DP 200 - Implementing a Data Platform Solution

Lab 5 - Working with Relational Data Stores in the Cloud

Estimated Time: 75 minutes

Pre-requisites: It is assumed that the case study for this lab has already been read. It is assumed that the content and lab for module 1: Azure for the Data Engineer has also been completed

Lab files: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.5* folder.

Lab overview

The students will be able to provision an Azure SQL Database and Azure Synapse Analytics server and be able to issue queries against one of the instances that are created. They will be also be able to integrate a data warehouse with a number of other data platform technologies and use PolyBase to load data from one data source into Azure Synapse Analytics.

Lab objectives

After completing this lab, you will be able to:

1. Use Azure SQL Database
2. Describe Azure Synapse Analytics
3. Create and query Azure Synapse Analytics
4. Use PolyBase to load data into Azure Synapse Analytics

Scenario

You are the senior data engineer at AdventureWorks, and you are working with your team to transition a relational database system from an on-premises SQL Server to a Azure SQL Database located in Azure. You will begin by creating an instance of Azure SQL Database with the company's sample database. Your intention is to hand this instance of to a junior data engineer to perform some testing of departmental databases.

You will then provision Azure Synapse Analytics server and test that the provisioning of the server is successful by testing a sample database with a series of queries. You will then use PolyBase to load a dimension table from Azure Blob to test that the integration of this data platform technology with Azure Synapse Analytics.

At the end of this lad, you will have:

1. Used Azure SQL Database
2. Described Azure Synapse Analytics
3. Created and queryied Azure Synapse Analytics
4. Used PolyBase to doad data into Azure Synapse Analytics

> **IMPORTANT**: As you go through this lab, make a note of any issue(s) that you have encountered in any provisioning or configuration tasks and log it in the table in the document located at *\Labfiles\DP-200-Issues-Doc.docx*. Document the Lab number, note the technology, Describe the issue, and what was the resolution. Save this document as you will refer back to it in a later module.

Exercise 1: Use Azure SQL Database

Estimated Time: 15 minutes

Individual exercise

The main task for this exercise are as follows:

1. Create and configure a SQL Database instance.

Task 1: Create and configure a SQL Database instance.

1. In the Azure portal, navigate to the + **Create a resource** blade.

2. In the New screen, click the **Search the Marketplace** text box, and type the word **SQL Database**. Click **SQL Database** in the list that appears.

3. In the **SQL Database** screen, click **Create**.

4. From the **Create SQL Database** screen, create an Azure SQL Database with the following settings:

   - In the Project details section, type in the following information

     - **Subscription**: the name of the subscription you are using in this lab

     - **Resource group**: **awrgstudxx**, where **xx** are your initials.

   - Click on the **Additional setting** tab, click **Sample** . The AdventureworksLT sample database is selected automatically.

   - Click the **Basics** tab once this has been done.

   - In the Database details section, type in the following information

     - Database name: type in **AdventureworksLT**

     - Server: Create a new server by clicking **Create new** with the following settings and click on **OK**:

       - **Server name**: **sqlservicexx**, where **xx** are your initials

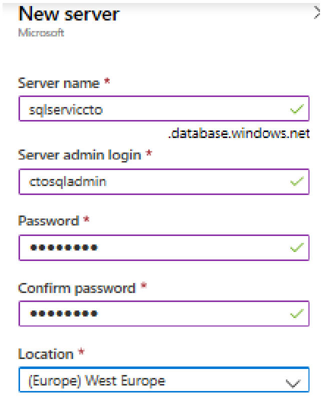
       - **Server admin login**: **xxsqladmin**, where **xx** are your initials

       - **Password**: **Pa55w.rd**

       - **Confirm Password**: **Pa55w.rd**

       - **Location**: choose a **location** near to you.

