Serverless SQL pool in Azure Synapse Analytics

Every Azure Synapse Analytics workspace comes with serverless SQL pool endpoints that you can use to query data in the Azure Data Lake (Parquet, Delta Lake, delimited text formats), Azure Cosmos DB, or Dataverse.

Query CSV files

```
select top 10 *

from openrowset(

   bulk 'https://pandemicdatalake.blob.core.windows.net/public/curated/covid-19/ecdc_cases/latest/ecdc_cases.csv',

   format = 'csv',

 parser_version = '2.0',

   firstrow = 2 ) as rows
```

There are some additional options that can be used to adjust parsing rules to custom CSv format:

- ESCAPE_CHAR = 'char' Specifies the character in the file that is used to escape itself and all delimiter values in the file. If the escape character is followed by a value other than itself, or any of the delimiter values, the escape character is dropped when reading the value. The ESCAPE_CHAR parameter will be applied whether the FIELDQUOTE is or isn't enabled. It won't be used to escape the quoting character. The quoting character must be escaped with another quoting character. Quoting character can appear within column value only if value is encapsulated with quoting characters.
- FIELDTERMINATOR ='field_terminator' Specifies the field terminator to be used. The default field terminator is a comma (",")
- ROWTERMINATOR ='row_terminator' Specifies the row terminator to be used. The default row terminator is a newline character: **\r\n**.

Query Delimted text files

```
select  *

from openrowset(
```

```
        bulk 'csv/population-unix-hdr-quoted/population.csv',
        format = 'csv', parser_version = '2.0',
        fieldterminator =',',
        rowterminator = '0x0a',
        firstrow = 2,
        fieldquote = '"'
    ) as [r]
```

Query Parquet files

```
Select  top  10 *
from openrowset(
    bulk    'https://pandemicdatalake.blob.core.windows.net/public/curated/covid-19/ecdc_cases/latest/ecdc_cases.parquet',
    format = 'parquet')  as  rows
```

Query JSON files

```
select top 10 *
from openrowset(
        bulk
'https://pandemicdatalake.blob.core.windows.net/public/curated/covid-19/ecdc_cases/latest/ecdc_cases.jsonl',
        format = 'csv',
        fieldterminator ='0x0b',
        fieldquote = '0x0b'
    ) with (doc nvarchar(max)) as rows
Go
```

Query Delta Lake files

```
select top 10 *
from openrowset(
    bulk 'https://sqlondemandstorage.blob.core.windows.net/delta-lake/covid/',
```

format = 'delta') as rows;

Create and Use external tables with Synapse SQL

You can create external tables in Synapse SQL pools via the following steps:

1. CREATE EXTERNAL DATA SOURCE to reference an external Azure storage and specify the credential that should be used to access the storage.
2. CREATE EXTERNAL FILE FORMAT to describe format of CSV or Parquet files.
3. CREATE EXTERNAL TABLE on top of the files placed on the data source with the same file format.

Creating a database scoped credential for a shared access signature

Create a db

Syntaxsql

CREATE DATABASE SYNAPSE1

Syntaxsql

USE SYNAPSE1

GO

Create a db master key if one does not already exist, using your own password.

Syntaxsql

```
CREATE MASTER KEY ENCRYPTION BY
PASSWORD='<EnterStrongPasswordHere>';

-- Create a database scoped credential.
CREATE DATABASE SCOPED CREDENTIAL MyCredentials
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET =
'QLYMgmSXMklt%2FI1U6DcVrQixnlU5Sgbtk1qDRakUBGs%3D';
```

CREATE EXTERNAL DATA SOURCE

External data sources are used to connect to storage accounts.

Syntax for CREATE EXTERNAL DATA SOURCE

- Hadoop
- Native

External data sources without TYPE=HADOOP are generally available in serverless SQL pools and in public preview in dedicated pools.

Syntaxsql

```
CREATE EXTERNAL DATA SOURCE <data_source_name>
WITH
(    LOCATION         = '<prefix>://<path>'
     [, CREDENTIAL = <database scoped credential> ]
)
[;]
```

Arguments for CREATE EXTERNAL DATA SOURCE

data_source_name

Specifies the user-defined name for the data source. The name must be unique within the database.

Location

LOCATION = '<prefix>://<path>' - Provides the connectivity protocol and path to the external data source. The following patterns can be used in location:

https: prefix enables you to use subfolder in the path.

