaxis es la siguiente:

```
USE { database }
```

En Microsoft Base de datos de SQL Azure, el parámetro *database* solo puede hacer referencia a la base de datos actual. La instrucción USE no cambia de base de datos. Para cambiar las bases de datos, debe conectarse directamente a la base de datos.

## **LENGUAJE TCL DE CONTROL DE TRANSACCIONES EN BASE DATOS DE SQL AZURE**

Las instrucciones del lenguaje de control de transacciones (TCL) de Transact-SQL (BEGIN TRANSACTION, COMMIT TRANSACTION, COMMIT WORK, ROLLBACK TRANSACTION, etc.) son admisibles sin acotaciones en SQL Azure. Su sintaxis en SQL Azure será la misma que en Transact-SQL.

# ÍNDICE ALFABÉTICO

## A

Acceso a datos de SQL Azure.....	5
Acceso a los datos de facturación y uso .....	15
Administración a nivel de servidor y la base de datos maestra.....	19
Administración de la seguridad de SQL Azure .....	17
Administración en SQL AZURE.....	17
Administración lógica frente a administración física.....	3
Administrar inicios de sesión .....	19
ALTER AUTHORIZATION .....	157
ALTER DATABASE.....	158
ALTER FUNCTION .....	161
ALTER INDEX.....	164
ALTER LOGIN .....	165
ALTER PROCEDURE.....	166
ALTER TABLE.....	167
ALTER TRIGGER .....	170
ALTER USER .....	171
ALTER VIEW .....	172
Ancho de banda en la plataforma WIndows Azure	14
Aplicaciones de capa de datos .....	70
Aprovisionamiento.....	4
Arquitectura de SQL Azure .....	7
Asistente de migración de SQL Server (SSMA).....	73
Asistente para generar y publicar scripts .....	71
Autenticación .....	82
<b>B</b>	
Bases de datos.....	11
Buscar las N consultas superiores.....	51
<b>C</b>	
Calcular el tamaño de la base de datos.....	50
Características de SQL Server 2005 no admitidas por Base de datos SQL Azure .....	86
Características de SQL Server 2008 no admitidas por Base de datos SQL Azure .....	85
Características de SQL Server 2008 R2 no admitidas por Base de datos SQL Azure .....	85
Características de Transact-SQL compatibles y parcialmente compatibles .....	67
Características de Transact-SQL no admitidas.....	68
Características y tipos .....	4
Cifrado y validación de certificados.....	81
Coherencia transaccional .....	48
Compatibilidad con controladores, bibliotecas y protocolos.....	74
Compatibilidad con herramientas y tecnologías ..	75
Compatibilidad con instrucciones SET .....	139
Compatibilidad con la intercalación de SQL Server.	78
Compatibilidad con la migración de datos.....	77
Compatibilidad con métodos del tipo de datos geography .....	116
Compatibilidad con métodos del tipo de datos geometry .....	117
Compatibilidad con métodos del tipo de datos hierarchyid.....	118
Compatibilidad con ODBC .....	76
Compatibilidad con sugerencias .....	138
Compatibilidad con Transact-SQL.....	4
Compatibilidad con Transact-SQL (Base de datos de SQL Azure).....	66
Compatibilidad con Visual Studio.....	75
Compatibilidad DE Base de datos SQL Azure con herramientas y utilidades de SQL Server.....	69
Compilar y hospedar aplicaciones SQL Azure.....	65

Conceder acceso a base de datos a un inicio de sesión .....	22
Conceder permisos de nivel de servidor a un inicio de sesión .....	20
Conectar con SQL Azure mediante ADO.NET .....	92
Conectar con SQL Azure mediante ADO.NET Entity Framework .....	111
Conectar con SQL Azure mediante ASP.NET .....	97
Conectar con SQL Azure mediante JDBC .....	109
Conectar con SQL Azure mediante PHP .....	105
Conectar con SQL Azure mediante Servicios de datos de WCF (ADO.NET) .....	99
Conectar con SQL Azure mediante sqlcmd .....	91
Conectarse de Windows Azure .....	54
Conectarse desde Internet .....	54
CONFIGURAR EL FIREWALL DE SQL AZURE .....	56
Copia en el mismo servidor .....	42
Copia entre servidores .....	44
Copiar bases de datos en SQL Azure .....	41
Copiar una base de datos .....	42
Crear bases de datos SQL Azure .....	64
Crear el modelo de datos .....	100, 111
Crear el script de TRANSACT-SQL .....	25
Crear el servicio de datos .....	101
Crear la aplicación cliente .....	102
Crear la aplicación de consola .....	112
Crear la aplicación PHP .....	106
Crear la aplicación Windows Azure .....	105
Crear la base de datos de ejemplo SCHOOL .....	24
Crear la base de datos TESTDB .....	97
Crear la primera regla de firewall .....	55
Crear servidores SQL Azure .....	63
CREATE DATABASE .....	172
CREATE FUNCTION .....	176
CREATE INDEX .....	179
CREATE LOGIN .....	180
CREATE PROCEDURE .....	181
CREATE SPATIAL INDEX .....	182
CREATE SYNONYM .....	183
CREATE TABLE .....	184
CREATE TRIGGER .....	187
CREATE TYPE .....	188
CREATE USER .....	189
CREATE VIEW .....	190