       - click on **OK**



New server
Microsoft

Server name *
sqlserviccto
.database.windows.net

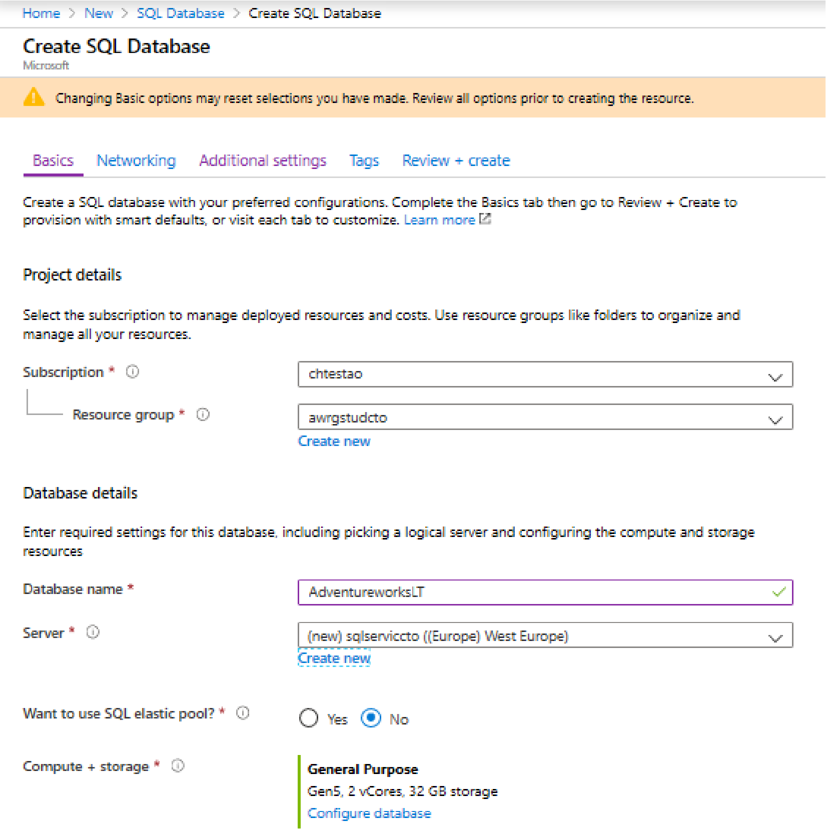
Server admin login *
ctosqladmin

Password *
••••••••

Confirm password *
••••••••

Location *
(Europe) West Europe

       - Leave the remaining settings to their defaults, and then click on **OK**



Create SQL Database

Microsoft

⚠ Changing Basic options may reset selections you have made. Review all options prior to creating the resource.

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: chtestao

Resource group *: awrgstudcto

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name *: AdventureworksLT

Server *: (new) sqlserviccto ((Europe) West Europe)

Create new

Want to use SQL elastic pool? *: ○ Yes ● No

Compute + storage *: **General Purpose**
Gen5, 2 vCores, 32 GB storage
Configure database

5. In the **Create SQL Database** blade, click **Review + create**.

6. After the validation of the **Create SQL Database*** blade, click **Create**.

   > **Note**: The provision will takes approximately 4 minutes.

> **Result**: After you completed this exercise, you have an Azure SQL Database instance

Exercise 2: Describe Azure Synapse Analytics

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Create and configure a Azure Synapse Analytics instance.

2. Configure the Server Firewall

3. Pause the warehouse database

Task 1: Create and configure a Azure Synapse Analytics instance.

1. In the Azure portal, click on the link **home** at the top left of the screen.

2. In the Azure portal, click **+ Create a resource**.

3. In the New blade, navigate to the **Search the Marketplace** text box, and type the word **Synapse**. Click **Azure Synapse Analytics (formerly SQL DW)** in the list that appears.

4. In the **Azure Synapse Analytics (formerly SQL DW)** blade, click **Create**.

5. From the **SQL Data Warehouse** blade, create an Azure Synapse Analytics with the following settings:

   - In the Project details section, type in the following information

     - **Subscription**: the name of the subscription you are using in this lab

        - **Resource group**: **awrgstudxx**, where **xx** are your initials.

    - In **Additional setting** tab, under data source, click **Sample**.

    - Click the **Basics** tab once this has been done.

