



Introducción a la base de datos SQL Azure

Microsoft SQL Azure es una base de datos relacional en la nube basada en tecnologías de SQL Server. Mediante el uso de SQL Azure, puede fácilmente suministrar e implementar soluciones de base de datos relacionales a la nube, y tomar ventaja de un centro de datos distribuido que proporciona disponibilidad de clase empresarial, escalabilidad y seguridad con los beneficios de protección de datos y auto corrección inter construidos.

Beneficios clave del servicio

Los beneficios de utilizar de SQL Azure son múltiples. Estos incluyen la capacidad de administración, alta disponibilidad, escalabilidad, un modelo de desarrollo familiar, y un modelo de datos relacional.

Manejabilidad

SQL Azure ofrece la escalabilidad y la funcionalidad de un centro de datos de la empresa sin los gastos administrativos asociados a implementaciones locales de SQL Server.

Alta Disponibilidad

SQL Azure se basa en las tecnologías de servidor Windows Server y SQL, y es lo suficientemente flexible para hacer frente a cualquier variación en el uso y la carga. El servicio replica múltiples copias redundantes de los datos a múltiples servidores físicos para mantener la disponibilidad de los datos y la continuidad de su negocio. En caso de que el hardware falle, SQL Azure provee un failover automático para optimizar la disponibilidad para su aplicación.

Escalabilidad

Una ventaja clave de SQL Azure es la facilidad con la que usted puede ampliar su solución. Después de particionar sus datos, el servicio escala a medida que sus datos crecen. Un modelo de precios "pay-as-you-grow" asegura de que usted sólo paga por el almacenamiento que se utiliza, por lo que también puede reducir el servicio cuando no lo necesita.

Modelo de Desarrollo Familiar

Cuando los desarrolladores crean aplicaciones bajo SQL Server, usan las bibliotecas cliente que utilizan el protocolo de flujo de datos tabular (TDS) para la comunicación entre cliente y servidor. SQL Azure proporciona la misma interfaz de TDS de SQL Server para que pueda utilizar las mismas herramientas y bibliotecas para crear aplicaciones cliente para datos almacenados en SQL Azure.

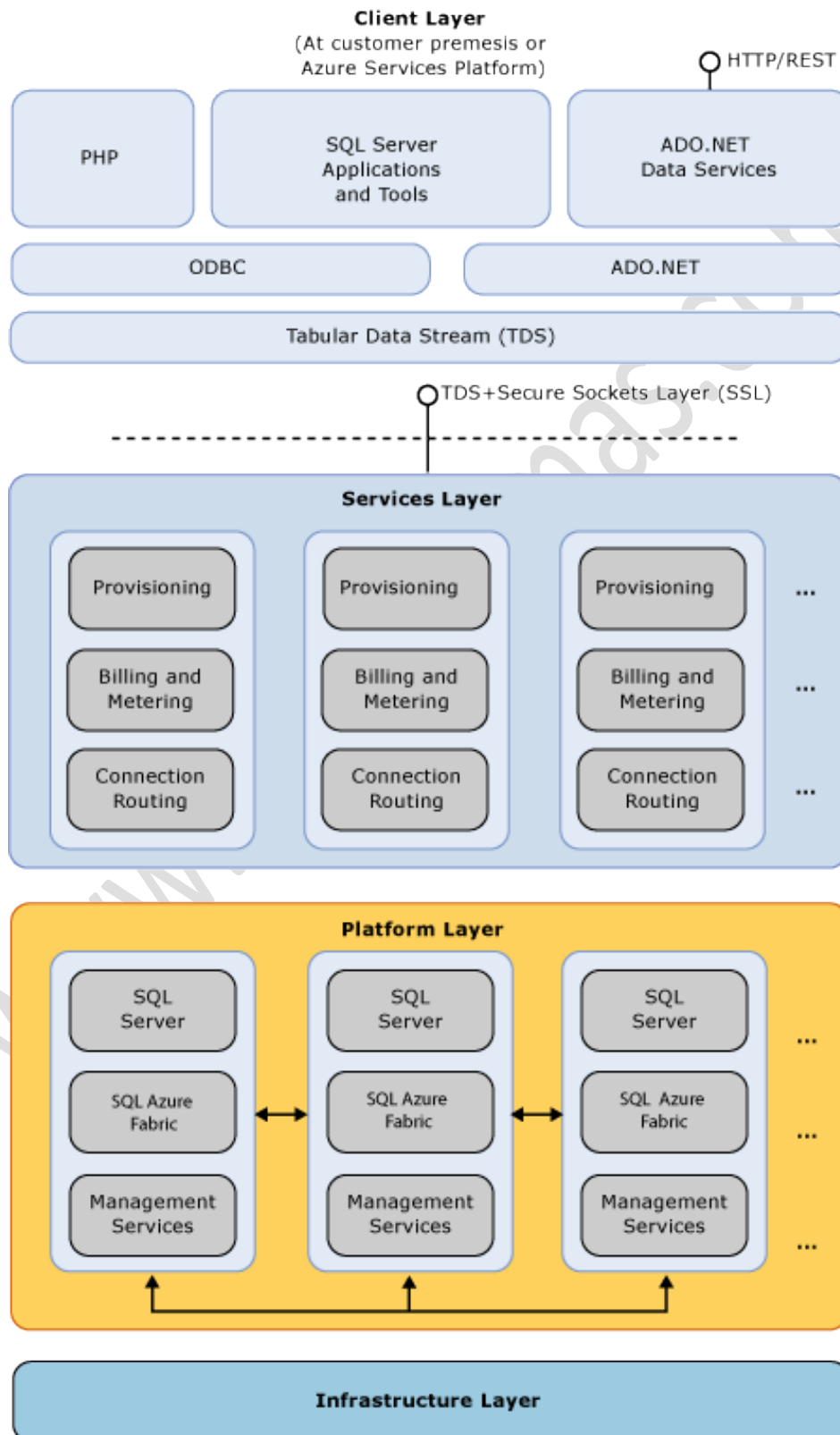
Modelo Relacional de Datos

SQL Azure parecerá muy familiar para los desarrolladores y administradores, porque los datos son almacenados en SQL Azure como en SQL Server, mediante Transact-SQL. Conceptualmente similar a una instancia local de SQL Server, un servidor SQL Server Azure es el grupo lógico de bases de datos que actúa como un límite de autorización. Dentro de cada servidor SQL Azure, puede crear varias bases de datos que tienen tablas, vistas, procedimientos almacenados, índices, y otros objetos familiares a una base de datos. Este modelo de datos hace un buen uso del diseño existente de su base de datos relacional y habilidades de programación de Transact-SQL, y simplifica el proceso de migración de aplicaciones existentes en la base de datos locales SQL hacia Azure.

Los servidores Azure SQL y sus bases de datos son objetos virtuales que no corresponden a los servidores físicos y bases de datos. Aislándolo a usted de la implementación física, SQL Azure le permite dedicar más tiempo a su diseño de base de datos.