Cuentas y facturación en SQL Azure .....	12
--	----

## D

Definición de Transact-SQL de la base de datos School .....	26
DENY (permisos de base de datos de base de datos de SQL Azure) .....	199
DENY (Permisos de entidad de seguridad de base de datos de base de datos de SQL Azure) .....	200
DENY (Permisos de tipo de base de datos de SQL Azure) .....	200
Desarrollar aplicaciones SQL Azure .....	65
Desarrollo en Base de datos de SQL Azure .....	63
Descodificar códigos de motivo .....	153
Detener el proceso de copia .....	48
DISABLE TRIGGER .....	203
DROP DATABASE .....	191
DROP INDEX .....	191
DROP TABLE .....	192
DROP TRIGGER .....	192

## E

Ediciones de SQL Azure .....	12
Ejecutar el script en SQL Azure .....	25
Ejemplo .....	43, 45, 58, 109
Ejemplo de aplicación de consola .....	92
El administrador de base de datos para SQL Azure .....	70
El nivel de cliente .....	7
El nivel de plataforma .....	9
El nivel de servicios .....	9
El nuevo portal de administración .....	1
El tipo de datos XML .....	118
ENABLE TRIGGER .....	203
Errores de copia de base de datos .....	149
Errores de pérdida de conexión .....	151
Errores generales .....	148
Excepciones de smo en Base de datos de SQL Azure .....	153
EXECUTE .....	201
EXECUTE AS .....	202
EXECUTE AS CLAUSE .....	202

## F

Facturación .....	13
-------------------	----

Firewall..... 81  
 Firewall de SQL Azure ..... 53  
 Funciones de agregado ..... 120  
 Funciones de Base de datos de SQL Azure ..... 119  
 Funciones de categoría ..... 120  
 Funciones de conjuntos de filas..... 120  
 Funciones escalares ..... 120  
 Funciones escalares de ODBC ..... 126

**G**

GRANT (permisos de Base de datos de base de datos de SQL Azure) ..... 195, 196  
 GRANT (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)..... 195

**H**

Herramienta de administración de base de datos, 70  
 Hospedar su aplicación en la nube..... 6  
 Hospedar sus datos en la nube..... 6

**I**

Información general ..... 53  
 información general de SQL Azure ..... 2  
 Información general sobre el modelo de aprovisionamiento de SQL Azure ..... 11  
 Iniciar el proceso de copia ..... 47  
 Inicios de sesión ..... 11  
 Inicios de sesión y usuarios ..... 82  
 Instrucciones para conectar con Base de datos SQL Azure ..... 87, 89  
 Instrucciones Transact-SQL admitidas de forma parcial en Base de datos de SQL Azure) ..... 155  
 Instrucciones Transact-SQL admitidas en Base de datos de SQL Azure) ..... 137  
 Instrucciones Transact-SQL no admitidas en Base de datos de SQL Azure) ..... 140  
 Instrucciones y limitaciones de seguridad en Base de datos de SQL Azure..... 80  
 Instrucciones y limitaciones generales (Base de datos SQL Azure)..... 74  
 Intervalos ..... 47  
 Introducción..... 10

**L**

La base de datos maestra..... 55  
 La utilidad bcp..... 72  
 La utilidad sqlcmd..... 69  
 Lenguaje DDL. Instrucciones ALTER en Base de datos de SQL Azure ..... 157  
 Lenguaje DDL. Instrucciones CREATE en Base de datos de SQL Azure ..... 172  
 Lenguaje de control de datos DCL en Base datos de SQL Azure ..... 194  
 Lenguaje de manipulación de datos DML. Instrucción INSERT en Base datos de SQL Azure ..... 193  
 Lenguaje DML. Instrucciones DROP en Base de datos de SQL Azure ..... 190  
 Lenguaje tcl de control de transacciones en Base datos de SQL Azure ..... 204  
 Limitación de SQL Azure y descodificación de códigos de motivo..... 152  
 Limitaciones de las características de SQL Server en Base de datos SQL Azure ..... 84  
 Limitar el tamaño de la base de datos con el valor MAXSIZE ..... 15

