Design your first Azure SQL database

Azure SQL Database is a relational database-as-a service (DBaaS) in the Microsoft Cloud (Azure). In this tutorial, you learn how to use the Azure portal and [SQL Server Management Studio](https://msdn.microsoft.com/library/ms174173.aspx) (SSMS) to:

* Create a database in the Azure portal
* Set up a server-level firewall rule in the Azure portal
* Connect to the database with SSMS
* Create tables with SSMS
* Bulk load data with BCP
* Query that data with SSMS
* Restore the database to a previous [point in time restore](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-recovery-using-backups#point-in-time-restore) in the Azure portal

If you don't have an Azure subscription, [create a free account](https://azure.microsoft.com/free/) before you begin.

Prerequisites

To complete this tutorial, make sure you have installed:

* The newest version of [SQL Server Management Studio](https://msdn.microsoft.com/library/ms174173.aspx) (SSMS).
* The newest version of [BCP and SQLCMD](https://www.microsoft.com/download/details.aspx?id=36433).

Log in to the Azure portal

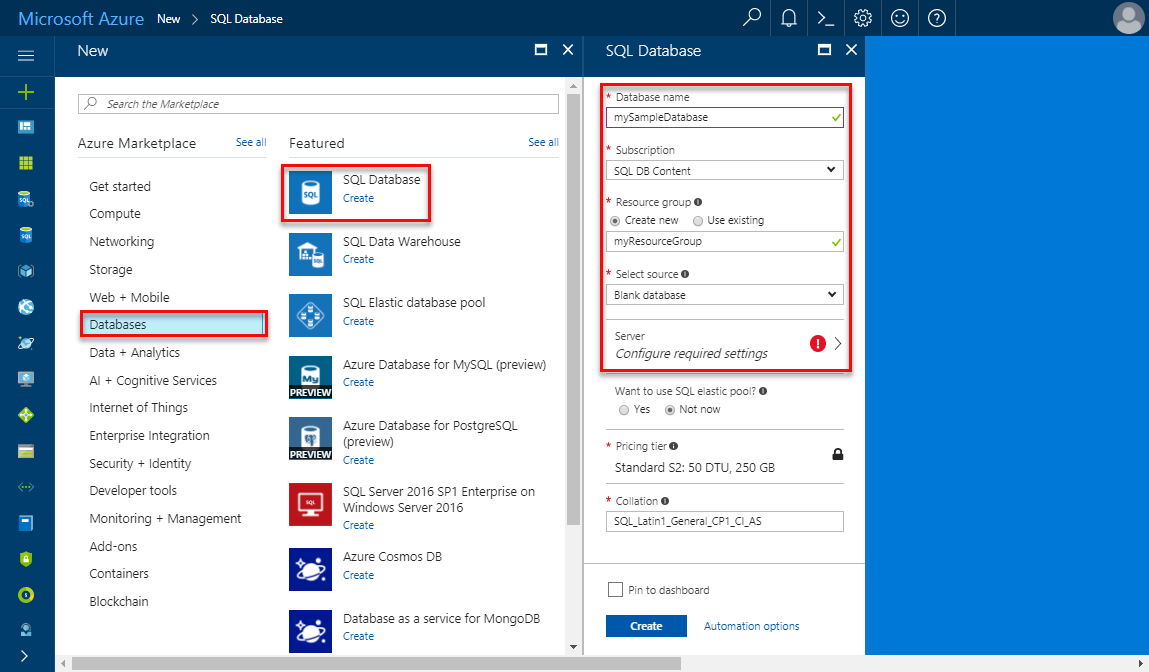
Log in to the [Azure portal](https://portal.azure.com/).

Create a blank SQL database

An Azure SQL database is created with a defined set of [compute and storage resources](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-service-tiers). The database is created within an [Azure resource group](https://docs.microsoft.com/en-in/azure/azure-resource-manager/resource-group-overview) and in an [Azure SQL Database logical server](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-features).

Follow these steps to create a blank SQL database.

1. Click the **New** button found on the upper left-hand corner of the Azure portal.
2. Select **Databases** from the **New** page, and select **Create** under **SQL Database** on the **New** page.