Arquitectura de SQL Azure

La Base de datos de Microsoft SQL Azure se encuentra alojado en servidores que ejecutan tecnologías SQL Server en centros de datos de Microsoft. Desde una perspectiva arquitectónica, hay cuatro niveles distintos de abstracción que trabajan juntos para proporcionar una base de datos relacional para que utilice su aplicación: la capa de cliente, la capa de servicios, la capa de la plataforma, y la capa de infraestructura. Como se observa en el siguiente diagrama, SQL Azure trabaja con aplicaciones de terceros, de código abierto, y muchas aplicaciones familiares Microsoft, tales como servicios de datos de ADO.NET, ODBC y ADO.NET.



La capa cliente

La capa cliente es la que reside más cerca de su aplicación, y es utilizada por su aplicación para comunicarse directamente con SQL Azure. La capa cliente puede residir localmente en su centro de datos o estar alojado en Windows Azure. Dado que SQL Azure ofrece el mismo sistema de datos tabular (TDS) que SQL Server, puede utilizar herramientas y bibliotecas conocidas para crear aplicaciones cliente para datos que están en la nube.

Hay muchas opciones disponibles para que su aplicación pueda comunicarse con SQL Azure. Los siguientes pueden ser utilizados en la capa cliente:

- PHP
- La API REST, disponibles a través de los servicios de datos de ADO.NET
- Las aplicaciones y herramientas de SQL Server
- ODBC o ADO.NET

La capa de Servicios

La capa de servicios funciona como una puerta de enlace entre la capa de cliente y la capa de la plataforma, donde residen los datos. La capa de servicios proporciona tres funciones: aprovisionamiento, facturación y medición, y enrutamiento de conexión.

La capa de servicios proporciona la base de datos que especifique con su cuenta de Azure. La facturación y medición de los aspectos de la capa de servicios permite soporte multi-arrendatario facilitando el seguimiento y la facturación para el uso de bases de datos basado en cuentas individuales de Azure. SQL Azure está construido sobre una plataforma escalable que involucra numerosos servidores físicos, esta capa se ocupa de todas las conexiones de enrutamiento entre su aplicación y los servidores físicos, donde residen sus datos.

La capa de la Plataforma

La capa de la plataforma incluye a los servidores físicos y los servicios que apoyan la capa de servicios. La capa de plataforma consta de muchas instancias de SQL Server, cada uno de ellos es manejado por el fabric de Azure SQL.

El fabric de SQL Azure es un sistema de computación distribuida compuesta por redes bien integrados, servidores y almacenamiento. Provee failover automático, balance de carga, y la replicación automática entre los servidores físicos.

Servicios de administración que vigilan la salud de los servidores individuales y permiten la instalación automática de actualizaciones de servicios y parches de software.

Capa de Infraestructura

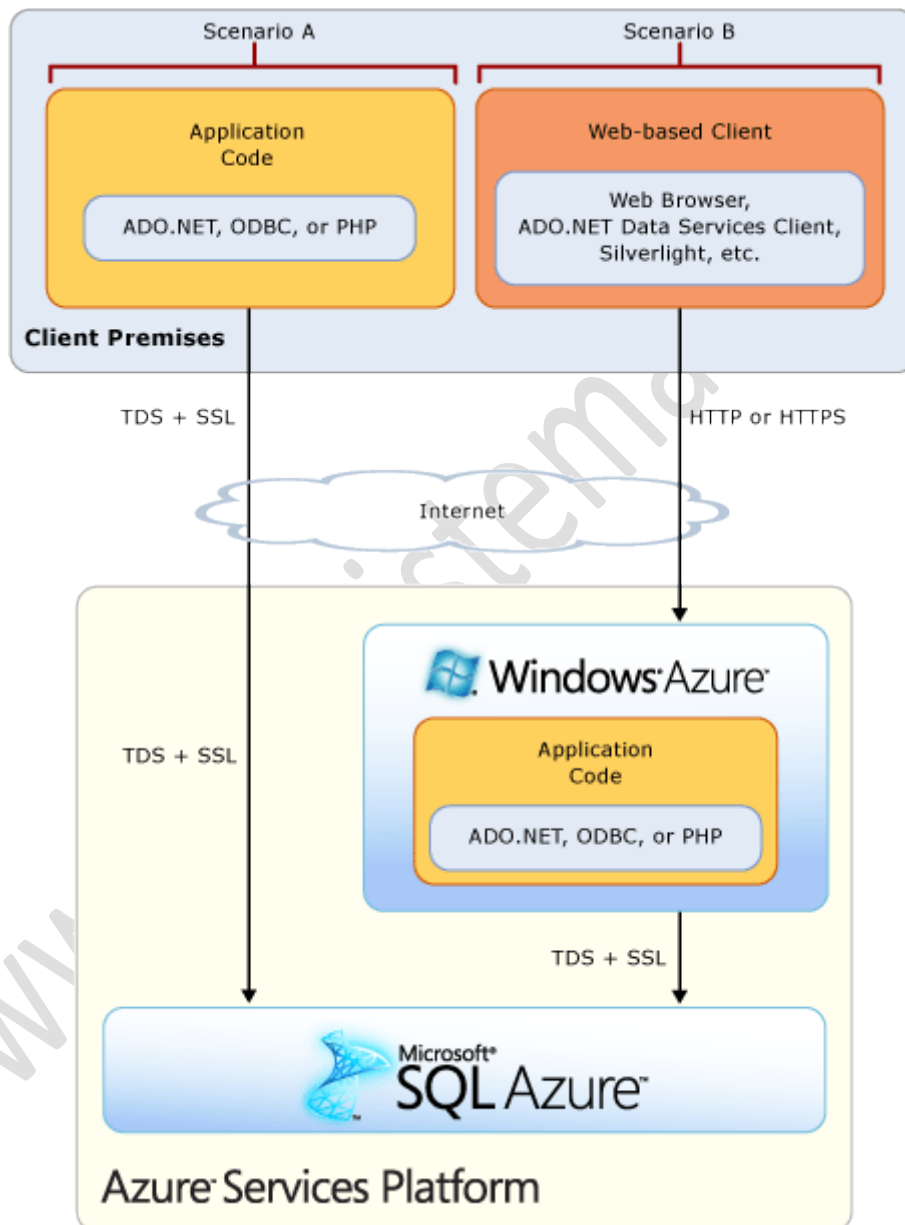
La capa de infraestructura representa la administración IT del hardware físico y los sistemas operativos que soportan la capa de servicios

Capa de acceso a datos en SQL Azure

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

Acceso a datos en SQL Azure

En una aplicación tradicional, el código de aplicación y base de datos se encuentran en el mismo centro físico de datos. SQL Azure y la Plataforma de Servicios Azure ofrece muchas alternativas para esa arquitectura. El diagrama siguiente muestra dos alternativas disponibles para la forma en que su aplicación pueda acceder a datos con SQL Azure.



