Exercise 1: Working with Azure SQL Database

In this exercise, you will provision a sample database in Azure SQL database, and use Transact-SQL to query the data it contains.

Provision Azure SQL Database

To get started, you must provision Azure SQL Database.

1. In the Microsoft Azure portal, in the menu, click **New**. Then in the **Databases** menu, click **SQL Database**.

* + 2. In the **SQL Database** blade, enter the following settings, and then click **Create**: • **Name**: AdventureWorksLT
  + • **Subscription**: *Select your Azure subscription*
  + • **Resource Group**: *Select the resource group you created previously*
  + • **Select source**: Sample (AdventureWorksLT)
  + • **Server:** *Create a new server with the following settings*: • **Server name**: *Enter a unique name (and make a note of it!)*
  + • **Server admin login:** *Enter a user name of your choice (and make a note of it!)*
  + • **Password:** *Enter and confirm a strong password (and make a note of it!)*
  + • **Location:** *Select any available region*
  + • **Allow azure services to access server**: Selected
  + • **Want to use SQL elastic pool?** Not now.
  + • **Pricing tier:** Basic
  + • **Collation**: SQL\_Latin1\_General\_CP1\_CI\_AS
  + • **Pin to dashboard:** Unselected

3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the SQL database to be deployed (this can take a few minutes.)

4. After the database has been created, browse to your Azure SQL server (not the AdventureWorksLT database) and under **Settings**, click **Properties**.

5. Note the fully qualified name of your server (which should take the form *server*.database.windows.net, where *server* is the server name you specified earlier) and the server admin user name (which should be the login you specified earlier).

Query a Table

A relational database contains tables, each of which contains data. Tables are organized into namespaces called *schemas* – in the case of the **AdventureWorksLT** sample database, most of the tables are defined within a schema named **SalesLT**.

You can query tables using Transact-SQL to retrieve the data they contain.

1. Click **All Resources**, and then click the **AdventureWorksLT** database.

2. On the **AdventureWorksLT** blade, view the **Data Explorer** page. This opens the web-based query interface for your Azure SQL Database.

3. In the toolbar for the query editor, click **Login**, and then log into your database using SQL Server authentication and entering the login name and password you specified when provisioning the Azure SQL Database server.

4. In the query editor, enter the following Transact-SQL query to retrieve the contents of the **SalesLT.Product** table in the **AdventureWorksLT** database:

SELECT \* FROM SalesLT.Product;

6. Click **Run**, and review the results.

Exercise 2: Loading Data into a Database

In this exercise, you will create a table in the sample database you created previously, and then use Azure Data Factory to copy data from a file in Azure Storage into the new table.

Create a Table

The sample database contains many tables, and you can add your own by using the Transact-SQL CREATE TABLE statement.

1. In the Query pane, replace the existing SELECT statement with the following code:

CREATE TABLE SalesLT.ProductReview

( ProductReviewID INTEGER PRIMARY KEY,

ProductID INTEGER REFERENCES SalesLT.Product(ProductID),

ReviewerName NVARCHAR(25),

ReviewDate DATETIME,

EmailAddress NVARCHAR(50),

Rating INTEGER,

Comments NTEXT );

2. Click **Run**, and verify that the query succeeds.

3. Replace the CREATE TABLE statement with the following SELECT statement:

SELECT \* FROM SalesLT.ProductReview;

4. Click **Run**, and verify that the query succeeds but returns 0 rows.

5. Close the query editor without saving any changes.