    - In the Database details section, type in the following information

        - **Database warehouse name**: **Warehousexx**, where **xx** are your initials.

        - **Server**: Create a new server by clicking **Create new** with the following settings and click on **OK**:

            - **Server name**: **dwhservicexx**, where **xx** are your initials

            - **Server admin login**: **xxsqladmin**, where **xx** are your initials

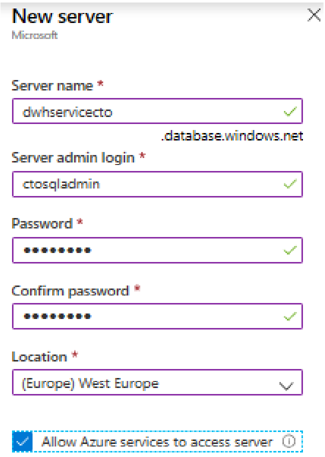
            - **Password**: **Pa55w.rd**

            - **Confirm Password**: **Pa55w.rd**

            - **Location**: choose a **location** near to you.

            - Select the checkbox to Allow Azure services to access server

            - click on **OK**



New server
Microsoft

Server name *
dwhservicecto
.database.windows.net

Server admin login *
ctosqladmin

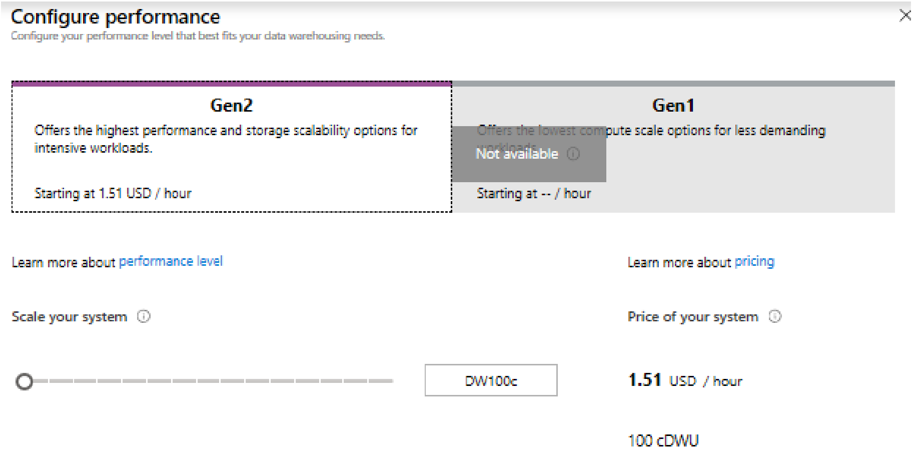
Password *
••••••••

Confirm password *
••••••••

Location *
(Europe) West Europe

☑ Allow Azure services to access server

    - Performance Level: Click **Select performance level** and select **Gen2 DW100C**.



Configure performance
Configure your performance level that best fits your data warehousing needs.

Gen2
Offers the highest performance and storage scalability options for intensive workloads.
Starting at 1.51 USD / hour

Gen1
Offers the lowest compute scale options for less demanding workloads.
Not available
Starting at -- / hour

