For more details on learning how to mount your ADLSgen2 account to Synapse workspace, please see Microsoft’s article on the topic here: <https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/synapse-file-mount-api>

The endpoints would be like the following and can be specified within a SQL Server Database data source connection in Power BI for either Import or Direct Query connectivity modes. When working with the preferred Parquet file format, creating pre-aggregated views in Serverless SQL Pools will lead to better performance and lower costs. For analytics workloads that require many queries across large datasets, it may be beneficial to use Dedicated SQL pools as the storage layer since you would pay a fixed cost which may result in being lower than paying per query for this scenario.

Dedicated SQL endpoint

synapse-rl-001.sql.azuresynapse.net

Serverless SQL endpoint

synapse-rl-001-ondemand.sql.azuresynapse.net

Development endpoint

https://synapse-rl-001.dev.azuresynapse.net

For more details on how to granularly set up access control in your Synapse Workspace, please refer to the following Microsoft article: https://docs.microsoft.com/en-us/azure/synapse-analytics/security/how-to-set-up-access-control

When creating a Delta table in a Spark database, simply run SQL syntax like the code shown below which includes defining the code language using the %%sql magic command, running the create table statement which includes defining the data’s schema, and finally specifying the use of Delta format along with the ADLSgen2 location folder. Complex merge statements with custom logic can be written to load data into the tables and the table can be queried using common SQL syntax. If you are running this code, please ensure that your ADLSgen2 account is linked.

%%sql

-- Create Delta Lake table, define schema and location

CREATE TABLE Delta\_Customer (

Customer STRING NOT NULL,

Customer\_ID INT NOT NULL,

BeginDate DATE NOT NULL,

EndDate DATE NOT NULL,

CurrentRecord INT NOT NULL

)

USING DELTA

-- specify data lake folder location

LOCATION '/Delta/Customer/'

The concept of time travel is also a capability to read previous versions of data by running code like the following:

# Load a previous version of the Delta\_Customer table into a dataframe

df = spark.read.format("delta").option("versionAsOf", 3).load("/Delta/Customer/")

df.show()

The query can be as simple as the one shown in the code below, or quite complex to include multiple joins, filters, orderings, and groupings. For more information about querying Delta Lake from Azure Synapse Analytics Workspace, please see the following Microsoft article: [https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/query-delta-lake-format](about:blank) Be sure to stay informed about the limitations of this capability.

SELECT \*

FROM OPENROWSET(

BULK 'https://adlsaccount.blob.core.windows.net/data/Delta/Customer/', --Specify Delta Lake folder

FORMAT = 'delta') as rows --Specify delta format

ORDER BY Customer

Code for creating table with Constraints in Synapse Dedicated SQL Pools

CREATE TABLE Customer(Id int PRIMARY KEY NONCLUSTERED NOT ENFORCED, value int)

More information about getting started with Synapse ML, including code samples can be found in the following site: <https://microsoft.github.io/SynapseML/>