|  |
| --- |
| if db\_name() = 'master' |
|  | throw 50001, 'This script cannot be executed in master database. Create new database and run the script there.', 1; |
|  |  |
|  | if SERVERPROPERTY('EngineEdition') <> 11 |
|  | throw 50001, 'This script must be executed on Azure Synapse - Serverless SQL endpoint.', 1; |
|  |  |
|  | ------------------------------------------------------------------------------------------ |
|  | -- Part 1 - Cleanup script |
|  | -- This part removes objects from sample database |
|  | ------------------------------------------------------------------------------------------ |
|  | DROP VIEW IF EXISTS parquet.YellowTaxi |
|  | GO |
|  | DROP VIEW IF EXISTS json.Books |
|  | GO |
|  | DROP VIEW IF EXISTS csv.YellowTaxi |
|  | GO |
|  | DROP VIEW IF EXISTS cosmosdb.Ecdc |
|  | GO |
|  | IF (EXISTS(SELECT \* FROM sys.external\_tables WHERE name = 'Population')) BEGIN |
|  | DROP EXTERNAL TABLE csv.Population |
|  | END |
|  | IF (EXISTS(SELECT \* FROM sys.external\_tables WHERE name = 'Covid')) BEGIN |
|  | DROP EXTERNAL TABLE delta.Covid |
|  | END |
|  | IF (EXISTS(SELECT \* FROM sys.external\_file\_formats WHERE name = 'QuotedCsvWithHeader')) BEGIN |
|  | DROP EXTERNAL FILE FORMAT QuotedCsvWithHeader |
|  | END |
|  | GO |
|  | IF (EXISTS(SELECT \* FROM sys.external\_file\_formats WHERE name = 'QuotedCsvWithoutHeader')) BEGIN |
|  | DROP EXTERNAL FILE FORMAT QuotedCsvWithoutHeader |
|  | END |
|  | GO |
|  | IF (EXISTS(SELECT \* FROM sys.external\_file\_formats WHERE name = 'NativeParquet')) BEGIN |
|  | DROP EXTERNAL FILE FORMAT NativeParquet |
|  | END |
|  | GO |
|  | IF (EXISTS(SELECT \* FROM sys.external\_file\_formats WHERE name = 'DeltaLakeFormat')) BEGIN |
|  | DROP EXTERNAL FILE FORMAT DeltaLakeFormat |
|  | END |
|  | GO |
|  | DROP SCHEMA IF EXISTS parquet; |
|  | GO |
|  | DROP SCHEMA IF EXISTS csv; |
|  | GO |
|  | DROP SCHEMA IF EXISTS json; |
|  | GO |
|  | DROP SCHEMA IF EXISTS cosmosdb; |
|  | GO |
|  | DROP SCHEMA IF EXISTS delta; |
|  | GO |
|  | IF (EXISTS(SELECT \* FROM sys.external\_data\_sources WHERE name = 'SqlOnDemandDemo')) BEGIN |
|  | DROP EXTERNAL DATA SOURCE SqlOnDemandDemo |
|  | END |
|  |  |
|  | IF (EXISTS(SELECT \* FROM sys.external\_data\_sources WHERE name = 'DeltaLakeStorage')) BEGIN |
|  | DROP EXTERNAL DATA SOURCE DeltaLakeStorage |
|  | END |
|  |  |
|  | IF (EXISTS(SELECT \* FROM sys.external\_data\_sources WHERE name = 'AzureOpenData')) BEGIN |
|  | DROP EXTERNAL DATA SOURCE AzureOpenData |
|  | END |
|  |  |
|  | IF (EXISTS(SELECT \* FROM sys.external\_data\_sources WHERE name = 'YellowTaxi')) BEGIN |
|  | DROP EXTERNAL DATA SOURCE YellowTaxi |
|  | END |
|  |  |
|  | IF (EXISTS(SELECT \* FROM sys.external\_data\_sources WHERE name = 'GreenTaxi')) BEGIN |
|  | DROP EXTERNAL DATA SOURCE GreenTaxi |
|  | END |
|  |  |
|  | IF NOT EXISTS (SELECT \* FROM sys.symmetric\_keys) BEGIN |
|  | declare @pasword nvarchar(400) = CAST(newid() as VARCHAR(400)); |
|  | EXEC('CREATE MASTER KEY ENCRYPTION BY PASSWORD = ''' + @pasword + '''') |
|  | END |
|  |  |
|  | IF EXISTS |
|  | (SELECT \* FROM sys.credentials |
|  | WHERE name = 'https://sqlondemandstorage.blob.core.windows.net') |
|  | DROP CREDENTIAL [https://sqlondemandstorage.blob.core.windows.net] |
|  | GO |
|  |  |
|  | IF EXISTS |
|  | (SELECT \* FROM sys.credentials |
|  | WHERE name = 'MyCosmosDbAccountCredential') |
