Data Processing with U-SQL – Data Virtualization

Updated on: 1/4/2017

# Introduction

By **Data Virtualization** we mean the ability for U-SQL to query multiple structured datasources in Azure in a single U-SQL script.

Today the datasources include:

* The U-SQL Catalog
* Azure SQL DB
* Azure SQL DW

We sometimes use different (older) names to refer to Data Virtualization. These include:

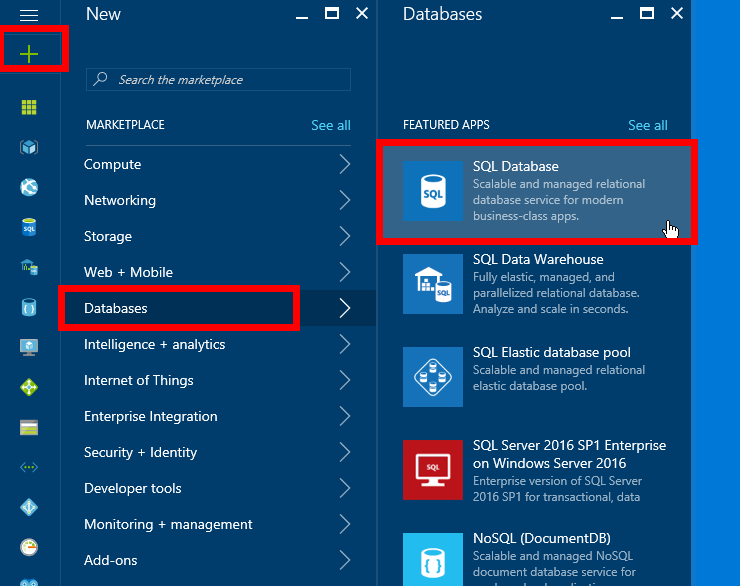
* “U-SQL Distributed Query”
* “U-SQL Federated Query”

The canonical use for Data Virtualization is when you already have a process in place to standardize dimensional data for use across your environments. For example you have a Product Catalog in an Azure SQL DB and which to use that data in your U-SQL Code.

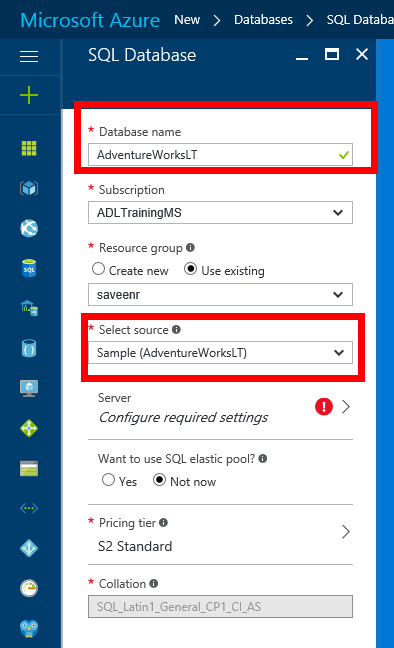
# Setting up your Azure SQL DB

In this section, you’ll get a SQL DB ready with the AdventureWorks data.

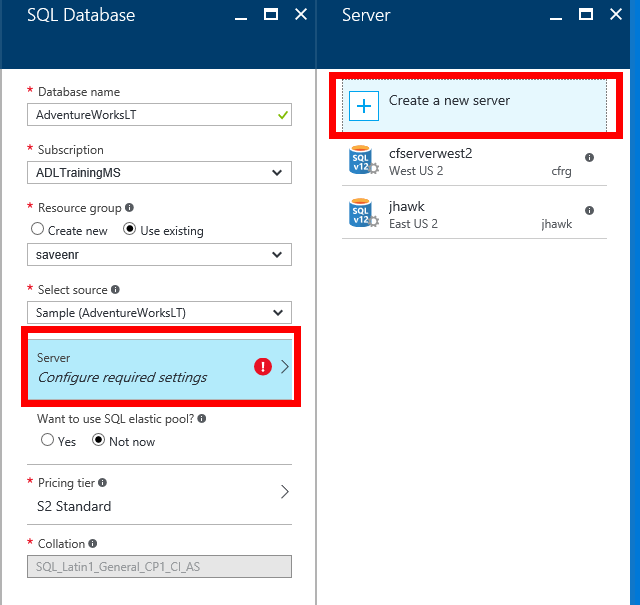
* Log into the Azure portal and click + > Databases > SQL Database