En el escenario A en la izquierda, el código de la aplicación sigue estando en las instalaciones de su centro de datos corporativos, pero la base de datos reside en SQL Azure. Su código de la aplicación utiliza las bibliotecas de cliente, como ADO.NET, ODBC, o PHP para acceder a su base de datos en SQL Azure. Independientemente de la biblioteca cliente elegida, los datos se transfieren por medio de secuencia de datos tabular (TDS) sobre una capa de sockets seguros (SSL). En el escenario B de la derecha, el código de aplicación se encuentra alojado en Windows Azure y su base de datos reside en SQL Azure. Su aplicación puede usar las mismas bibliotecas cliente para acceder a su base de datos en SQL Azure. Hay muchos tipos diferentes de aplicaciones que puede alojar en la Plataforma de Servicios Azure.

Almacenando sus datos en la nube

El aspecto y la funcionalidad de su base de datos de SQL Server no se modifica cuando se utiliza SQL Azure en lugar de una instancia local de SQL Server. Hay algunas diferencias de menor importancia, que incluyen cambios de Transact-SQL y gestión de acceso. Cuando aloja solamente sus datos, en el centro de datos de Microsoft, es importante tener en cuenta la latencia de red entre SQL Azure y sus servidores locales corriendo el código de su aplicación. La latencia de red a SQL Azure es mayor de lo que sería si la base de datos también existiera localmente y es necesario tener en cuenta esto en el código de su aplicación.

Una forma de minimizar la latencia de red en los accesos a la base de datos es, además, ejecutar su aplicación en la Plataforma de Servicios Azure.

El alojamiento de su aplicación en la nube

El Alojamiento de aplicaciones en Windows Azure es opcional. Teniendo en cuenta sus necesidades, usted puede migrar sus aplicaciones hacia o desde Windows Azure. El alojamiento de su aplicación en Windows Azure es beneficioso para el rendimiento de su aplicación, ya que minimiza la latencia de la red asociada a las solicitudes de datos de la aplicación para SQL Azure.

Por ejemplo, usted puede decidir crear una interfaz de usuario basada en Web en Windows Azure para su aplicación de base de datos que se aloja en SQL Azure. En este escenario, el código de cliente y los datos se alojan en el mismo centro de datos físicos, de modo que la latencia de la red es mínima entre los servidores correspondientes. Los clientes todavía tendrán que recorrer la Internet para llegar a su base de datos, pero el tráfico de red entre el código de cliente y los datos será mucho más rápido porque las numerosas peticiones de datos estarán a una distancia mucho más corta.

Directivas y limitaciones (base de datos SQL Azure)

Este tema proporciona información acerca de las directrices y limitaciones específicas cuando se utiliza SQL Azure.

Herramientas y soporte de drivers

En esta versión, el soporte de drivers y herramientas de SQL SERVER está todavía en fase de desarrollo.

Al escribir aplicaciones para SQL Azure, puede utilizar el proveedor de datos de ADO.NET para SQL Server, SQL Server 2008 Native Client - ODBC Driver y el SQL Server 2008 Driver for PHP version 1.1.

Además, puede utilizar la utilidad sqlcmd para entrar en Transact-SQL, los procedimientos del sistema y los archivos de secuencias de comandos en el símbolo del sistema. No se puede conectar a SQL Azure mediante OLE DB.

Nota:

En esta versión, SQL Server Management Studio (SSMS) proporciona una funcionalidad limitada para SDS y no es admitido actualmente.

Autenticación

SQL Azure sólo admite autenticación de SQL Server. Los usuarios deben proporcionar las credenciales (usuario y contraseña) cada vez que se conecten a SQL Azure.

Conexiones y Usuarios

Cuando la gestión de los inicios de sesión y los usuarios en SQL Azure, las siguientes restricciones:

El servidor de inicio de sesión inicial de la categoría principal, creado durante el aprovisionamiento de servidores es equivalente a la sesión de sa en una instancia de SQL Server. Sólo la contraseña asociada a este login principal a nivel de servidor puede ser cambiado. Los miembros de la función de servidor sds_securityadmin no pueden cambiar esta contraseña. Para acceder a la base de datos master, cada usuario debe estar asignado a una cuenta de usuario en la base de datos master. El usuario de base de datos en la base de datos principal asociado con este login principal a nivel de servidor no puede ser alterado o quitado. Aunque el usuario de base de datos en la base de datos principal asociada con este login principal nivel de servidor no es miembro de las dos funciones sds_dbcreator y sds_securityadmin, este usuario de base de datos tiene todos los permisos concedidos a estas dos funciones.

Para un inicio de sesión del servidor, la master, es la base de datos predeterminada, y Estados Unidos-Inglés el idioma predeterminado.

Usted debe estar conectado a la base de datos master durante la ejecución del CREATE / ALTER / DROP LOGIN y CREATE / DROP DATABASE.

Cuando se ejecuta una sentencia CREATE / ALTER / DROP LOGIN y CREATE / DROP DATABASE en una aplicación de ADO.NET, no está permitido utilizar los comandos con parámetros. Cada una de estas declaraciones debe ser la única instrucción en un lote de SQL.

Al ejecutar la instrucción CREATE USER con la opción FOR / FROM LOGIN, debe ser la única instrucción en un lote de SQL.

Al ejecutar la instrucción ALTER USER con la opción WITH LOGIN, debe ser la única instrucción SQL en un lote.

Sólo el login a nivel de servidor y los miembros de la función sds_dbcreator tiene permiso para ejecutar las sentencias CREATE DATABASE, y DROP DATABASE.

Sólo el login a nivel de servidor y los miembros de la función sds_securityadmin tiene permiso para ejecutar el CREATE LOGIN, ALTER LOGIN, y DROP LOGIN.

Para acceder a la base de datos master, el usuario debe estar asignado a una cuenta de usuario en la base de datos master.

Limitaciones de conexión

A fin de ofrecer una buena experiencia para todos los clientes de SQL Azure, su conexión con el servicio puede ser cerrada debido a las condiciones siguientes:

El uso excesivo de recursos

Consultas de ejecución prolongada

Transacciones individuales de larga duración, entre las sentencias de BEGIN y END TRAN

Conexiones inactivas

Nota:

La duración máxima permitida están sujetos a cambio. En el momento de escribir esto, las conexiones inactivas y las operaciones de larga duración se cierran después de cinco minutos.

Se recomienda que parta las transacciones grandes en otras más pequeñas e incorpore código de reintento para adaptarse a estas situaciones.

Esquema de diseño

Dado que SQL Azure no es compatible con las heap tables, una tabla debe tener un índice agrupado. Si se crea una tabla sin una limitación en clúster (clustered constraint), un índice agrupado se deben crear antes de que una operación de inserción esté permitida en la tabla.