|  | DROP CREDENTIAL [MyCosmosDbAccountCredential] |
|  | GO |
|  |  |
|  | IF EXISTS |
|  | (SELECT \* FROM sys.database\_scoped\_credentials |
|  | WHERE name = 'sqlondemand') |
|  | DROP DATABASE SCOPED CREDENTIAL [sqlondemand] |
|  | GO |
|  |  |
|  | IF EXISTS |
|  | (SELECT \* FROM sys.database\_scoped\_credentials |
|  | WHERE name = 'WorkspaceIdentity') |
|  | DROP DATABASE SCOPED CREDENTIAL [WorkspaceIdentity] |
|  | GO |
|  |  |
|  |  |
|  | ------------------------------------------------------------------------------------------ |
|  | -- Part 2 - initialization script |
|  | -- This part creates required objects in sample database |
|  | ------------------------------------------------------------------------------------------ |
|  |  |
|  | -- create database-scoped credential for the containers in demo storage account |
|  | -- this credential will be used in OPENROWSET function with data source that uses relative file URL |
|  | CREATE DATABASE SCOPED CREDENTIAL [sqlondemand] |
|  | WITH IDENTITY='SHARED ACCESS SIGNATURE', |
|  | SECRET = 'sv=2018-03-28&ss=bf&srt=sco&sp=rl&st=2019-10-14T12%3A10%3A25Z&se=2061-12-31T12%3A10%3A00Z&sig=KlSU2ullCscyTS0An0nozEpo4tO5JAgGBvw%2FJX2lguw%3D' |
|  | GO |
|  | -- Create credential that will allow user to impersonate using Managed Identity assigned to workspace |
|  | CREATE DATABASE SCOPED CREDENTIAL WorkspaceIdentity WITH IDENTITY = 'Managed Identity' |
|  | GO |
|  |  |
|  | -- SQL logins only: |
|  | -- create server-scoped credential for the containers in demo storage account |
|  | -- SQL logins will use this credential in OPENROWSET function without data source that uses absolute file URL |
|  | CREATE CREDENTIAL [https://sqlondemandstorage.blob.core.windows.net] |
|  | WITH IDENTITY='SHARED ACCESS SIGNATURE', |
|  | SECRET = 'sv=2018-03-28&ss=bf&srt=sco&sp=rl&st=2019-10-14T12%3A10%3A25Z&se=2061-12-31T12%3A10%3A00Z&sig=KlSU2ullCscyTS0An0nozEpo4tO5JAgGBvw%2FJX2lguw%3D' |
|  | GO |
|  |  |
|  | CREATE CREDENTIAL MyCosmosDbAccountCredential |
|  | WITH IDENTITY = 'SHARED ACCESS SIGNATURE', SECRET = 's5zarR2pT0JWH9k8roipnWxUYBegOuFGjJpSjGlR36y86cW0GQ6RaaG8kGjsRAQoWMw1QKTkkX8HQtFpJjC8Hg=='; |
|  | GO |
|  |  |
|  | CREATE SCHEMA parquet; |
|  | GO |
|  | CREATE SCHEMA csv; |
|  | GO |
|  | CREATE SCHEMA json; |
|  | GO |
|  | CREATE SCHEMA cosmosdb; |
|  | GO |
|  | CREATE SCHEMA delta; |
|  | GO |
|  |  |
|  | -- Create external data source secured using credential |
|  | CREATE EXTERNAL DATA SOURCE SqlOnDemandDemo WITH ( |
|  | LOCATION = 'https://sqlondemandstorage.blob.core.windows.net', |
|  | CREDENTIAL = sqlondemand |
|  | ); |
|  | GO |
|  |  |
|  | -- Data source referencing Delta Lake folders |
|  | CREATE EXTERNAL DATA SOURCE DeltaLakeStorage WITH ( |
|  | LOCATION = 'https://sqlondemandstorage.blob.core.windows.net/delta-lake', |
|  | CREDENTIAL = sqlondemand |
|  | ); |
|  | GO |
|  |  |
|  | -- Create publicly available external data sources |
|  | CREATE EXTERNAL DATA SOURCE AzureOpenData |
|  | WITH ( LOCATION = 'https://azureopendatastorage.blob.core.windows.net/') |
|  | GO |
|  | CREATE EXTERNAL DATA SOURCE YellowTaxi |
|  | WITH ( LOCATION = 'https://azureopendatastorage.blob.core.windows.net/nyctlc/yellow/') |
|  | GO |
|  | CREATE EXTERNAL DATA SOURCE GreenTaxi |
|  | WITH ( LOCATION = 'https://azureopendatastorage.blob.core.windows.net/nyctlc/green/') |
|  | GO |
|  |  |
|  | CREATE EXTERNAL FILE FORMAT QuotedCsvWithHeader |