Credential

CREDENTIAL = <database scoped credential> is optional credential that will be used to authenticate on Azure storage. External data source without credential can access public storage account or use the caller's Azure AD identity to access files on storage.

- In dedicated SQL pool, database scoped credential can specify custom application identity, workspace Managed Identity, or SAK key.
- In serverless SQL pool, database scoped credential can specify workspace Managed Identity, or SAS key.

TYPE

TYPE = HADOOP is the option that specifies that Java-based technology should be used to access underlying files. This parameter can't be used in serverless SQL pool that uses built-in native reader.

Example for CREATE EXTERNAL DATA SOURCE

- Hadoop
- Native

The following example creates an external data source in serverless or dedicated SQL pool for Azure Data Lake Gen2 that can be accessed using SAS credential:
SQL

```
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 EXTERNAL DATA SOURCE SqlOnDemandDemo WITH (
   LOCATION = 'https://sqlondemandstorage.blob.core.windows.net',
   CREDENTIAL = sqlondemand
);
```

CREATE EXTERNAL FILE FORMAT

Creates an external file format object that defines external data stored in Azure Blob Storage or Azure Data Lake Storage. Creating an external file format is a prerequisite for creating an external table. The complete documentation is here.

By creating an external file format, you specify the actual layout of the data referenced by an external table.

Syntax for CREATE EXTERNAL FILE FORMAT

Syntaxsql

```
-- Create an external file format for PARQUET files.
CREATE EXTERNAL FILE FORMAT file_format_name
WITH (
   FORMAT_TYPE = PARQUET
```

```
   [ , DATA_COMPRESSION = {
      'org.apache.hadoop.io.compress.SnappyCodec'
     | 'org.apache.hadoop.io.compress.GzipCodec'      }
   ]);

--Create an external file format for DELIMITED TEXT files
CREATE EXTERNAL FILE FORMAT file_format_name
WITH (
   FORMAT_TYPE = DELIMITEDTEXT
   [ , DATA_COMPRESSION = 'org.apache.hadoop.io.compress.GzipCodec' ]
   [ , FORMAT_OPTIONS ( <format_options> [ ,...n  ] ) ]
   );
```

Example for CREATE EXTERNAL FILE FORMAT

The following example creates an external file format for census files:

SQL

```
CREATE EXTERNAL FILE FORMAT census_file_format
WITH
(
   FORMAT_TYPE = PARQUET,
   DATA_COMPRESSION = 'org.apache.hadoop.io.compress.SnappyCodec'
)
```

CREATE EXTERNAL TABLE

The CREATE EXTERNAL TABLE command creates an external table for Synapse SQL to access data stored in Azure Blob Storage or Azure Data Lake Storage.

Syntax for CREATE EXTERNAL TABLE

Syntaxsql

```
CREATE EXTERNAL TABLE { database_name.schema_name.table_name |
schema_name.table_name | table_name }
   ( <column_definition> [ ,...n ] )
   WITH (
      LOCATION = 'folder_or_filepath',
      DATA_SOURCE = external_data_source_name,
      FILE_FORMAT = external_file_format_name
```

```
    [,                    TABLE_OPTIONS                    =
N'{"READ_OPTIONS":["ALLOW_INCONSISTENT_READS"]}' ]
    [, <reject_options> [ ,...n ] ]
  )
[;]

<column_definition> ::=
column_name <data_type>
  [ COLLATE collation_name ]

<reject_options> ::=
{
  | REJECT_TYPE = value,
  | REJECT_VALUE = reject_value,
  | REJECT_SAMPLE_VALUE = reject_sample_value,
  | REJECTED_ROW_LOCATION = '/REJECT_Directory'
}
```

Example CREATE EXTERNAL TABLE

The following example creates an external table. It returns the first row:

```
SQL
CREATE EXTERNAL TABLE census_external_table
(
  decennialTime varchar(20),
  stateName varchar(100),
  countyName varchar(100),
  population int,
  race varchar(50),
  sex   varchar(10),
  minAge int,
  maxAge int
)
WITH (
  LOCATION = '/parquet/',
  DATA_SOURCE = population_ds,
  FILE_FORMAT = census_file_format
)
GO

SELECT TOP 1 * FROM census_external_table
```