Learn more about performance level

Learn more about pricing

Scale your system

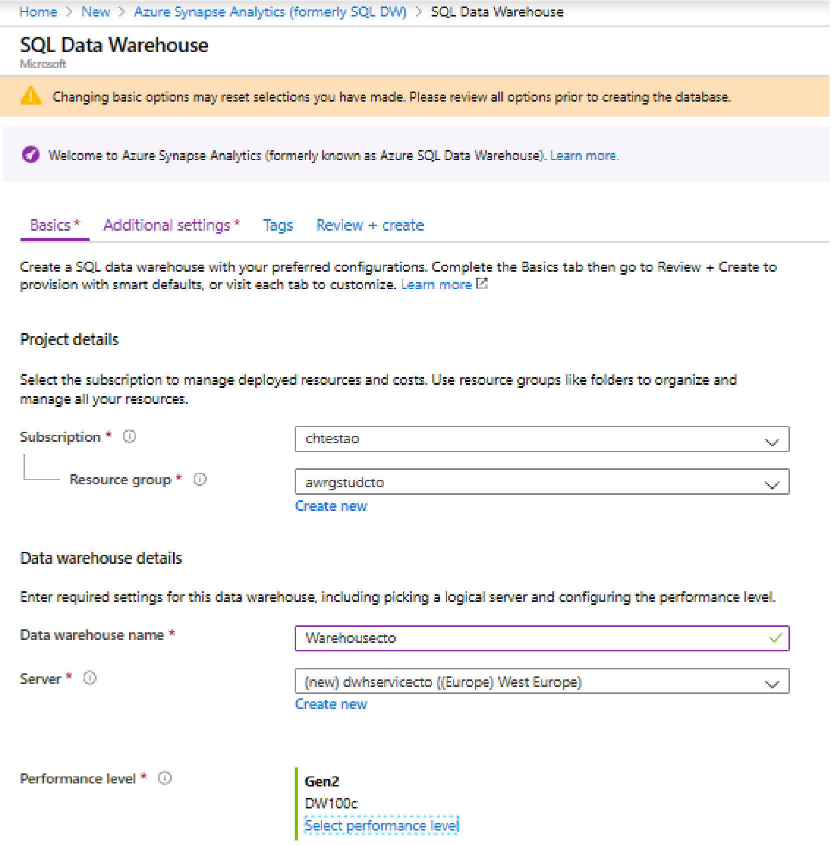
DW100c

Price of your system

1.51 USD / hour

100 cDWU

    - Click **Apply**. the following configuration is shown.


Home > New > Azure Synapse Analytics (formerly SQL DW) > SQL Data Warehouse
Basics * | Additional settings * | Tags | Review + create
Create new
Create new
Select performance level

SQL Data Warehouse
Microsoft

Changing basic options may reset selections you have made. Please review all options prior to creating the database.

Welcome to Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse). Learn more.

Create a SQL data warehouse with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: chtestao

Resource group *: awrgstudcto

Data warehouse details

Enter required settings for this data warehouse, including picking a logical server and configuring the performance level.

Data warehouse name *: Warehousecto

Server *: (new) dwhservicecto ((Europe) West Europe)

Performance level *: Gen2 DW100c

6. In the **SQL Data Warehouse** screen, click **Review + create**.

7. In the **SQL Data Warehouse** blade, click **Create**.

> **Note**: The provision will takes approximately 7 minutes.