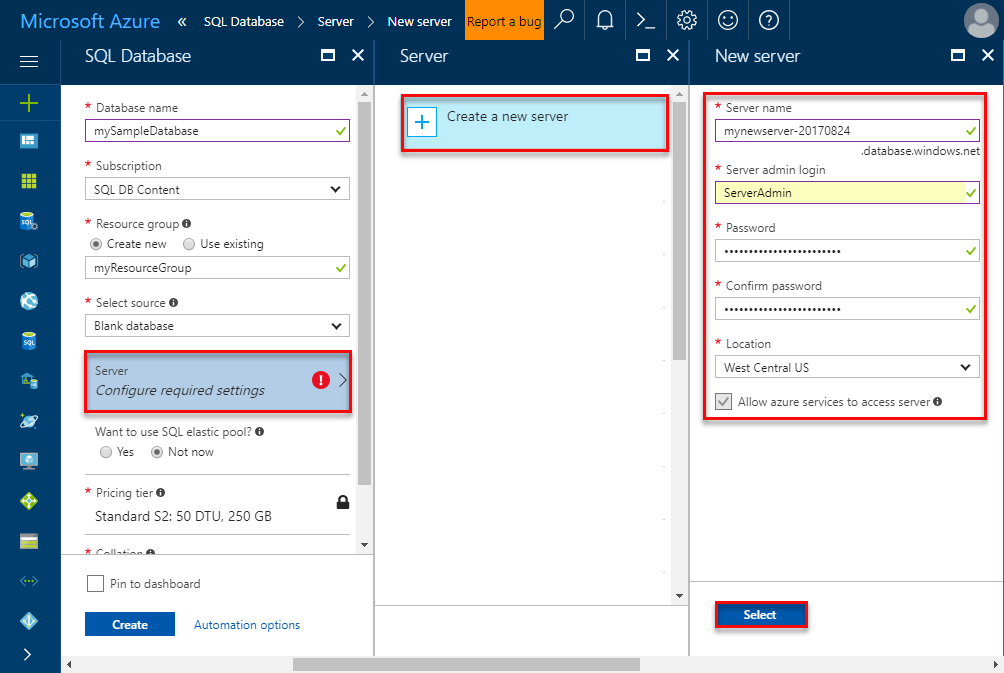


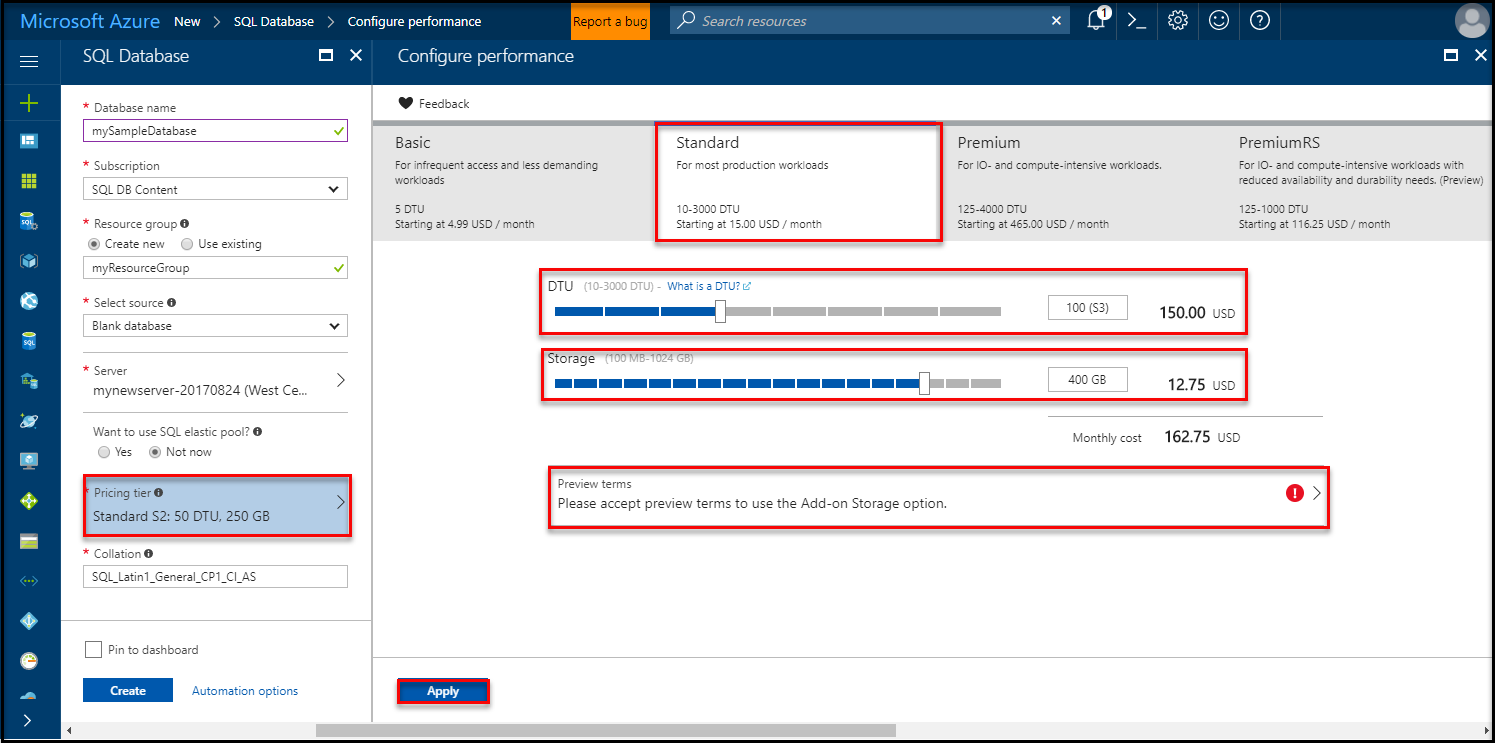
1. Fill out the SQL Database form with the following information, as shown on the preceding image:

| Setting | Suggested value | Description |
| --- | --- | --- |
| **Database name** | mySampleDatabase | For valid database names, see [Database Identifiers](https://docs.microsoft.com/sql/relational-databases/databases/database-identifiers). |
| **Subscription** | Your subscription | For details about your subscriptions, see [Subscriptions](https://account.windowsazure.com/Subscriptions). |
| **Resource group** | myResourceGroup | For valid resource group names, see [Naming rules and restrictions](https://docs.microsoft.com/azure/architecture/best-practices/naming-conventions). |
| **Select source** | Blank database | Specifies that a blank database should be created. |

1. Click **Server** to create and configure a new server for your new database. Fill out the **New server form** with the following information:

| Setting | Suggested value | Description |
| --- | --- | --- |
| **Server name** | Any globally unique name | For valid server names, see [Naming rules and restrictions](https://docs.microsoft.com/azure/architecture/best-practices/naming-conventions). |
| **Server admin login** | Any valid name | For valid login names, see [Database Identifiers](https://docs.microsoft.com/sql/relational-databases/databases/database-identifiers). |
| **Password** | Any valid password | Your password must have at least eight characters and must contain characters from three of the following categories: upper case characters, lower case characters, numbers, and non-alphanumeric characters. |
| **Location** | Any valid location | For information about regions, see [Azure Regions](https://azure.microsoft.com/regions/). |

1. 
2. Click **Select**.
3. Click **Pricing tier** to specify the service tier, the number of DTUs, and the amount of storage. Explore the options for the number of DTUs and storage that is available to you for each service tier.
4. For this tutorial, select the **Standard** service tier and then use the slider to select **100 DTUs (S3)** and **400** GB of storage.