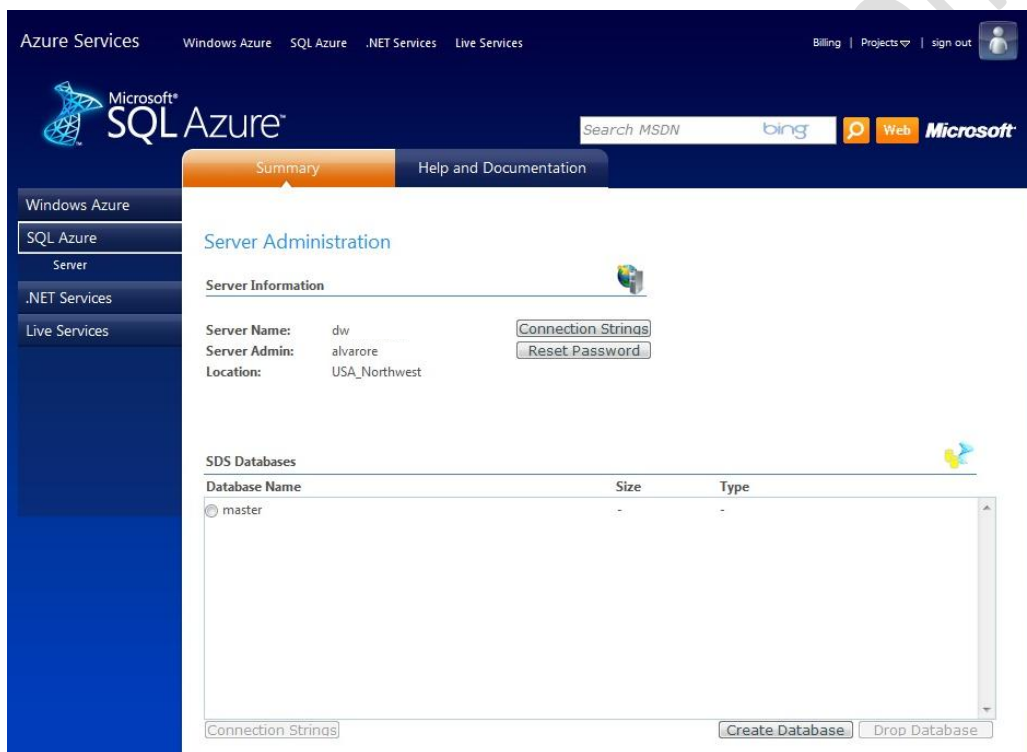
Modelo de aprovisionamiento de SQL Azure

El proceso de aprovisionamiento Azure Microsoft SQL prepara y configura el hardware y el software necesario para mantener sus datos en un centro de datos de Microsoft. El modelo de aprovisionamiento de Azure SQL describe la jerarquía lógica de su cuenta de Servicios Azure, servidores Azure SQL y bases de datos, le provee almacenamiento de datos con rapidez.

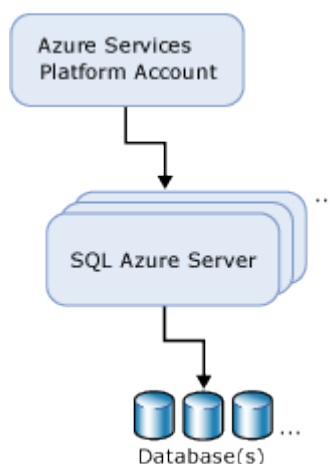
Cómo empezar

Para utilizar SQL Azure, primero debe crear una nueva cuenta de Servicios Azure, que le permite acceder a todos los servicios, como Windows Azure, .NET Services, y SQL Azure. La cuenta de Servicios Azure se utiliza para crear y gestionar sus suscripciones y para facturar el consumo de los servicios de Azure.

Una vez que la cuenta es creada, usted puede utilizar el portal de SQL Azure para crear un servidor SQL Server Azure. Para disponer de servidores SQL Azure adicional, debe crear suscripciones adicionales. El portal de SQL Azure proporciona una interfaz de usuario que puede utilizar para aprovisionar servidores y conexiones, y para crear rápidamente las bases de datos.



El diagrama siguiente muestra la relación entre una cuenta de Servicios Azure, los servidores de Azure SQL y las bases de datos



Como se muestra en el diagrama, cada cuenta de Servicios Azure pueden estar asociados con múltiples servidores SQL Azure (con una suscripción diferente para cada uno). Cada servidor SQL Azure puede asociarse con una o más bases de datos.

Servidores

Un servidor SQL Azure es un grupo lógico de bases de datos y actúa como un punto central de administración de bases de datos múltiples. Cada servidor SQL Azure incluye conexiones similares a las de las instancias de SQL Server en sus instalaciones.

Cada servidor SQL Azure tiene un nombre de dominio completo único, que se produce durante el proceso de aprovisionamiento de Azure SQL. En el siguiente ejemplo, **nombreDeServidor** hace referencia al nombre del servidor SQL Azure: **nombreDeservidor.ctp.database.windows.net**

Cada servidor SQL Azure puede contener múltiples bases de datos. En cada base de datos, puede crear tablas, vistas, índices, procedimientos almacenados, y otros objetos familiares de una base de datos. Con el fin de crear una nueva base de datos, puede utilizar el portal de Azure o la sentencia SQL CREATE DATABASE.

El proceso de aprovisionamiento crea una base de datos **master** de forma automática. La base de datos master mantiene una lista de bases de datos y mantiene un registro de inicios de sesión, los inicios de sesión y que tienen permiso para crear bases de datos o de otros inicios de sesión. (Esta base de datos master es equivalente a la base de datos master en una nueva instancia local de SQL Server.) La base de datos master de SQL también proporciona las mediciones del uso Azure que Ud. puede ver.

Inicios de sesión

Durante el proceso de aprovisionamiento, SQL Azure crea un inicio de sesión para usted que es la cuenta principal a nivel de servidor de SQL Azure. Esta cuenta principal es equivalente a la sesión de **sa** en SQL Server. Bases de datos e inicios de sesión adicionales pueden ser creados en el servidor, según sea necesario.

La administración para inicios de sesión y base de datos en SQL Azure difiere de una instancia local de SQL Server.

SQL Azure ofrece el mismo conjunto de entidades de seguridad que están disponibles en SQL Server, tales como:

- Los inicios de sesión de SQL Server: Se utiliza para autenticar el acceso a SQL Azure a nivel de servidor.
- Los usuarios de base de datos: Se utiliza para conceder el acceso a SQL Azure a nivel de base de datos.
- Roles de base de datos: Se utiliza para agrupar usuarios y autorizar el acceso a SQL Azure a nivel de base de datos.

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

Administrar inicios de sesión y usuarios en SQL Azure

En Microsoft SQL Azure, cuando se registra para el servicio, el proceso de aprovisionamiento crea un servidor SQL Azure, una base de datos llamada master, y un inicio de sesión que es la cuenta principal de nivel de servidor de SQL Azure. Ese inicio de sesión es el equivalente de SQL Azure a la cuenta **sa** de una instancia local de SQL Server.

La cuenta principal a nivel de servidor de SQL Azure viene con los roles de seguridad necesarios para gestionar toda la seguridad a nivel de servidor y base de datos. En este tema se describe cómo puede utilizar la cuenta principal a nivel de servidor y otras cuentas para administrar inicios de sesión y bases de datos en SQL Azure.