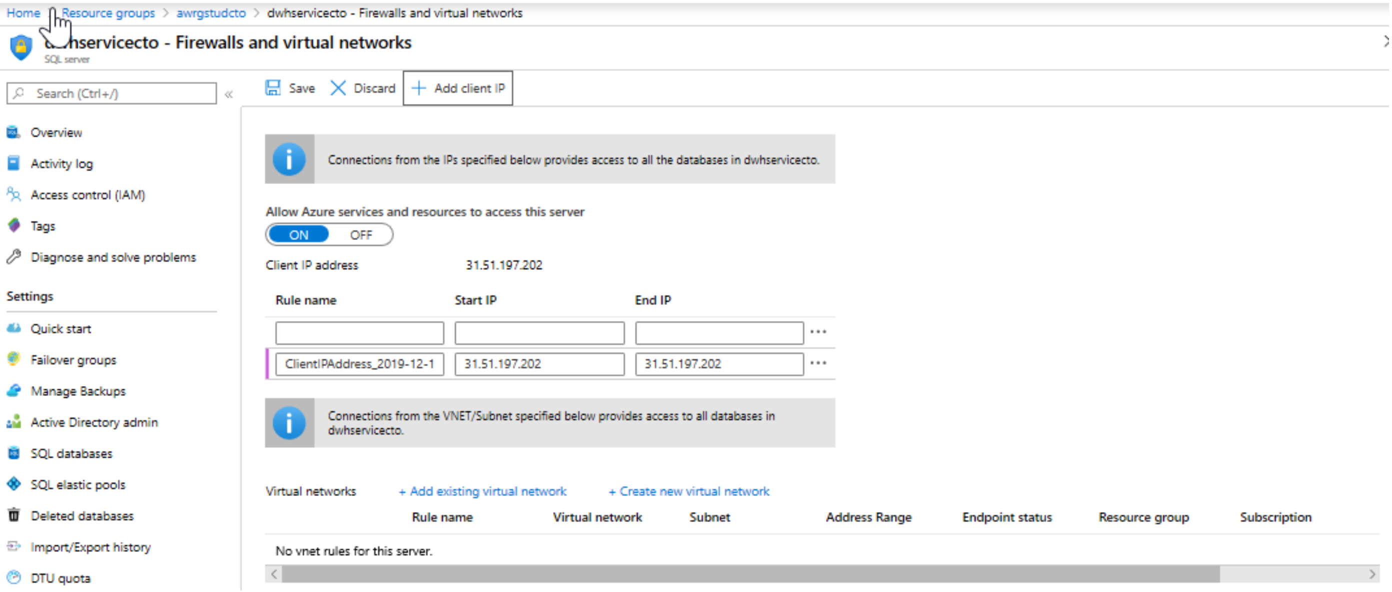
Task 2: Configure the Server Firewall

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awdlsstudxx**, where **xx** are your initials

2. Click on **dwhservicexx**, where **xx** are your initials.

3. In the **dwhservicexx** screen, click on **Firewalls and virtual networks**.

4. In the dwhservicexx - Firewalls and virtual networks screen, click on the option **+ Add client IP**, and then click on **Save**. On the success screen click **OK**.



dwhservicecto - Firewalls and virtual networks

Connections from the IPs specified below provides access to all the databases in dwhservicecto.

Allow Azure services and resources to access this server: ON

Client IP address 31.51.197.202

Rule name	Start IP	End IP
ClientIPAddress_2019-12-1	31.51.197.202	31.51.197.202

Connections from the VNET/Subnet specified below provides access to all databases in dwhservicecto.

Virtual networks

Rule name	Virtual network	Subnet	Address Range	Endpoint status	Resource group	Subscription

No vnet rules for this server.

> **Note**: You will receive a message stating that the the server firewall rules have been successfully updated

5. Close down the Firewalls and virtual networks screen.

> **Result**: After you completed this exercise, you have created an Azure Synapse Analytics instance and configures the server firewall to enable connections against it.

Task 3: Pause the Warehousexx database

1. Click on **Warehousexx**, where **xx** are your initials.

2. In the **Warehousexx (dwhservicexx/Warehousexx)** screen, click on **Pause**.

3. In the Pause Warehousexx screen, click **Yes**

Exercise 3: Creating an Azure Synapse Analytics database and tables

Estimated Time: 25 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Install SQL Server Management Studio and connect to a data warehouse instance.

2. Create a SQL Data Warehouse database

3. Create SQL Data Warehouse tables

> **Note**: If you are not familiar with Transact-SQL, statements are available for the following labs in the following location **Allfiles\Labfiles\Starter\DP-200.5\SQL DW Files**

Task 1: Install SQL Server Management Studio and connect to a SQL Data Warehouse instance.

1. In the Azure Portal, in the **dwhservicexx - Firewalls and virtual networks**, in the blade, click on **Properties**

2. Copy the **"Server name"** and paste it into Notepad.

3. Download SQL Server Management Studio and install onto your machine

4. On the windows desktop, click on the **Start**, and type **"SQL Server"** and then click on **MIcrosoft SQL Server Management Studio 17**

5. In the **Connect to Server** dialog box, fill in the following details

   - Server Name: **dwhservicexx.database.windows.net**
   - Authentication: **SQL Server Authentication**
   - Username: **xxsqladmin**
   - Password: **Pa55w.rd**

6. In the **Connect to Server** dialog box, click **Connect**

Task 2: Create a SQL Data Warehouse database.

1. In **SQL Server Management Studio**, in Object Explorer, right click **dwhservicexx.database.windows.net** and click on **New Query**.