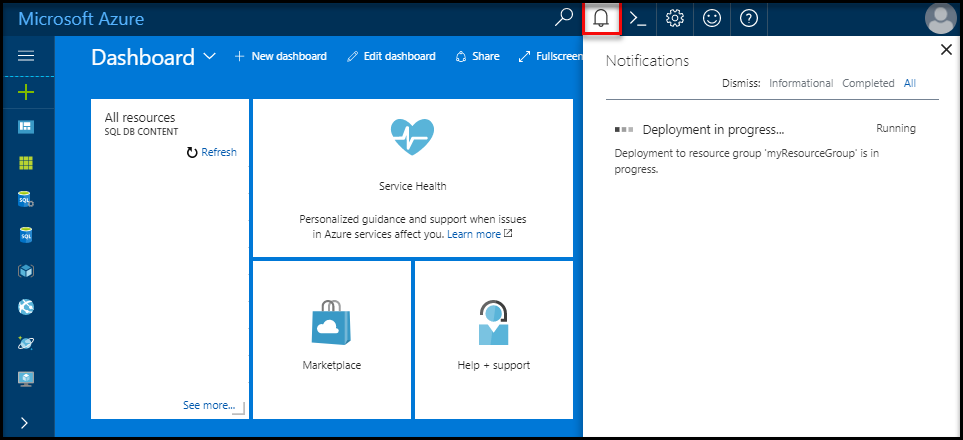
1. Accept the preview terms to use the **Add-on Storage** option.

Important

\* Storage sizes greater than the amount of included storage are in preview and extra costs apply. For details, see [SQL Database pricing](https://azure.microsoft.com/pricing/details/sql-database/).

\* In the Premium tier, more than 1 TB of storage is currently available in the following regions: US East2, West US, US Gov Virginia, West Europe, Germany Central, South East Asia, Japan East, Australia East, Canada Central, and Canada East. See [P11-P15 Current Limitations](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-resource-limits#single-database-limitations-of-p11-and-p15-when-the-maximum-size-greater-than-1-tb).

1. After selecting the server tier, the number of DTUs, and the amount of storage, click **Apply**.
2. Select a **collation** for the blank database (for this tutorial, use the default value). For more information about collations, see [Collations](https://docs.microsoft.com/sql/t-sql/statements/collations)
3. Now that you have completed the SQL Database form, click **Create** to provision the database. Provisioning takes a few minutes.
4. On the toolbar, click **Notifications** to monitor the deployment process.



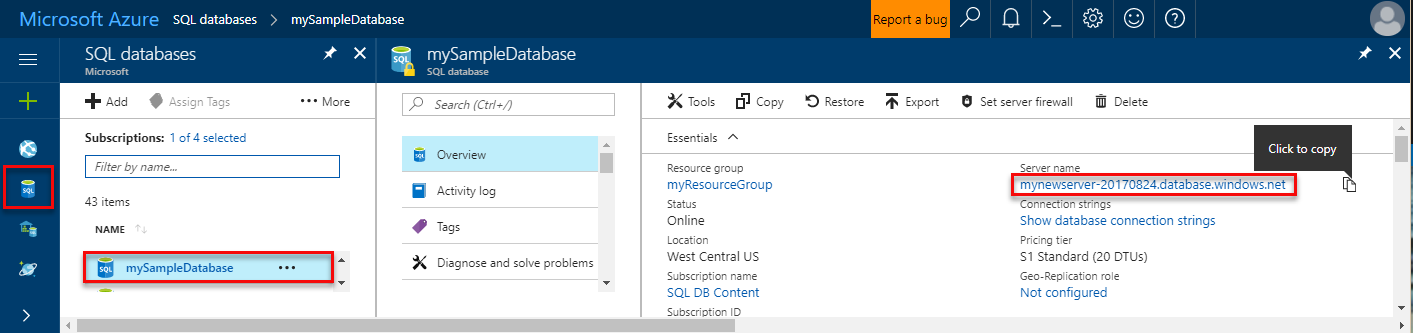
Create a server-level firewall rule

The SQL Database service creates a firewall at the server-level that prevents external applications and tools from connecting to the server or any databases on the server unless a firewall rule is created to open the firewall for specific IP addresses. Follow these steps to create a [SQL Database server-level firewall rule](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-firewall-configure) for your client's IP address and enable external connectivity through the SQL Database firewall for your IP address only.

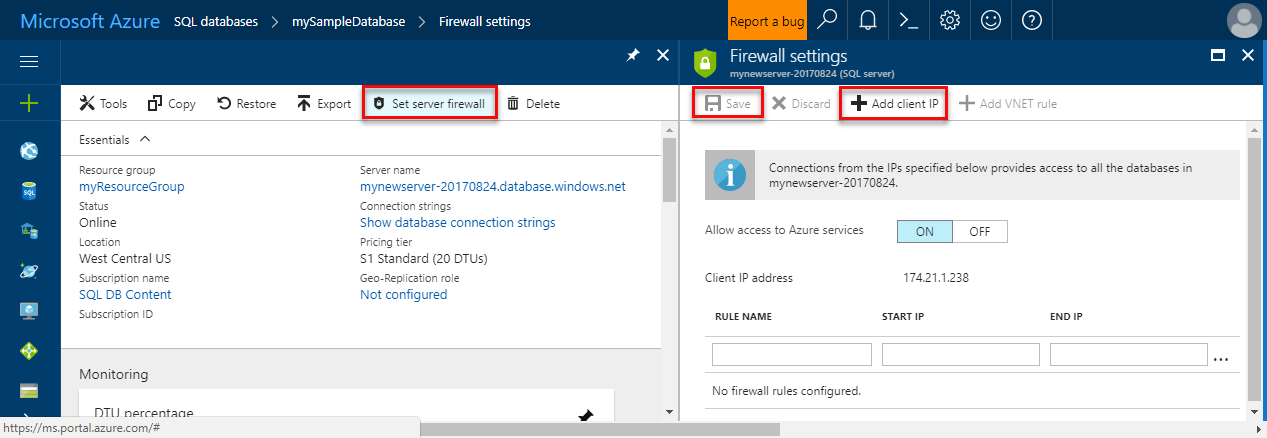
Note

SQL Database communicates over port 1433. If you are trying to connect from within a corporate network, outbound traffic over port 1433 may not be allowed by your network's firewall. If so, you cannot connect to your Azure SQL Database server unless your IT department opens port 1433.