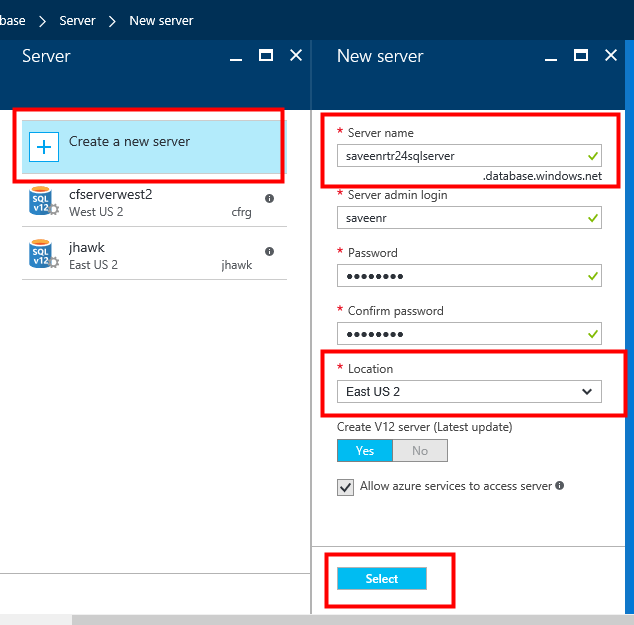
* For **Database name** enter **AdventureWorksLT**
* Under **Select source** choose **Sample (AdventureWorksLT)**



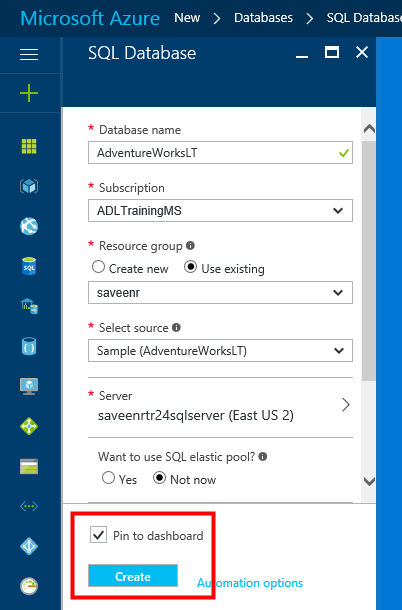
Click **Server** choose **Create a new server**



* Pick a server name
* Choose any values you desire for **Server admin login** and **Password**. Don’t forget these values.
* Make sure that Location is **EAST US 2**
* Click **Select**



* Enable **Pin to dashboard**
* Click **Create**



* Wait a few minutes for this process to complete.
* Once it is done you will have an Azure SQL Server (with a name you picked) that hosts a single Azure SQL DB called **AdventureWorksLT**

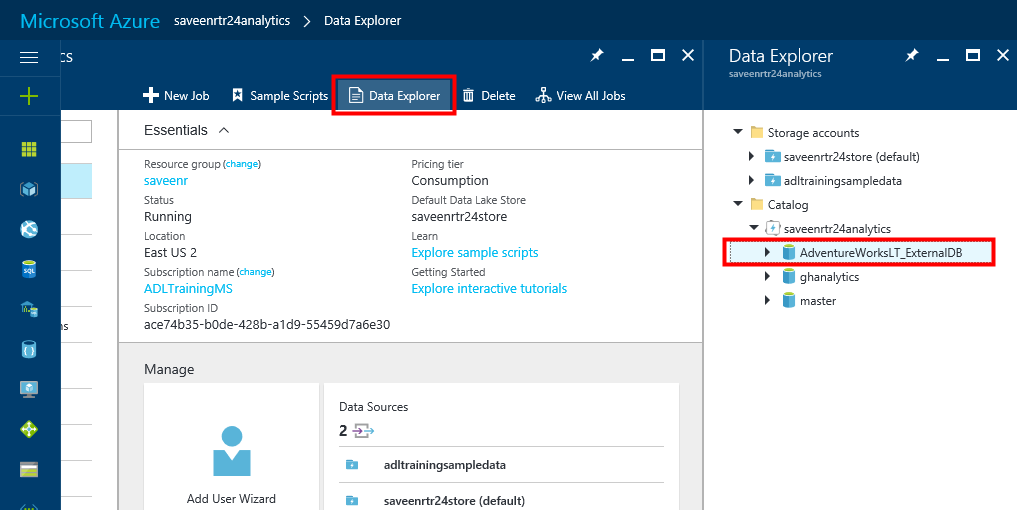
# Create a USQL Database

U-SQL Data Virtualization requires a developer to the datasources. That description lives in a U-SQL Database.

* Create a database called **AdventureWorksLT\_ExternalDB**

CREATE DATABASE IF NOT EXISTS [AdventureWorksLT\_ExternalDB];

* Verify the database exists before proceeding



# Storing Credentials for the External Data Source

* Launch PowerShell ISE
* Use **Get-ExecutionPolicy** to verify that the execution policy is set to **Unrestricted**.
* If you need to set it to Unrestricted. Close your PowerShell session. Launch a new PowerShell session as Administrator and then run this command

Set-ExecutionPolicy Unrestricted

* Login to your Azure Subscription

Login-AzureRmAccount -SubscriptionName “Your Subscription”

* Run the following script.