Administración de la Seguridad

La administración de seguridad en SQL Azure es similar a la administración de seguridad de una instancia local de SQL Server. La gestión de la seguridad a nivel de la base de datos es casi idéntica, sólo con diferencias en los parámetros disponibles. Debido a que las bases de datos SQL Azure pueden escalar a uno o más equipos físicos, SQL Azure utiliza una estrategia diferente para la administración a nivel de servidor. La siguiente tabla resume las diferencias entre la administración de seguridad local de SQL Server y SQL Azure

Punto de Diferencia	SQL Server local	SDS
Donde maneja la seguridad a nivel de servidor	La carpeta Seguridad en el explorador de objetos de SQL Server Management Studio	master database
Roles de seguridad a nivel de servidor para crear inicios de sesión	securityadmin	sds_securityadmin
Comandos para manejar inicios de sesión	CREATE LOGIN ALTER LOGIN DROP LOGIN	CREATE LOGIN ALTER LOGIN DROP LOGIN (hay limitaciones de parametros y debe estar conectado a la master database)
Vista que muestra todos los inicios de sesión	sys.sql_logins	sys.sql_logins (debe estar conectado a la master database)
Rol a nivel de servidor para crear base de datos	dbcreator	sds_dbcreator
Comando para crear base de datos	CREATE DATABASE	CREATE DATABASE (hay limitaciones de parametros y debe estar conectado a la master database)
Ver la lista de todas las bases de datos	sys.databases (vista)	sys.databases (debe estar conectado a la master database)

Administración a nivel de servidor y la base de datos master

El servidor SQL Server Azure es una abstracción que define una agrupación de bases de datos. Las bases de datos asociadas con el servidor SQL Azure pueden residir en distintos equipos físicos en el centro de datos de Microsoft. Realice la administración a nivel de servidor de todos ellos utilizando una única base de datos con nombre **master**.

La base de datos master hace un seguimiento de los inicios de sesión, y qué inicios de sesión tienen permiso para crear bases de datos o de otros inicios de sesión. Usted debe estar conectado a la base de datos principal cada vez que ejecute CREATE, ALTER o DROP a bases de datos o inicios de sesión. La base de datos master también tiene las vistas sys.sql_logins y sys.databases que puede utilizar para ver los inicios de sesión y bases de datos, respectivamente.

Nota: El comando USE no es compatible con el cambio entre bases de datos. Establecer una conexión directa a la base de datos de destino.

Usted puede manejar la seguridad a nivel de base de datos de los usuarios y los objetos en SQL Azure de la misma manera que una instancia local de SQL Server. Sólo hay diferencias en los parámetros disponibles para los comandos correspondientes.

Administrar inicios de sesión

Administre inicios de sesión con la cuenta principal a nivel de servidor mediante la conexión a la base de datos master. Usted puede usar CREATE, ALTER LOGIN, o DROP LOGIN. El ejemplo siguiente crea un inicio de sesión con nombre login1:

-- primero, conectarse a la db master

CREATE LOGIN login1 WITH password='<ProvidePassword>;

Se recomienda utilizar una contraseña segura cuando se crea un inicio de sesión.

Nuevos inicios de sesión

Debido a que algunas herramientas implementan la secuencia de datos tabular (TDS) de forma diferente, puede que tenga que añadir el nombre del servidor SQL Azure al inicio de sesión en la cadena de conexión utilizando la notación <login> @ <server> . En estos casos, separe el nombre de usuario y el nombre del servidor SQL Azure con el símbolo @. Por ejemplo, si su nombre de usuario fue nombrado login1 y el nombre completo de su servidor SQL Azure es **nombredeservidor.ctp.database.windows.net**, el parámetro de nombre de usuario de la cadena de conexión debe ser: **login1@nombredeservidor**. Esta restricción impone límites sobre el texto que usted puede elegir para el nombre de inicio de sesión.

Con el fin de conectarse a SQL Azure usando los inicios de sesión que ud crea, primero debe conceder a cada inicio de sesión permisos a nivel de base de datos mediante el comando CREATE USER.

Otorgamiento de permisos a inicio de sesión a nivel de servidor

A fin de que los inicios de sesión que no sean la cuenta principal del servidor gestionen la seguridad a nivel de servidor, SQL Azure ofrece dos roles de seguridad: **sds_securityadmin**, para la creación de los inicios de sesión y **sds_dbcreator** para la creación de bases de datos. Estas funciones sólo se pueden asignar a los usuarios en la base de datos master.

Nota: Para crear los inicios de sesión o bases de datos, usted debe estar conectado a la base de datos master.

El Rol **sds_securityadmin**

Al igual que el rol **securityadmin** para una instancia local de SQL Server, el rol **sds_securityadmin** en SQL Azure es necesario para crear inicios de sesión. Sólo el inicio de sesión principal a nivel de servidor (creado por el proceso de aprovisionamiento) u otros inicios de sesión que tienen la función **sds_securityadmin** pueden crear otros inicios de sesión.

El Rol **sds_dbcreator**

El rol **sds_dbcreator** de Azure SQL es similar a la función **dbcreator** para una instancia local de SQL Server. Es necesario para crear bases de datos. Sólo el inicio de sesión principal a nivel de servidor (creado por el proceso de aprovisionamiento) u otros inicios de sesión que tienen la función **sds_dbcreator** pueden crear bases de datos. Una vez que un inicio de sesión tiene el rol **sds_dbcreator**, puede crear una base de datos con el comando CREATE DATABASE, pero esa orden debe ser ejecutada en la base de datos master.

Cómo asignar roles a nivel de servidor

Para crear un usuario que pueda crear bases de datos u otros inicios de sesión, realice los siguientes pasos:

1. Conéctese a la base de datos **master**
2. Cree un inicio de sesión utilizando el comando CREATE LOGIN.
3. Cree un nuevo usuario para ese inicio de sesión en la base de datos **master** usando el comando CREATE USER.
4. Use el procedimiento almacenado **sp_addrolemember** para agregarle la función de **sds_dbcreator**, **sds_securityadmin**, o ambas, al usuario.

El ejemplo de código siguiente se muestra cómo crear un inicio de sesión, login1, y un usuario de base de datos correspondiente llamado login1User que es capaz de crear bases de datos u otros inicios de sesión mientras está conectado a la base de datos master:

- en primer lugar, conectarse a la base de datos master

```
CREATE LOGIN CON login1 password = '<ProvidePassword>';
```

```
Crear desde login1User USUARIO login1;
```

```
Sds_dbcreator EXEC sp_addrolemember ', ' login1User ';
```

```
Sds_securityadmin EXEC sp_addrolemember ',' login1User ';
```

Otorgamiento de permisos a inicio de sesión a nivel de base de datos

Todos los inicios de sesión se deben crear en la base de datos **master**. Después de que un inicio de sesión se ha creado, puede crear una cuenta de usuario en otra base de datos para ese inicio de sesión. SQL Azure soporta las funciones de base de datos de la misma manera que una instancia local de SQL Server lo hace.