1. After the deployment completes, click **SQL databases** from the left-hand menu and then click **mySampleDatabase** on the **SQL databases** page. The overview page for your database opens, showing you the fully qualified server name (such as **mynewserver-20170824.database.windows.net**) and provides options for further configuration.
2. Copy this fully qualified server name for use to connect to your server and its databases in subsequent quick starts.



1. Click **Set server firewall** on the toolbar. The **Firewall settings** page for the SQL Database server opens.



1. Click **Add client IP** on the toolbar to add your current IP address to a new firewall rule. A firewall rule can open port 1433 for a single IP address or a range of IP addresses.
2. Click **Save**. A server-level firewall rule is created for your current IP address opening port 1433 on the logical server.
3. Click **OK** and then close the **Firewall settings** page.

You can now connect to the SQL Database server and its databases using SQL Server Management Studio or another tool of your choice from this IP address using the server admin account created previously.

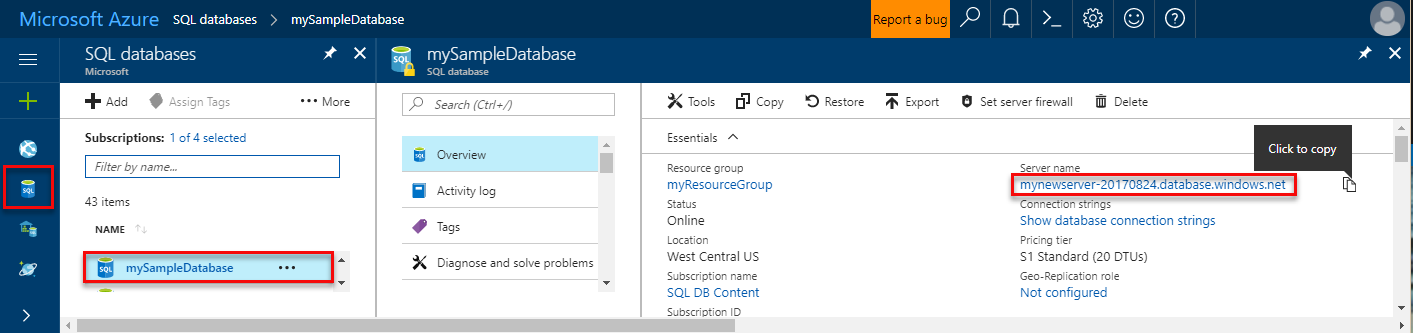
Important

By default, access through the SQL Database firewall is enabled for all Azure services. Click **OFF** on this page to disable for all Azure services.

SQL server connection information

Get the fully qualified server name for your Azure SQL Database server in the Azure portal. You use the fully qualified server name to connect to your server using SQL Server Management Studio.

1. Log in to the [Azure portal](https://portal.azure.com/).
2. Select **SQL Databases** from the left-hand menu, and click your database on the **SQL databases** page.
3. In the **Essentials** pane in the Azure portal page for your database, locate and then copy the **Server name**.