2. In the query window, create a DataWarehouse database named **DWDB**, with a service objective of DW100 and a maximum size of 1024GB.

```
CREATE DATABASE DWDB COLLATE SQL_Latin1_General_CP1_CI_AS
(
    EDITION             = 'DataWarehouse'
,   SERVICE_OBJECTIVE   = 'DW100C'
,   MAXSIZE             = 1024 GB
);
```

> **Note**: The creation of the database takes approximately 2 minutes.

Task 3: Create SQL Data Warehouse tables.

1. In **SQL Server Management Studio**, in Object Explorer, right click **dwhservicexx.database.windows.net** and click on **New Query**.

2. In **SQL Server Management Studio**, in SQL Editor toolbar, in **Available Databases**, click on **DWDB**.

   > **Note**: If you are unfamiliar with Transact-SQL, there is a script in the Allfiles\Solution\DP-200.5\folder named **Exercise3 Task3Step2 script.sql**. It contains the bulk of the code required to create the tables, but you do have to complete the code by selecting the distribution type to use for each table

3. Create a table named **dbo.Users** with a **clustered columnstore** index with a distribution of **replicate** with the following columns:

column name	data type	Nullability
userId	int	NULL
City	nvarchar(100)	NULL
Region	nvarchar(100)	NULL
Country	nvarchar(100)	NULL

4. In **SQL Server Management Studio**, click on **Execute**.

5. In **SQL Server Management Studio**, in Object Explorer, right click **dwhservicexx.database.windows.net** and click on **New Query**.

6. In **SQL Server Management Studio**, in SQL Editor toolbar, in **Available Databases**, click on **DWDB**.

7. Create a table named **dbo.Products** with a **clustered columnstore** index with a distribution of **round robin** with the following columns:

column name	data type	Nullability
ProductId	int	NULL
EnglishProductName	nvarchar(100)	NULL
Color	nvarchar(100)	NULL
StandardCost	int	NULL
ListPrice	int	NULL
Size	nvarchar(100)	NULL
Weight	int	NULL
DaysToManufacture	int	NULL
Class	nvarchar(100)	NULL
Style	nvarchar(100)	NULL

8. In **SQL Server Management Studio**, click on **Execute**.

9. In **SQL Server Management Studio**, in Object Explorer, right click **dwhservicexx.database.windows.net** and click on **New Query**.

10. In **SQL Server Management Studio**, in SQL Editor toolbar, in **Available Databases**, click on **DWDB**.

11. Create a table named **dbo.FactSales** with a **clustered columnstore** index with a distribution of **Hash** on the **SalesUnit** with the following columns:

column name	data type	Nullability
DateId	int	NULL
ProductId	int	NULL
UserId	int	NULL

column name	data type	Nullability
UserPreferenceId	int	NULL
SalesUnit	int	NULL

12. In **SQL Server Management Studio**, click on **Execute**.

> **Result**: After you completed this exercise, you have installed SQL Server Management Studio to create a data warhouse named DWDB and three tables named Users, Products and FactSales.

Exercise 4: Using PolyBase to Load Data into Azure Synapse Analytics

Estimated Time: 10 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Collect Data Lake Storage container and key details

2. Create a dbo.Dates table using PolyBase from Azure Data Lake Storage

Task 1: Collect Azure Blob account name and key details

1. In the Azure portal, click on **Resource groups** and then click on **awrgstudxx**, and then click on **awdlsstudxx** where xx are the initials of your name.

2. In the **awdlsstudxx** screen, click **Access keys**. Click on the icon next to the **Storage account name** and paste it into Notepad.

3. In the **awdlsstudxx - Access keys** screen, under **key1**, Click on the icon next to the **Key** and paste it into Notepad.

Task 2: Create a dbo.Dates table using PolyBase from Azure Blob

1. In **SQL Server Management Studio**, in Object Explorer, right click **dwhservicexx.database.windows.net** and click on **New Query**.