Para crear una cuenta de usuario en otra base de datos, suponiendo que no han creado un inicio de sesión o una base de datos, realice los siguientes pasos:

1. Conéctese a la base de datos **master (con un inicio de sesión con la sds_securityadmin y funciones sds_dbcreator)**
2. Cree un nuevo acceso con el comando CREATE LOGIN.
3. Cree una base de datos utilizando el comando CREATE DATABASE.
4. Establezca una conexión a la nueva base de datos (con el inicio de sesión que ha creado la base de datos)
5. Cree un nuevo usuario en la nueva base de datos utilizando el comando CREATE USER.

El ejemplo de código siguiente se muestra cómo crear un inicio de sesión con nombre login1 y una base de datos llamada Database1:

- en primer lugar, conectarse a la base de datos master

```
CREATE LOGIN WITH login1 password = '<ProvidePassword>;
```

```
CREATE DATABASE Database1;
```

El siguiente ejemplo muestra cómo crear un usuario de base de datos llamada login1User en la base de datos Database1 que corresponde al inicio de sesión login1:

-- Establezca una nueva conexión a la base de datos database1

```
CREATE USER login1User FROM LOGIN login1;
```

Visualizar inicios de sesión y base de datos

Para ver inicios de sesión y base de datos en su servidor SQL Azure, use las vistas sys.sql_logins y sys.databases disponibles en la base de datos master. El siguiente ejemplo muestra como desplegar una lista de todos los inicios de sesión y bases de datos

-- Primero, conectarse a la base de datos master

```
SELECT * FROM sys.sql_logins;
```

```
SELECT * FROM sys.databases;
```

Conectarse a Azure SQL utilizando ADO.NET

El uso de ADO.NET para conectarse a una base de datos en Microsoft SQL Azure es muy similar a conectarse a una instancia de SQL Server en sus instalaciones. A continuación se muestra un ejemplo.

- 1) Cree una solución en Visual Studio del tipo Windows Form
- 2) Agregue a la form un DataGridView
- 3) Copie el siguiente código en el code behind de la form:

```
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

Public Class Form1

    'Ingrese la informacion requerida
    Private ReadOnly userName As String = "xxxxxxxx"
    Private ReadOnly password As String = "xxxxxxx."
    Private ReadOnly dataSource As String = "tcp:xxxxxx.ctp.database.windows.net"
    Private ReadOnly BaseDeDatosDeEjemplo As String = "BaseDeDatosDeEjemplo"

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

        Try
            ' Crear la conexion a la base Master - necesaria para hacer consultas DDL
            Dim connString1Builder As New SqlConnectionStringBuilder()
            With connString1Builder
                .DataSource = dataSource
                .InitialCatalog = "master"
                .Encrypt = True
                .TrustServerCertificate = True
                .UserID = userName
                .Password = password
            End With

            ' Crear la conexion a la base de ejemplo
            Dim connString2Builder As New SqlConnectionStringBuilder()
            With connString2Builder
                .DataSource = dataSource
                .InitialCatalog = BaseDeDatosDeEjemplo
                .Encrypt = True
                .TrustServerCertificate = True
                .UserID = userName
                .Password = password
            End With

            ' Conectarse a la base master y crear la base de ejemplo
            Using conn As New SqlConnection(connString1Builder.ToString())
                Using command As SqlCommand = conn.CreateCommand()
                    conn.Open()
                    command.CommandText = String.Format("SELECT * FROM sys.databases WHERE name = '{0}'",
BaseDeDatosDeEjemplo)
                    'Si la base no existe ... crearla
                    If command.ExecuteScalar() Is Nothing Then
                        command.CommandText = String.Format("CREATE DATABASE {0}", BaseDeDatosDeEjemplo)
                        command.ExecuteNonQuery()
                    End If
                    conn.Close()
                End Using
            End Using

            ' Conectarse a la base de ejemplo y realizar operaciones sobre la misma
            Using conn As New SqlConnection(connString2Builder.ToString())
                Using command As SqlCommand = conn.CreateCommand()
                    conn.Open()
                    command.CommandText = "SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='Cliente'"
                    'Si la tabla no existe ... crearla
                    If command.ExecuteScalar() Is Nothing Then
```

```

        command.CommandText = GenerarStringParaCrearTablaClientes()
        command.ExecuteNonQuery()
    End If
    conn.Close()

    'Insertar registro de ejemplo
    'command.CommandText = "INSERT INTO [dbo].[Cliente..."
    'Dim rowsInsertadas As Integer = command.ExecuteNonQuery()

    command.CommandText = "SELECT * FROM Cliente"
    Dim adapter As SqlDataAdapter = New SqlDataAdapter(command)
    Dim clientes As DataSet = New DataSet
    adapter.Fill(clientes, "Clientes")
    DataGridView1.DataSource = clientes.Tables(0)

    End Using
End Using

Catch ex As Exception

End Try
End Sub

''' <summary>
''' Metodo que crea la tabla cliente
''' </summary>
''' <returns></returns>
''' <remarks></remarks>
Private Function GenerarStringParaCrearTablaClientes() As String
    Dim tabla As New StringBuilder
    tabla.AppendLine("CREATE TABLE Cliente (")
    tabla.AppendLine("[CLI_Id] [uniqueidentifier] NOT NULL PRIMARY KEY,")
    tabla.AppendLine("[CLI_Nombre] [nvarchar](255) NOT NULL,")
    tabla.AppendLine("[CLI_RazonSocial] [nvarchar](255) NULL,")
    tabla.AppendLine("[CLI_RazonSocialFiscal] [nvarchar](255) NULL,")
    tabla.AppendLine("[CLI_Direccion] [nvarchar](255) NOT NULL,")
    tabla.AppendLine("[CLI_Localidad] [nvarchar](100) NULL,")
    tabla.AppendLine("[CLI_RUC] [nvarchar](255) NULL,")
    tabla.AppendLine("[CLI_Documento] [nvarchar](100) NULL,")
    tabla.AppendLine("[CLI_Telefono] [nvarchar](255) NOT NULL,")
    tabla.AppendLine("[CLI_Email] [nvarchar](100) NULL,")
    tabla.AppendLine("[CLI_CodigoPostal] [nvarchar](100) NULL,")
    tabla.AppendLine("[CLI_Fax] [nvarchar](100) NULL,")
    tabla.AppendLine("[CLI_Contacto] [nvarchar](255) NULL,")
    tabla.AppendLine("[CLI_Observaciones] [nvarchar](255) NULL,")
    tabla.AppendLine("[CLI_Pais] [bigint] NOT NULL,")
    tabla.AppendLine("[CLI_Departamento] [bigint] NOT NULL)")
    Return tabla.ToString()
End Function
End Class

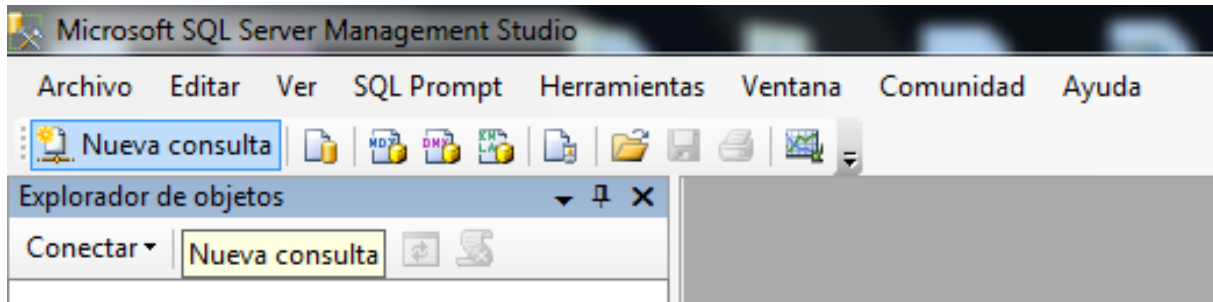
```