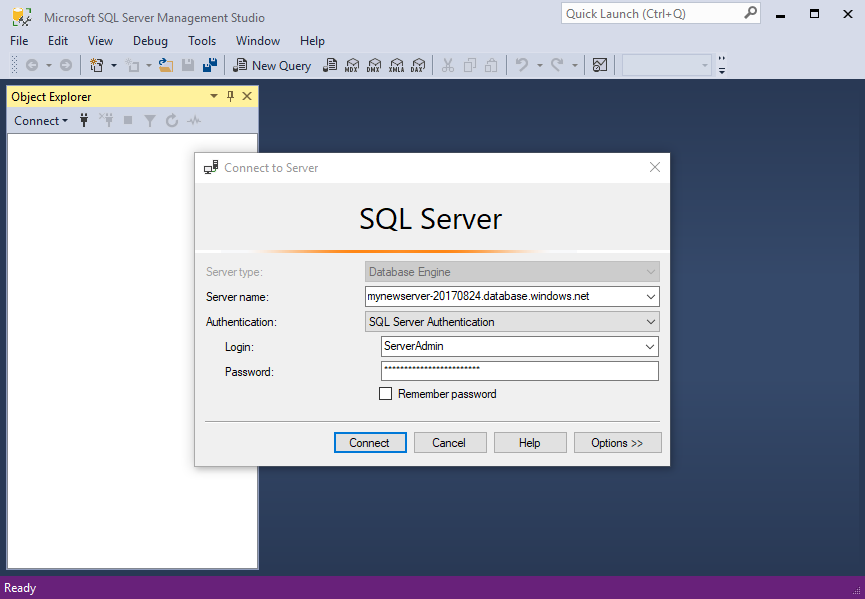


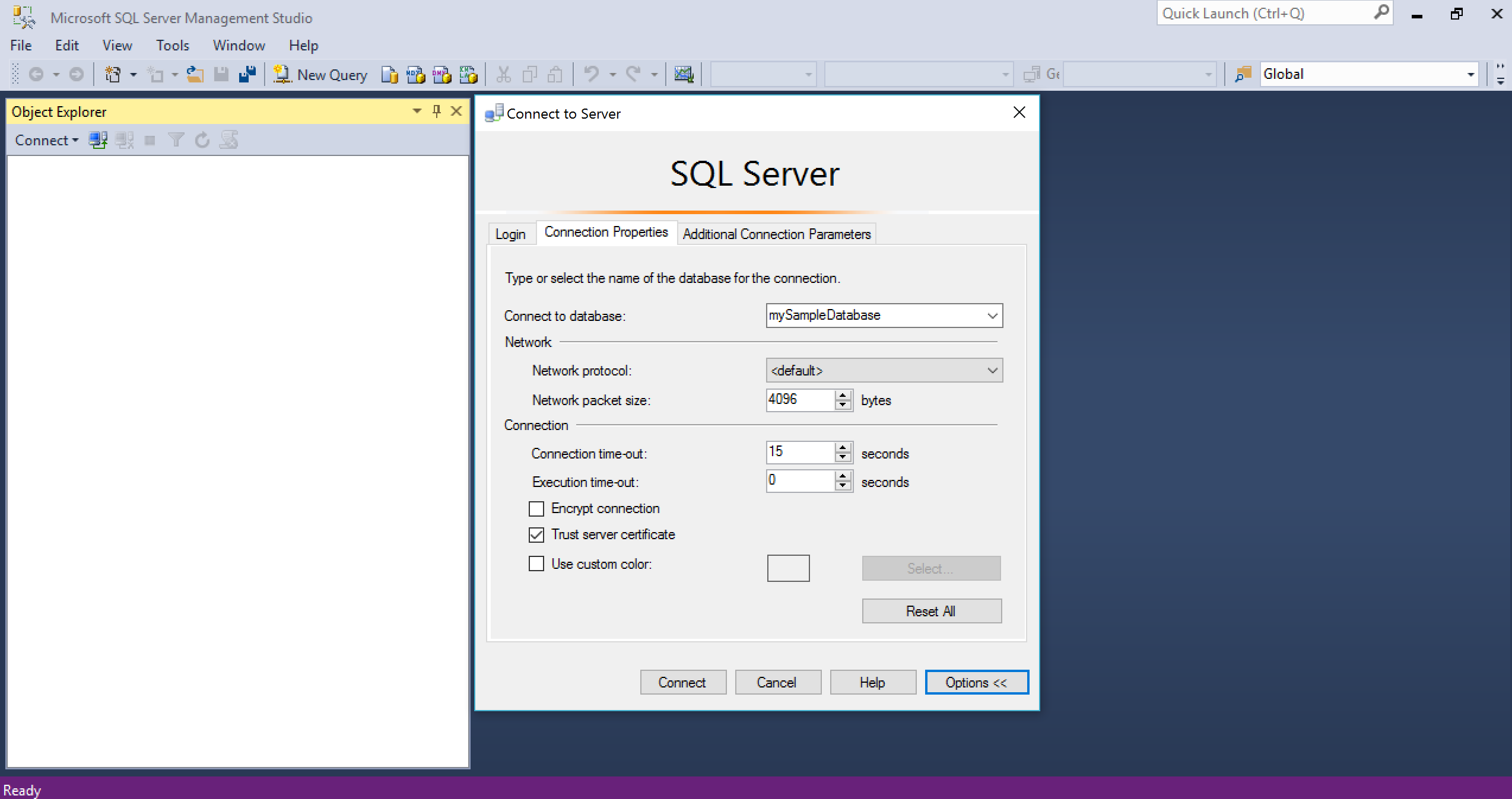
Connect to the database with SSMS

Use [SQL Server Management Studio](https://docs.microsoft.com/sql/ssms/sql-server-management-studio-ssms) to establish a connection to your Azure SQL Database server.

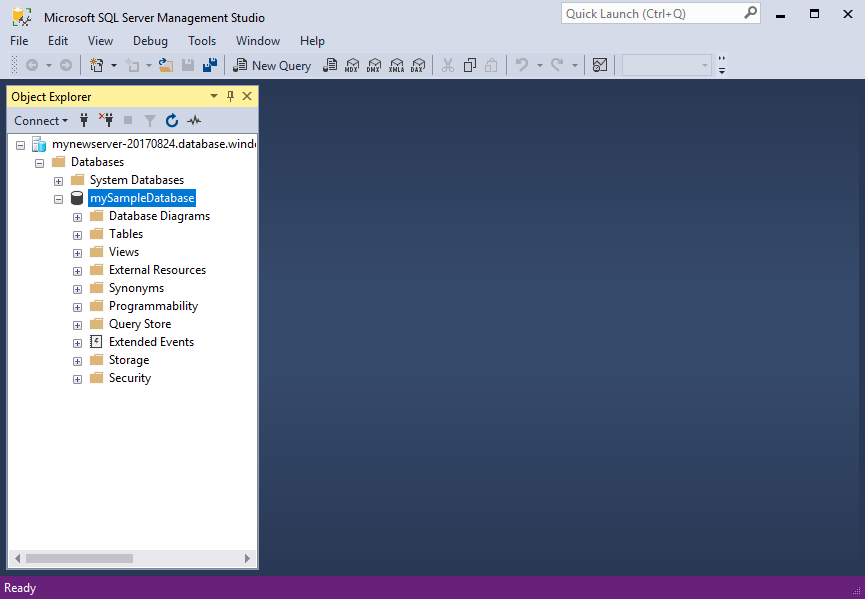
1. Open SQL Server Management Studio.
2. In the **Connect to Server** dialog box, enter the following information:

| Setting | Suggested value | Description |
| --- | --- | --- |
| Server type | Database engine | This value is required |
| Server name | The fully qualified server name | The name should be something like this: **mynewserver20170824.database.windows.net**. |
| Authentication | SQL Server Authentication | SQL Authentication is the only authentication type that we have configured in this tutorial. |
| Login | The server admin account | This is the account that you specified when you created the server. |
| Password | The password for your server admin account | This is the password that you specified when you created the server. |

1. 
2. Click **Options** in the **Connect to server** dialog box. In the **Connect to database** section, enter **mySampleDatabase** to connect to this database.



1. Click **Connect**. The Object Explorer window opens in SSMS.
2. In Object Explorer, expand **Databases** and then expand **mySampleDatabase** to view the objects in the sample database.