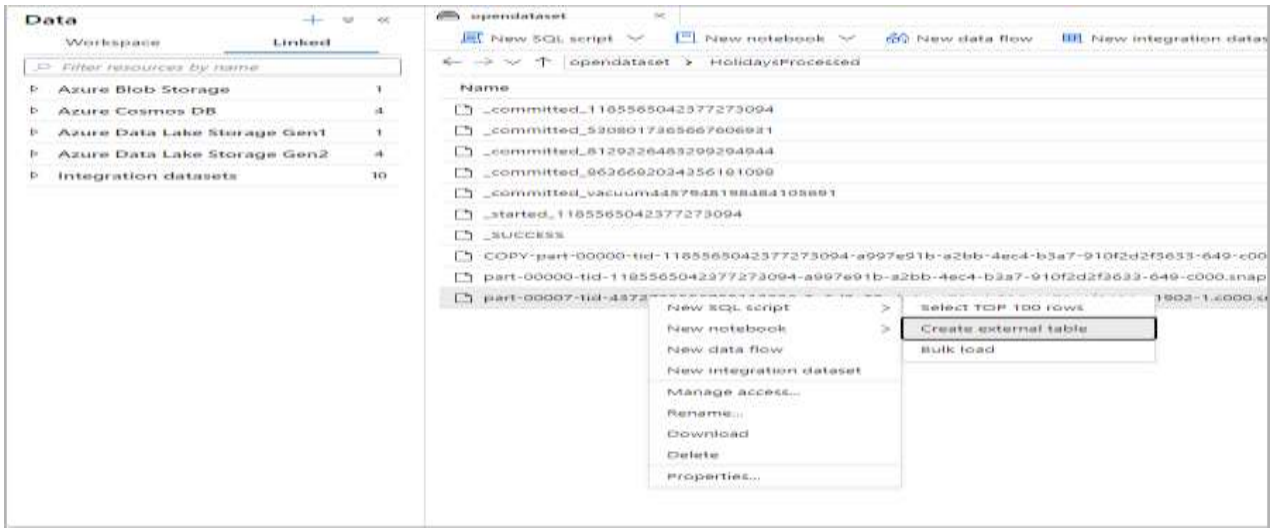
Create and query external tables from a file in Azuredl

Using Data Lake exploration capabilities of Synapse Studio you can now create and query an external table using Synapse SQL pool with a simple right-click on the file. The one-click gesture to create external tables from the ADLS Gen2 storage account is only supported for Parquet files.

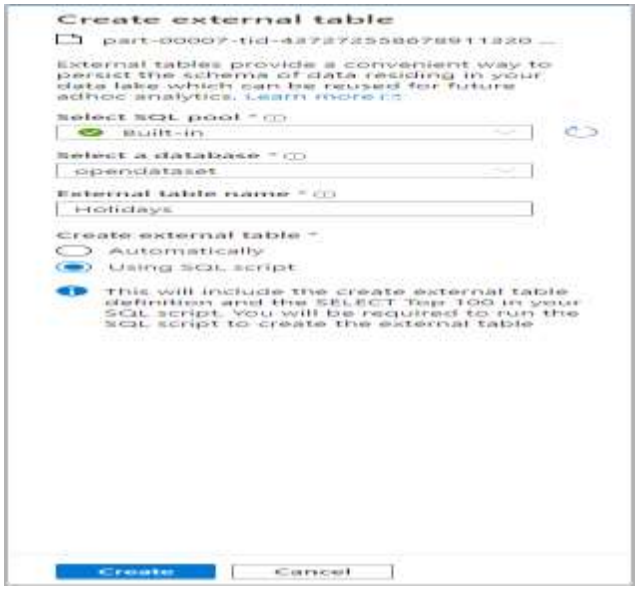
Prerequisites

- You must have access to the workspace with at least the Storage Blob Data Contributor access role to the ADLS Gen2 Account or Access Control Lists (ACL) that enable you to query the files.
- You must have at least permissions to create and query external tables on the Synapse SQL pool (dedicated or serverless).