- 4) Reemplace usuario, password y servidor (marcados en el código con xxxxxx)
- 5) Ejecute la solución y si todo sale bien deberá ver una pantalla similar a ésta:

CLI_Id	CLI_Nombre	CLI_RazonSocial	CLI_RazonSocialFi	CLI_Direccion	CLI_Localidad	CLI_RUC	CLI_Documento	CLI
87778770-0095...	L2UZGE1T42JL...	U446FQINEOF1...	Z5T6N2MTNBB...	J5HBJO9NJOH...	29022	HAOBRK39N42...	NRXRXN13CW...	002-3
6b635de2-9f63-9...	Q1W3JDC8MU7...	N1MU2HG2D8E...	J7	PBK6S8WY90YI...	63737	6PXBFDI2LVKO...	785DQISREJON...	4898
19d865a4-1c0d-...	CG0UEUVIBPMJ...	EX3VPT6DKTFU...	STTP89GJ2D01...	HEE7XOYQU2XI...	20592	MSFBV8C83F3P...	MXBPT4W2KTG...	4074
a6160a25-389f-f...	S617HK6793LJY...	W7GFA3E0ICNK...	7CX249HRT3LF...	MHYVDFZD7VH...	74532	JN7RKM54F258...	139DW37ET9KX...	028-9
40faa097-3a8c-e...	OCBRRE8UYTB...	P380YS7534W3...	8CD2C2FRFHOM...	KNGVLLKB4QE...	32929	6F1YTH47A2ZX...	E745ZCGLB8SE...	5127
b852f4eb-0a26-5...	SBDGTMP8E79...	Y2UQFEU9UJ2...	E4AHCVN1SMA...	FVMWVKDB8Y...	72207	R5X216RRR805D...	PSESEVT4NU...	547-3
e01dd7e2-3434-...	X509TUOAW4C...	EASAM	S0VJ0RY8BEML...	4T6ECC2W0T70...	84394	DLZ9KLS0V1059...	60SKILH02ZQM...	311-8
af1f6667-9afc-6c...	YORZZAKOC5H...	V3LG13CDNLM...	MF9JPJTC1GRD...	V74RV169WHA...	77654	3YK8695XVWG3...	0CSLMN1R7GFJ...	237-3
b20eaed0-9b62-...	4IBLW43Y69QW...	HH0914I7SKHT...	HP83XBT70HON...	QA7CYEP6I5C9...	30402	CBR96572JKW8...	DG4RRD78EYJ...	098-5
be4dfe18-2c08-f...	L34KSJNX2RC3...	EK5YWS9YQAQYK...	WIU3YBYH82IP...	989CZ3WBELJA...	91828	LD8HRCR4GZLC...	P4RVT2PNEX6A...	815-C
0e0f1889-bc2-8...	4TXLM6PL5S94...	9NMHAF2HRXB...	KX1BDRGF35C8...	478CODQL9IQE...	95447	C250U67F80GD...	C3I	0502
151a0626-1d8b-f...	XA57PC8A0X1A...	1UEDUWA93BU3...	QA10A30Y6BFN...	ZPEVR1L67M3I...	97610	F4YQZBH8PAQ...	6PFKX723N7BD...	663-8
afe99c8f-790f-c1...	1CIGJESQ7MH2...	89FZU8HKCWR...	JDJ5PXG8GM79...	655SDLEUHUT...	81424	FFVXV8ZP4CQ2...	FSTHY7R89MS3...	126-3
30335da4-be50-...	8SF70UT3W6Q...	6RAZ8XQCQS...	VXGNASLSYOR...	1UE69ASRC29G...	65294	MLA9L4LSQXLA...	R4VTSI2MKFW...	393-3
32502023-4b75-...	HBW8J25LBZKK...	RA78QVY6YN3...	IYTDV65RM95Y...	6D5PY7T2BDPS...	72791	YML1BPVCHAO...	8D2	864-1
2e017358-1fe7-3...	F3BX3RPOXP8N...	90B8DGT0VKR...	D6DCX55100Q...	FZDL93RI7J8W...	24447	UXLQ9U21GHR1...	668X1F817H8BL...	825-C
1e308786-0581-f...	8170AO5AHKCV...	DYWZLW9RDS...	3SKVTGTP4X2...	BUR8KDJWZEC...	27755	LXZ8MJL8ILRB...	VACXM0YUFUS...	0696
6bb5fa3e-ac63-4...	HWJMN01Z6CN...	DDEV3KZXHFUI...	ZNYK45MCFYI...	35V5YP97EAILX...	24630	GB55XFEO5JL8...	SIDAAXWXUTH...	1512
fe2069ea-e9b4-2...	IK4Z2J2BYEX7N...	T8AF61TGL68S...	NY602675N6LK...	QKHS718M1CU...	21696	8FEDVFMWYJ9...	BIUFGRRRMO...	947-6
fb6eff33-c231-b0...	GG16Y1Z65Q3Q...	6D0GX77Y9TW...	9BWSFDL0IQQP...	E1945V5U0IYG...	90048	JMQMRNWE61...	ZC1GHB0UJL3Y...	353-5
e9898fe9-8582-c...	H2C866TT1LJ7V...	UJDXDRTHVPM...	YBW951WTTN...	GJY9XA6W4K7A...	64312	ONJGENP73BF...	WHDQ7UTSVX...	705-3