Create tables in the database

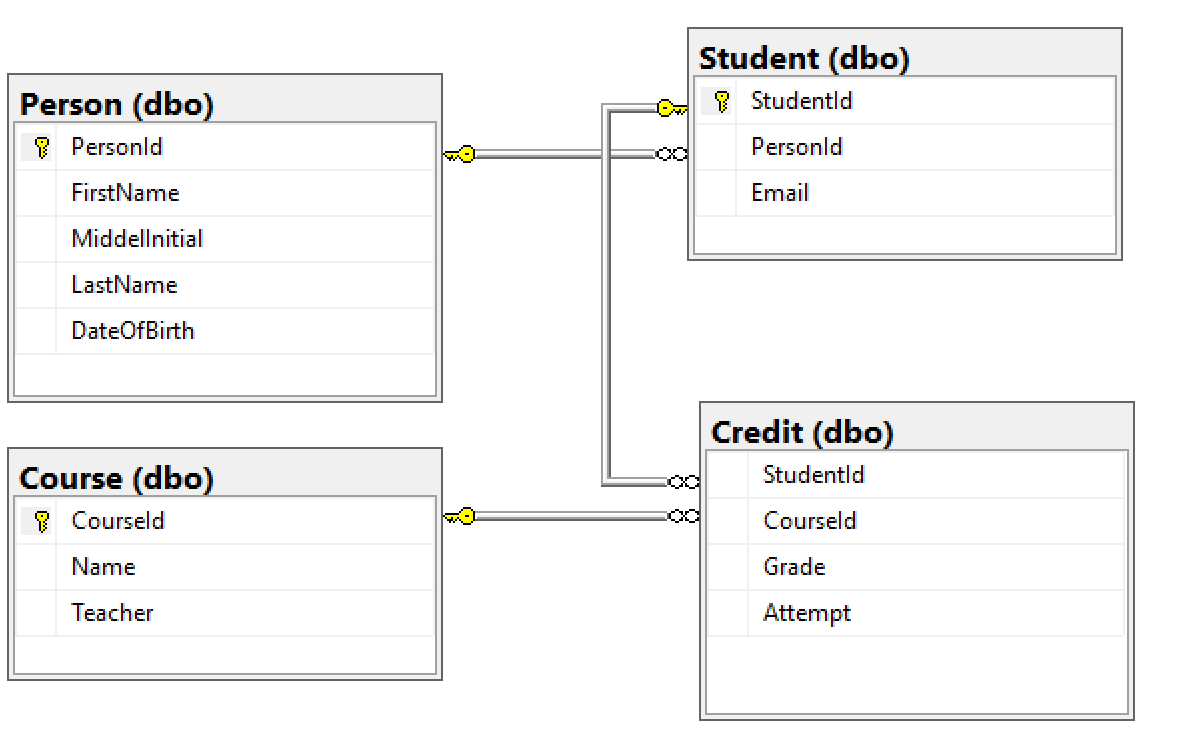
Create a database schema with four tables that model a student management system for universities using [Transact-SQL](https://docs.microsoft.com/sql/t-sql/language-reference):

* Person
* Course
* Student
* Credit that model a student management system for universities

The following diagram shows how these tables are related to each other. Some of these tables reference columns in other tables. For example, the Student table references the **PersonId**column of the **Person** table. Study the diagram to understand how the tables in this tutorial are related to one another. For an in-depth look at how to create effective database tables, see [Create effective database tables](https://msdn.microsoft.com/library/cc505842.aspx). For information about choosing data types, see [Data types](https://docs.microsoft.com/sql/t-sql/data-types/data-types-transact-sql).

Note

You can also use the [table designer in SQL Server Management Studio](https://msdn.microsoft.com/library/hh272695.aspx) to create and design your tables.



1. In Object Explorer, right-click **mySampleDatabase** and click **New Query**. A blank query window opens that is connected to your database.
2. In the query window, execute the following query to create four tables in your database:

SQLCopy

-- Create Person table

CREATE TABLE Person

(

PersonId INT IDENTITY PRIMARY KEY,

FirstName NVARCHAR(128) NOT NULL,

MiddelInitial NVARCHAR(10),

LastName NVARCHAR(128) NOT NULL,

DateOfBirth DATE NOT NULL

)

-- Create Student table

CREATE TABLE Student

(

StudentId INT IDENTITY PRIMARY KEY,

PersonId INT REFERENCES Person (PersonId),

Email NVARCHAR(256)

)

-- Create Course table

CREATE TABLE Course

(

CourseId INT IDENTITY PRIMARY KEY,

Name NVARCHAR(50) NOT NULL,

Teacher NVARCHAR(256) NOT NULL

)

-- Create Credit table

CREATE TABLE Credit

(

StudentId INT REFERENCES Student (StudentId),

CourseId INT REFERENCES Course (CourseId),

Grade DECIMAL(5,2) CHECK (Grade <= 100.00),

Attempt TINYINT,

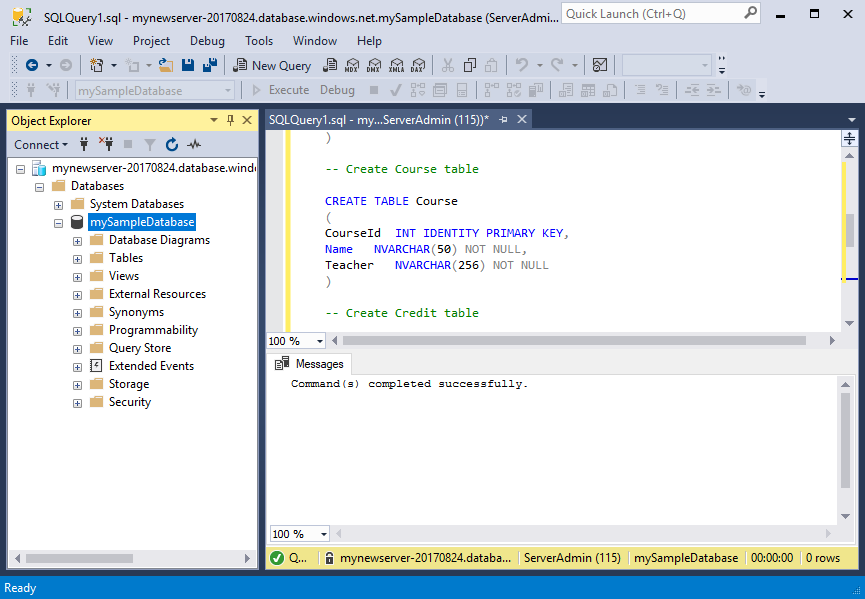
CONSTRAINT [UQ\_studentgrades] UNIQUE CLUSTERED