2. In **SQL Server Management Studio**, in SQL Editor toolbar, in **Available Databases**, click on **DWDB**.

3. Create a **master key** against the **DWDB** database. In the query editor, type in the following code:

```
CREATE MASTER KEY;
```

4. Create a database scoped credential named **AzureStorageCredential** with the following details, by typing in the following code:

   - IDENTITY: **MOCID**
   - SECRET: **The access key of your storage account**

```
CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
IDENTITY = 'MOCID',
SECRET = 'Your storage account key'
```

; ```

1. In **SQL Server Management Studio**, highlight both statements and then click on **Execute**.

2. In **SQL Server Management Studio**, in the Query window, type in code that will create an external data source named **AzureStorage** for the Blob storage account and data container created in with a type of **HADOOP** that makes use of the ****AzureStorageCredential**. Note that you should replace **awdlsstudxx** in the location key with your storage account with your initials

```
CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
    TYPE = HADOOP,
    LOCATION = 'abfs://data@awdlsstudxx.dfs.core.windows.net',
```

```
    CREDENTIAL = AzureStorageCredential
);
```

3. In **SQL Server Management Studio**, in the Query window, type in code that will create an external file format named **TextFile** with a formattype of **DelimitedText** and a filed terminator of **comma**.

```
CREATE EXTERNAL FILE FORMAT TextFile
WITH (
    FORMAT_TYPE = DelimitedText,
    FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
);
```

4. In **SQL Server Management Studio**, highlight the statement and then click on **Execute**.

5. In **SQL Server Management Studio**, in the Query window, type in code that will create an external table named **dbo.DimDate2External** with the **location** as the root file, the Data source as **AzureStorage**, the File_format of **TextFile** with the following columns:

column name	data type	Nullability
Date	datetime2(3)	NULL
DateKey	decimal(38, 0)	NULL
MonthKey	decimal(38, 0)	NULL
Month	nvarchar(100)	NULL
Quarter	nvarchar(100)	NULL
Year	decimal(38, 0)	NULL
Year-Quarter	nvarchar(100)	NULL
Year-Month	nvarchar(100)	NULL
Year-MonthKey	nvarchar(100)	NULL
WeekDayKey	decimal(38, 0)	NULL
WeekDay	nvarchar(100)	NULL
Day Of Month	decimal(38, 0)	NULL

```
CREATE EXTERNAL TABLE dbo.DimDate2External (
[Date] datetime2(3) NULL,
[DateKey] decimal(38, 0) NULL,
[MonthKey] decimal(38, 0) NULL,
[Month] nvarchar(100) NULL,
[Quarter] nvarchar(100) NULL,
[Year] decimal(38, 0) NULL,
[Year-Quarter] nvarchar(100) NULL,
[Year-Month] nvarchar(100) NULL,
[Year-MonthKey] nvarchar(100) NULL,
[WeekDayKey] decimal(38, 0) NULL,
[WeekDay] nvarchar(100) NULL,
[Day Of Month] decimal(38, 0) NULL
)
WITH (
    LOCATION='/DimDate2.txt',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

6. In **SQL Server Management Studio**, highlight the statement and then click on **Execute**.

7. Test that the table is created by running a select statement against it

```
SELECT * FROM dbo.DimDate2External;
```

8. In **SQL Server Management Studio**, in the Query window, type in a **CTAS** statement that creates a table named **dbo.Dates** with a **columnstore** index and a **distribution** of **round robin** that loads data from the **dbo.DimDate2External** table.

```
CREATE TABLE dbo.Dates
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT * FROM [dbo].[DimDate2External];
```

9. In **SQL Server Management Studio**, highlight the statement and then click on **Execute**.

10. In **SQL Server Management Studio**, in the Query window, type in a query that creates statistics on the **DateKey**, **Quarter** and **Month** column.

```
CREATE STATISTICS [DateKey] on [Dates] ([DateKey]);
CREATE STATISTICS [Quarter] on [Dates] ([Quarter]);
CREATE STATISTICS [Month] on [Dates] ([Month]);
```

11. Test that the table is created by running a select statement against it

```
SELECT * FROM dbo.Dates;
```