|  | WITH ( |
|  | FORMAT\_TYPE = DELIMITEDTEXT, |
|  | FORMAT\_OPTIONS ( |
|  | FIELD\_TERMINATOR = ',', |
|  | STRING\_DELIMITER = '"', |
|  | FIRST\_ROW = 2 |
|  | ) |
|  | ); |
|  | GO |
|  | CREATE EXTERNAL FILE FORMAT QuotedCsvWithoutHeader |
|  | WITH ( |
|  | FORMAT\_TYPE = DELIMITEDTEXT, |
|  | FORMAT\_OPTIONS ( |
|  | FIELD\_TERMINATOR = ',', |
|  | STRING\_DELIMITER = '"', |
|  | FIRST\_ROW = 1 |
|  | ) |
|  | ); |
|  | GO |
|  | CREATE EXTERNAL FILE FORMAT NativeParquet |
|  | WITH ( |
|  | FORMAT\_TYPE = PARQUET |
|  | ); |
|  | GO |
|  | CREATE EXTERNAL FILE FORMAT DeltaLakeFormat |
|  | WITH ( |
|  | FORMAT\_TYPE = DELTA |
|  | ); |
|  | GO |
|  |  |
|  | CREATE OR ALTER VIEW cosmosdb.Ecdc |
|  | AS SELECT \* |
|  | FROM OPENROWSET( |
|  | PROVIDER = 'CosmosDB', |
|  | CONNECTION = 'Account=synapselink-cosmosdb-sqlsample;Database=covid', |
|  | OBJECT = 'Ecdc', |
|  | SERVER\_CREDENTIAL = 'MyCosmosDbAccountCredential' |
|  | ) with ( date\_rep varchar(20), cases bigint, geo\_id varchar(6) ) as rows |
|  | GO |
|  |  |
|  |  |
|  | CREATE EXTERNAL TABLE csv.population |
|  | ( |
|  | [country\_code] VARCHAR (5) COLLATE Latin1\_General\_BIN2, |
|  | [country\_name] VARCHAR (100) COLLATE Latin1\_General\_BIN2, |
|  | [year] smallint, |
|  | [population] bigint |
|  | ) |
|  | WITH ( |
|  | LOCATION = 'csv/population/population.csv', |
|  | DATA\_SOURCE = SqlOnDemandDemo, |
|  | FILE\_FORMAT = QuotedCsvWithHeader |
|  | ); |
|  | GO |
|  |  |
|  | CREATE VIEW parquet.YellowTaxi |
|  | AS SELECT \*, nyc.filepath(1) AS [year], nyc.filepath(2) AS [month] |
|  | FROM |
|  | OPENROWSET( |
|  | BULK 'parquet/taxi/year=\*/month=\*/\*.parquet', |
|  | DATA\_SOURCE = 'SqlOnDemandDemo', |
|  | FORMAT='PARQUET' |
|  | ) AS nyc |
|  | GO |
|  |  |
|  | CREATE VIEW csv.YellowTaxi |
|  | AS |
|  | SELECT \*, nyc.filepath(1) AS [year], nyc.filepath(2) AS [month] |
|  | FROM OPENROWSET( |
|  | BULK 'csv/taxi/yellow\_tripdata\_\*-\*.csv', |
|  | DATA\_SOURCE = 'SqlOnDemandDemo', |
|  | FORMAT = 'CSV', |
|  | FIRSTROW = 2 |
|  | ) |
|  | WITH ( |
|  | vendor\_id VARCHAR(100) COLLATE Latin1\_General\_BIN2, |
|  | pickup\_datetime DATETIME2, |
|  | dropoff\_datetime DATETIME2, |
|  | passenger\_count INT, |
|  | trip\_distance FLOAT, |
|  | rate\_code INT, |
|  | store\_and\_fwd\_flag VARCHAR(100) COLLATE Latin1\_General\_BIN2, |
|  | pickup\_location\_id INT, |
|  | dropoff\_location\_id INT, |
|  | payment\_type INT, |
|  | fare\_amount FLOAT, |
|  | extra FLOAT, |
|  | mta\_tax FLOAT, |
|  | tip\_amount FLOAT, |
|  | tolls\_amount FLOAT, |
|  | improvement\_surcharge FLOAT, |
|  | total\_amount FLOAT |
|  | ) AS nyc |
|  | GO |
|  |  |
|  | CREATE VIEW json.Books |
|  | AS SELECT \* |
|  | FROM |
|  | OPENROWSET( |
|  | BULK 'json/books/\*.json', |
|  | DATA\_SOURCE = 'SqlOnDemandDemo', |
|  | FORMAT='CSV', |
|  | FIELDTERMINATOR ='0x0b', |
|  | FIELDQUOTE = '0x0b', |
|  | ROWTERMINATOR = '0x0b', |
|  | ROWSET\_OPTIONS = '{"READ\_OPTIONS":["ALLOW\_INCONSISTENT\_READS"]}' |
|  | ) |
|  | WITH ( |
|  | content varchar(8000) |
|  | ) AS books; |
|  | GO |
|  |  |
|  | CREATE EXTERNAL TABLE delta.Covid ( |
|  | date\_rep date, |
|  | cases int, |
|  | geo\_id varchar(6) |
|  | ) WITH ( |
|  | LOCATION = 'covid', --> the root folder containing the Delta Lake files |
|  | data\_source = DeltaLakeStorage, |
|  | FILE\_FORMAT = DeltaLakeFormat |
|  | ); |