$username = "yourusername"

$passwd = ConvertTo-SecureString "yourpassword" -AsPlainText -Force

$creds = New-Object System.Management.Automation.PSCredential($username, $passwd)

New-AdlCatalogCredential -Account "youradlaaccount" `

-DatabaseName "AdventureWorksLT\_ExternalDB" `

-DatabaseHost "saveenrtr24sqlserver.database.windows.net" `

-Port 1433 `

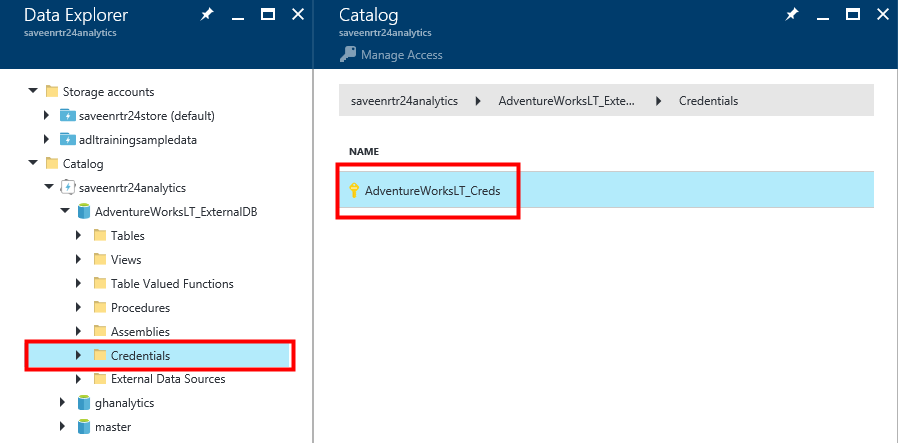
-CredentialName "AdventureWorksLT\_Creds" `

-Credential $creds

Verify the credentials exist with this script

Get-AdlCatalogItem -Account "YOURADLAACCOUNT" -ItemType Credential -Path "AdventureWorksLT\_ExternalDB"

Or use the Data Explorer to see it in the Portal



# Create an External Data Source

The following script will use the credentials that you registered in the previous step to create a credential in your U-SQL database and then create the Data Source using those credentials.

USE DATABASE [AdventureWorksLT\_ExternalDB];

CREATE DATA SOURCE IF NOT EXISTS AdventureWorksLT\_DS

FROM AZURESQLDB WITH

( PROVIDER\_STRING = "Database=AdventureWorksLT;Trusted\_Connection=False;Encrypt=True",

CREDENTIAL = AdventureWorksLT\_Creds,

REMOTABLE\_TYPES = (bool, byte, sbyte, short, ushort, int, uint, long, ulong, decimal, float, double, string, DateTime)

);

# Read from the External Data Source: Part 1

Submit the following U-SQL Query

USE DATABASE [AdventureWorksLT\_ExternalDB];

@customers =

SELECT \*

FROM EXTERNAL AdventureWorksLT\_DS LOCATION "[SalesLT].[Customer]";

OUTPUT @customers

TO @"/SalesLT\_Customer.csv"

USING Outputters.Csv();

# Copy from the External Data Source into a U-SQL Table

Although Data Virtualization is a great feature, as the amount of data grows you and the number of queries increase may find it performance will improve if the data is cached locally in a U-SQL database.

* Run this script below to copy the table. Notice that the U-SQL Table below will get its schema based on the rowset @customers. We do however have to specify how the table is indexed.

USE DATABASE [AdventureWorksLT\_ExternalDB];

@customers =

SELECT \*

FROM EXTERNAL AdventureWorksLT\_DS LOCATION "[SalesLT].[Customer]";

CREATE TABLE CustomersLocal

(

INDEX customers\_idx

CLUSTERED (CustomerID ASC)

PARTITIONED BY HASH (CustomerID)

) AS SELECT \* FROM @customers;

* You can verify the table exists in the catalog with this script

Get-AdlCatalogItem -Account "Your ADL Account" -ItemType Table -Path "AdventureWorksLT\_ExternalDB.dbo"

# Read from the External Data Source: Part 2 Using an External Table

Notice that the previous example where we read from the external datasource, that nowhere in AdventureWorksLT\_ExternalDB is there any mention of the schema of the external data.

While this is very flexible, sometimes it is also convenient to make sure the schema of the remote data is known. We can do this through an External Table. An External Table requires us to specify the schema of the external table explicitly.

Submit the following U-SQL script into the editor of your choice:

USE DATABASE [AdventureWorksLT\_ExternalDB];

CREATE EXTERNAL TABLE IF NOT EXISTS dbo.CustomersExternal

(

CustomerID int?,

NameStyle bool,

Title string,

FirstName string,

MiddleName string,

LastName string,

Suffix string,

CompanyName string,

SalesPerson string,

EmailAddress string,

Phone string,

PasswordHash string,

PasswordSalt string,

Rowguid Guid,

ModifiedDate DateTime?

)

FROM AdventureWorksLT\_DS LOCATION "[SalesLT].[Customer]";

* You can verify the external table exists in the catalog with this script

Get-AdlCatalogItem -Account "Your ADL Account" -ItemType Table -Path "AdventureWorksLT\_ExternalDB.dbo"