(

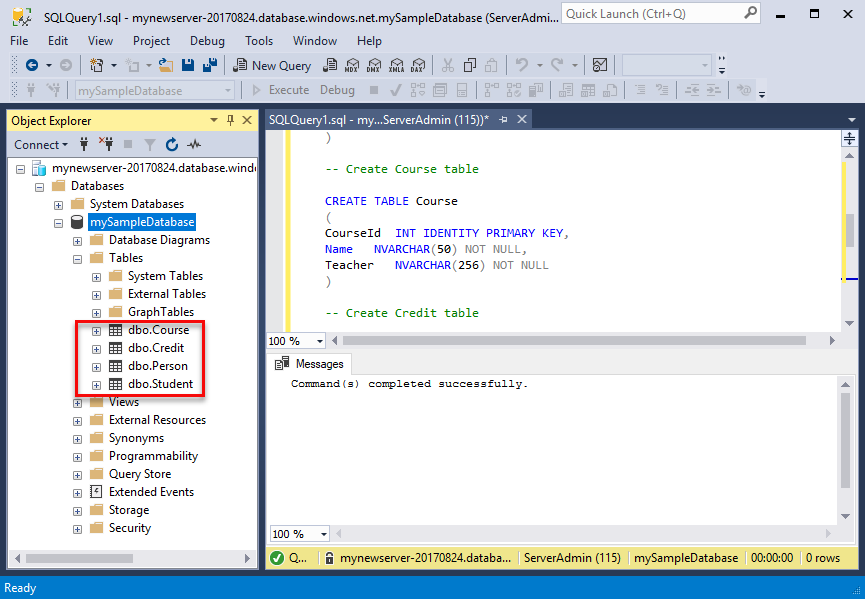
StudentId, CourseId, Grade, Attempt

)

)



1. Expand the 'tables' node in the SQL Server Management Studio Object explorer to see the tables you created.



Load data into the tables

1. Create a folder called **SampleTableData** in your Downloads folder to store sample data for your database.
2. Right-click the following links and save them into the **SampleTableData** folder.
   * [SampleCourseData](https://sqldbtutorial.blob.core.windows.net/tutorials/SampleCourseData)
   * [SamplePersonData](https://sqldbtutorial.blob.core.windows.net/tutorials/SamplePersonData)
   * [SampleStudentData](https://sqldbtutorial.blob.core.windows.net/tutorials/SampleStudentData)
   * [SampleCreditData](https://sqldbtutorial.blob.core.windows.net/tutorials/SampleCreditData)
3. Open a command prompt window and navigate to the SampleTableData folder.
4. Execute the following commands to insert sample data into the tables replacing the values for **ServerName**, **DatabaseName**, **UserName**, and **Password** with the values for your environment.

bcpCopy

bcp Course in SampleCourseData -S <ServerName>.database.windows.net -d <DatabaseName> -U <Username> -P <password> -q -c -t ","

bcp Person in SamplePersonData -S <ServerName>.database.windows.net -d <DatabaseName> -U <Username> -P <password> -q -c -t ","

bcp Student in SampleStudentData -S <ServerName>.database.windows.net -d <DatabaseName> -U <Username> -P <password> -q -c -t ","

bcp Credit in SampleCreditData -S <ServerName>.database.windows.net -d <DatabaseName> -U <Username> -P <password> -q -c -t ","

You have now loaded sample data into the tables you created earlier.

Query data

Execute the following queries to retrieve information from the database tables. See [Writing SQL Queries](https://technet.microsoft.com/library/bb264565.aspx) to learn more about writing SQL queries. The first query joins all four tables to find all the students taught by 'Dominick Pope' who have a grade higher than 75% in his class. The second query joins all four tables and finds all courses in which 'Noe Coleman' has ever enrolled.

1. In a SQL Server Management Studio query window, execute the following query:

SQLCopy

-- Find the students taught by Dominick Pope who have a grade higher than 75%

SELECT person.FirstName,

person.LastName,

course.Name,

credit.Grade

FROM Person AS person

INNER JOIN Student AS student ON person.PersonId = student.PersonId

INNER JOIN Credit AS credit ON student.StudentId = credit.StudentId

INNER JOIN Course AS course ON credit.CourseId = course.courseId

WHERE course.Teacher = 'Dominick Pope'

AND Grade > 75

1. In a SQL Server Management Studio query window, execute following query:

SQLCopy

-- Find all the courses in which Noe Coleman has ever enrolled

SELECT course.Name,

course.Teacher,

credit.Grade

FROM Course AS course

INNER JOIN Credit AS credit ON credit.CourseId = course.CourseId

INNER JOIN Student AS student ON student.StudentId = credit.StudentId

INNER JOIN Person AS person ON person.PersonId = student.PersonId

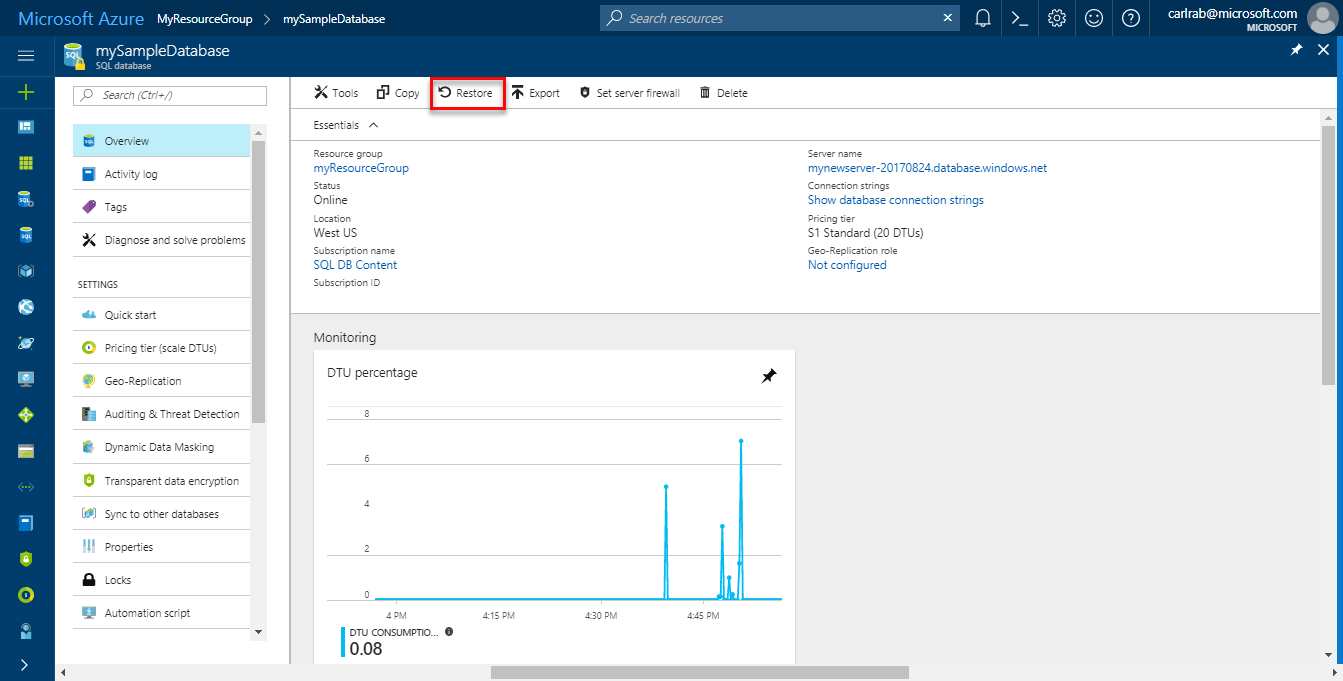
WHERE person.FirstName = 'Noe'