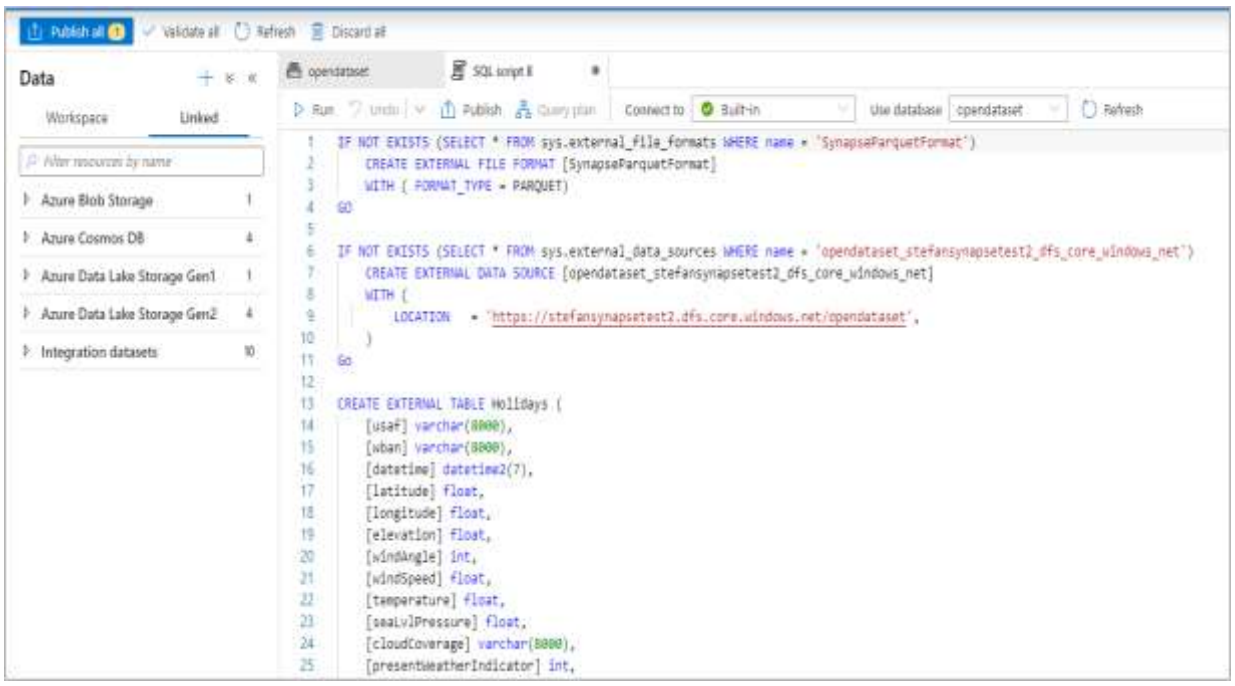
From the Data panel, select the file that you would like to create the external table from:



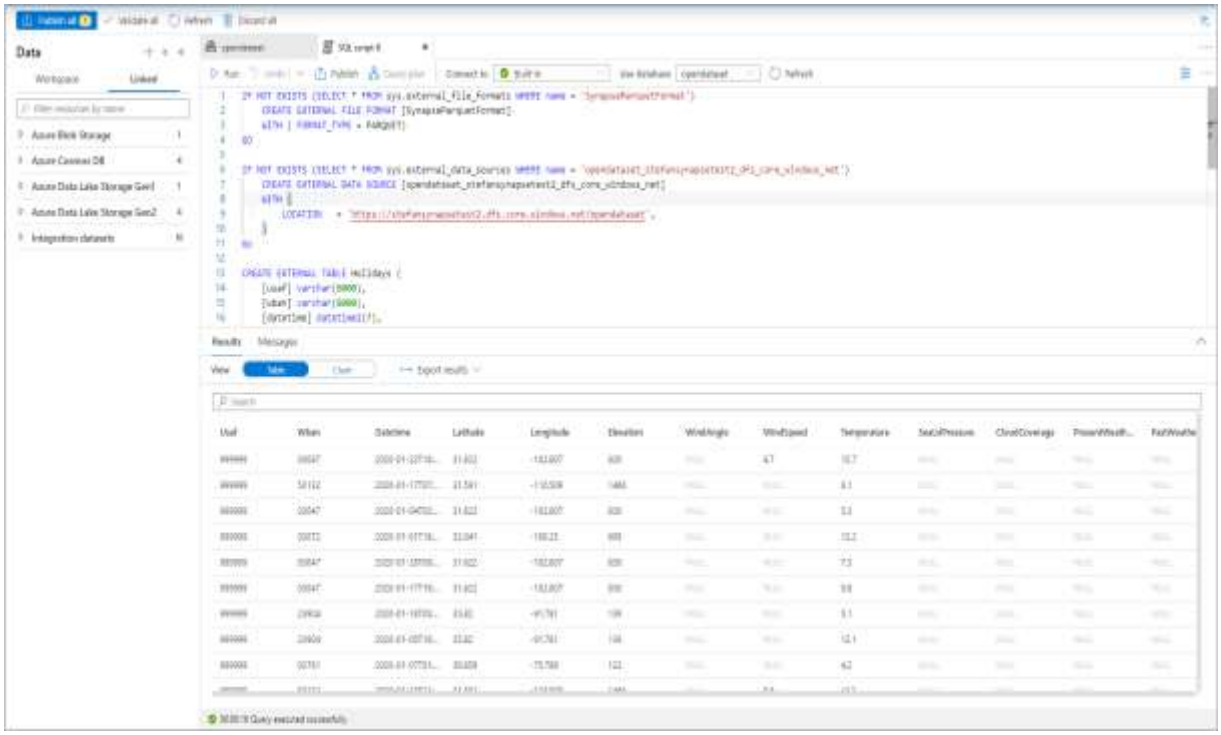
A dialog window will open. Select dedicated SQL pool or serverless SQL pool, give a name to the table and select open script:



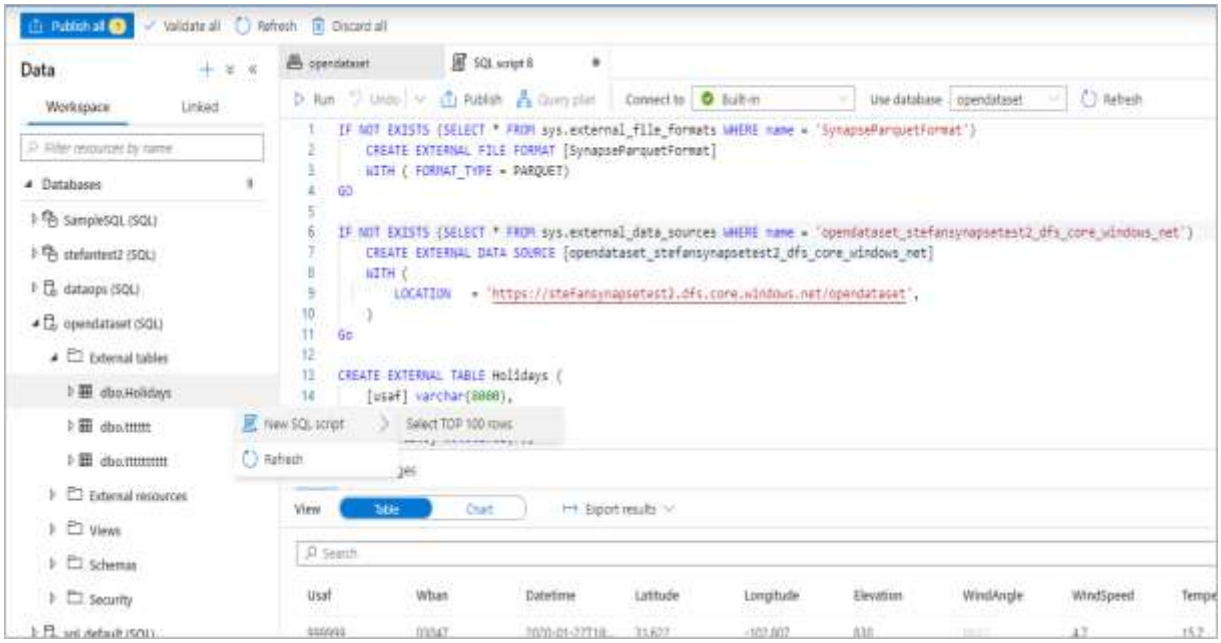
The SQL Script is autogenerated inferring the schema from the file:



Run the script. The script will automatically run a Select Top 100 *.:



The external table is now created, for future exploration of the content of this external table the user can query it directly from the Data pane:



Next steps

CETAS is for how to save query results to an external table in Azure Storage. Or you can start querying Apache Spark for Azure Synapse external tables.