**M**

Microsoft Access..... 73  
 Migrar bases de datos a SQL Azure ..... 23  
 Migrar la aplicación a SQL Azure ..... 104, 113  
 Migrar una base de datos a SQL Azure mediante el Asistente para generar scripts ..... 24  
 Modelo de aprovisionamiento de SQL Azure ..... 9  
 Mover datos a SQL Azure ..... 40

**N**

Nivel de infraestructura ..... 9  
 Niveles de aislamiento basado en versiones de fila 78  
 Número de bases de datos y límites de tamaño... 79

**O**

Objetos de administración de SQL Server (SMO).. 72  
 Operadores aritméticos ..... 127  
 Operadores bit a bit ..... 128  
 Operadores compuestos..... 127  
 Operadores de comparación..... 129  
 Operadores en Base de datos de SQL Azure ..... 127

<i>Operadores lógicos</i> .....	128
<i>Operadores unarios</i> .....	129
<i>Otras características de SQL Server2 no admitidas por Base de datos SQL Azure</i> .....	86
<i>Otros operadores misceláneos</i> .....	130

**P**

<i>Permisos</i> .....	46
<i>PowerPivot para Excel</i> .....	73
<i>Preparar el entorno de desarrollo</i> .....	109
<i>Procedimientos almacenados de Base de datos SQL Azure</i> .....	145
<i>Procedimientos almacenados de seguridad</i> .....	147
<i>Procedimientos almacenados del catálogo</i> .....	145
<i>Procedimientos almacenados del motor de base de datos</i> .....	146
<i>Procedimientos almacenados del sistema en Base de datos de SQL Azure</i> .....	144
<i>Procedimientos recomendados</i> .....	84

**R**

<i>Referencia de errores y excepciones en Base de datos de SQL Azure</i> .....	148
<i>Requisito de índices clúster</i> .....	78
<i>Requisitos de nomenclatura</i> .....	79
<i>Restricciones de conexión</i> .....	80
<i>REVOKE (permisos de base de datos de Base de datos de SQL Azure)</i> .....	197
<i>REVOKE (permisos de entidad de seguridad de base de datos de Base de datos de SQL Azure)</i> .....	198
<i>REVOKE (permisos de tipo de Base de datos de SQL Azure)</i> .....	198

**S**

<i>Servidores</i> .....	11
<i>Similitudes y diferencias con SQL Server</i> .....	2

<i>Solucionar problemas de SQL Azure</i> .....	59
<i>Solucionar problemas del firewall</i> .....	55
<i>SQL Azure Reporting</i> .....	72
<i>SQL Server Analysis Services y SQL Server Reporting Services</i> .....	72
<i>SQL Server Integration Services</i> .....	71
<i>SQL Server Management Studio</i> .....	70
<i>Supervisar consultas bloqueadas</i> .....	52
<i>Supervisar el rendimiento de las consultas</i> .....	51
<i>Supervisar las conexiones</i> .....	51
<i>Supervisar planes de consulta</i> .....	52
<i>Supervisar SQL Azure mediante vistas de administración dinámica</i> .....	49
<i>Supervisar una copia</i> .....	48
<i>Suscripciones de SQL Azure</i> .....	14

**T**

<i>Tablas del sistema de Base de datos de SQL Azure</i> .....	130
<i>Tener acceso al servicio de datos School desde el rol web de Windows Azure</i> .....	103
<i>Tipos de datos en Base de datos de SQL Azure</i> .....	115
<i>Trabajos y Agente SQL Server</i> .....	77

**U**

<i>USE (Base de datos de SQL Azure)</i> .....	203
<i>Utilizar nuevos inicios de sesión</i> .....	20

**V**

<i>Ver inicios de sesión y bases de datos</i> .....	23
<i>Vistas de administración dinámica</i> .....	134
<i>Vistas de bases de datos de SQL Azure</i> .....	131
<i>Vistas de catálogo</i> .....	132
<i>Vistas de compatibilidad</i> .....	134
<i>Vistas del esquema de información</i> .....	136
<i>Vistas del sistema de Base de datos de SQL Azure</i> .....	131