AND person.LastName = 'Coleman'

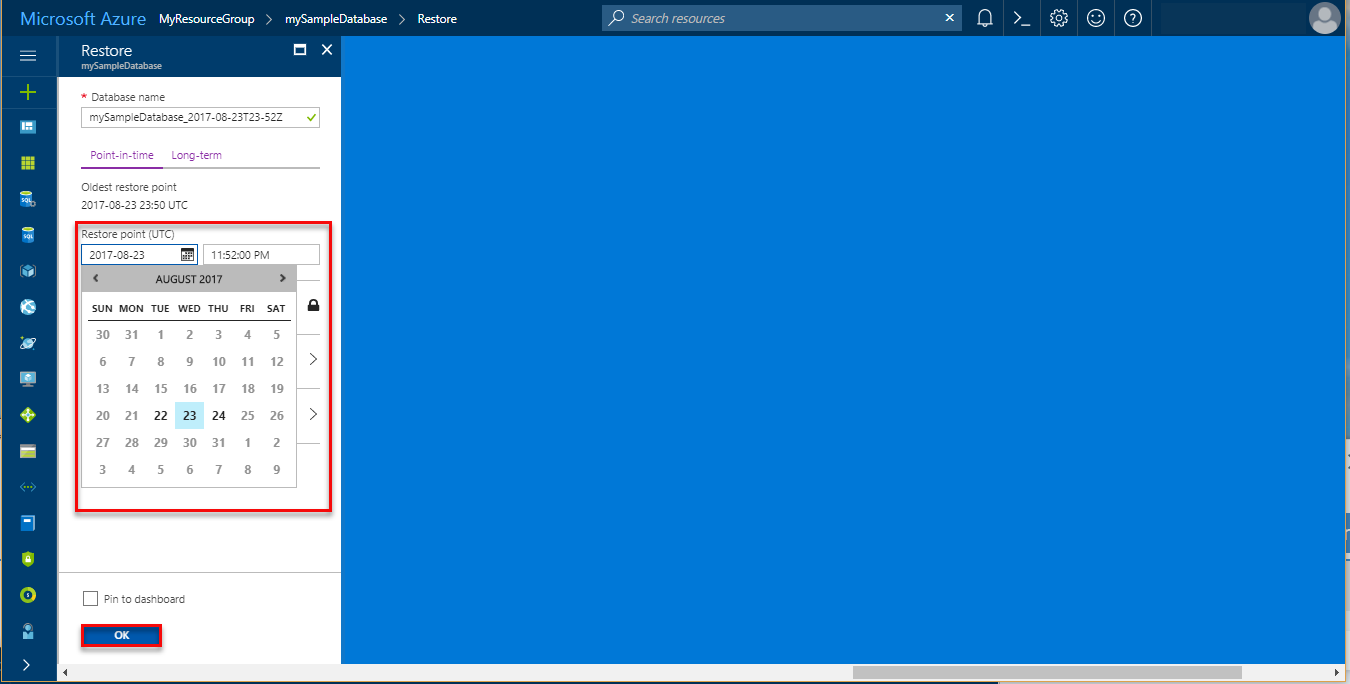
Restore a database to a previous point in time

Imagine you have accidentally deleted a table. This is something you cannot easily recover from. Azure SQL Database allows you to go back to any point in time in the last up to 35 days and restore this point in time to a new database. You can you this database to recover your deleted data. The following steps restore the sample database to a point before the tables were added.

1. On the SQL Database page for your database, click **Restore** on the toolbar. The **Restore**page opens.



1. Fill out the **Restore** form with the required information:
   * Database name: Provide a database name
   * Point-in-time: Select the **Point-in-time** tab on the Restore form
   * Restore point: Select a time that occurs before the database was changed
   * Target server: You cannot change this value when restoring a database
   * Elastic database pool: Select **None**
   * Pricing tier: Select **20 DTUs** and **40 GB** of storage.



1. Click **OK** to restore the database to [restore to a point in time](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-recovery-using-backups#point-in-time-restore) before the tables were added. Restoring a database to a different point in time creates a duplicate database in the same server as the original database as of the point in time you specify, as long as it is within the retention period for your [service tier](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-service-tiers).

Next steps

In this tutorial, you learned basic database tasks such as create a database and tables, load and query data, and restore the database to a previous point in time. You learned how to:

* Create a database
* Set up a firewall rule
* Connect to the database with [SQL Server Management Studio](https://msdn.microsoft.com/library/ms174173.aspx) (SSMS)
* Create tables
* Bulk load data
* Query that data
* Restore the database to a previous point in time using SQL Database [point in time restore](https://docs.microsoft.com/en-in/azure/sql-database/sql-database-recovery-using-backups#point-in-time-restore)capabilities

Advance to the next tutorial to learn about designing a database using Visual Studio and C#.