Conectarse a Azure SQL utilizando Microsoft SQL Management Studio

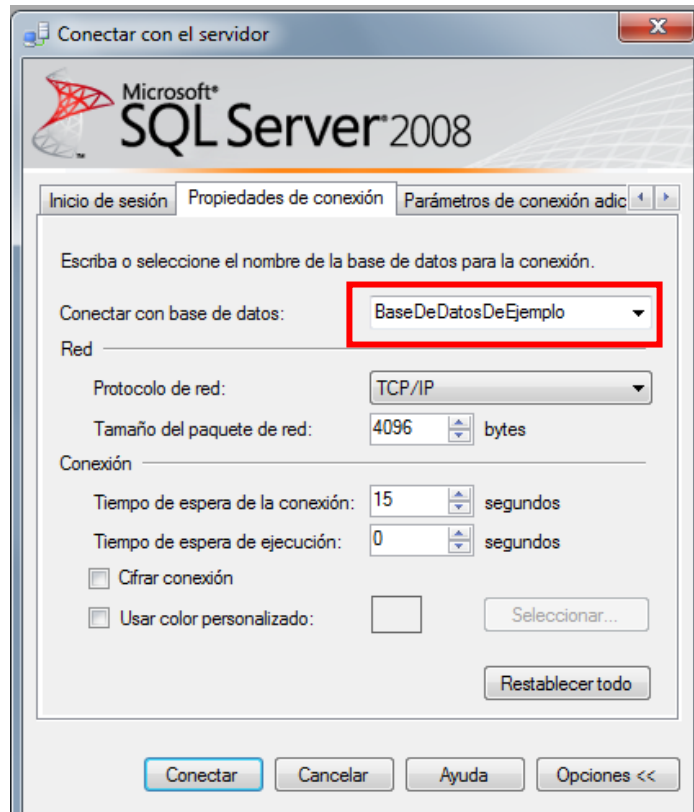
Sin estar conectado a ninguna base presione nueva consulta



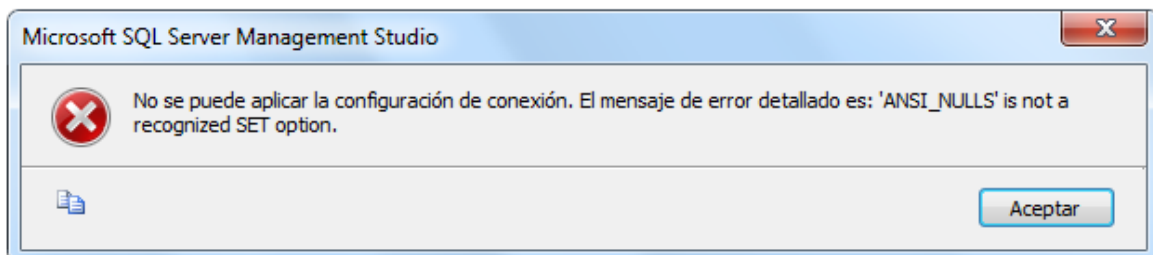
Ingrese la información en el cuadro de dialogo (Servidor, Usuario y Password) y a continuación presione el botón opciones (abajo a la derecha del botón ayuda)



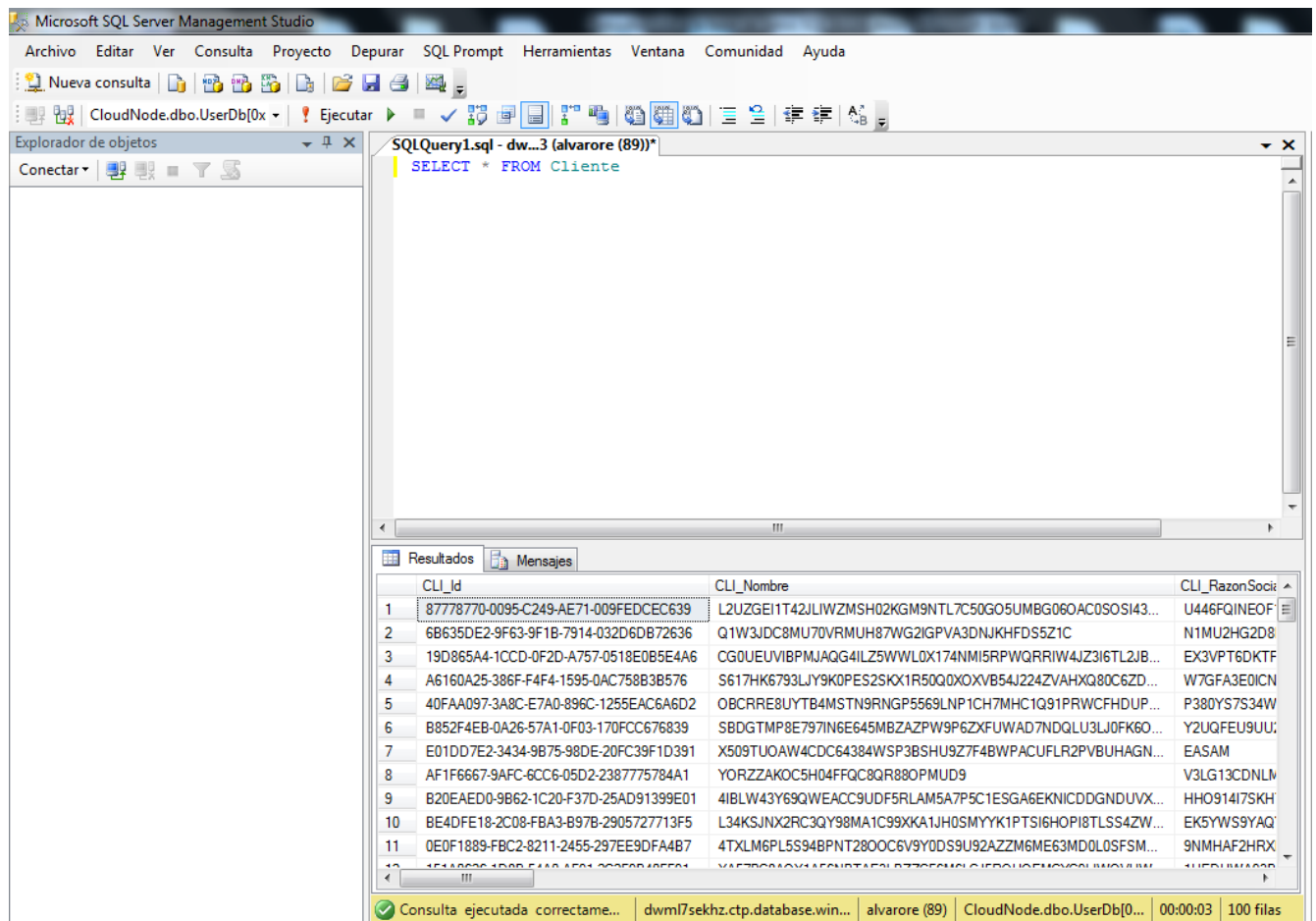
Ingrese el nombre de base de datos al que desea conectarse y presione conectar



Aparecerá un error que puede ignorar presionando el botón aceptar



Finalmente, escriba la consulta y presione Ejecutar



Nota: Como se aprecia en la imagen el explorador de objetos no está disponible

Referencias

<http://msdn.microsoft.com/en-us/library/ee336279.aspx>

<http://english.zachskylesowens.net/2009/08/18/connecting-to-sql-azure/>



Comunidad de Usuarios Microsoft Uruguay



Agosto 2009

L&A SISTEMAS

www.lyasistemas.com

513-76-13