Asessment-24 Vinutha S

02/JAN/2024

Creating Delta Lake Tables in Azure Databricks

how to create a delta table format in Azure Databricks.

Introduction

In modern data engineering, various file formats are used to host data like CSV, TSV, parquet, json, avro and many others. With the proliferation of data lakes in the industry, data formats like delta and hudi also have become very popular. Data platforms like Databricks have positioned delta format at the center of their data lake paradigm and is popularly known as Delta Lake. Azure Databricks supports creating tables in a variety of formats mentioned above including delta. While Databricks supports many platforms, to consume the tables created on this platform with external Azure services, many of them require the table format to be of delta format.

What is a delta lake table in Azure Databricks?

Delta lake is an open-source data format that provides ACID transactions, data reliability, query performance, data caching and indexing, and many other benefits. Delta lake can be thought of as an extension of existing data lakes and can be configured per the data requirements. Azure Databricks has a delta engine as one of the core components that facilitates delta lake format for data engineering and performance. Delta lake format is used to create modern data lake or lakehouse architectures. It is also used to build a combined streaming and batch architecture popularly known as lambda architecture.

One of the unique advantages of data formats like delta lake or apache hudi is time travel. As data keeps changing, one may want to preserve the history of data. Typically to preserve history, methods like Slowly Changing Dimension or creating pools of archive table are being used. With large volumes of data, manually managing data versioning can become a challenge over time. Also hosting historical data as well as active data in the same repository can have an impact on the query performance. At times, when there is a need to revert to an older version of data for different business or technical reasons, it would need a significant backup and restore operation. To add to the complexity, if one needs to view the older version of data or the chronological change in data over time, it is hard to facilitate the solution without a use-case specific setup with required additional times and resources. This feature is one of the many spectacular features in storage formats like delta lake where one can seamlessly execute time-travel queries that are facilitated by advanced data versioning provided by delta engine on delta lake format.

Creating Delta Lake tables

In this exercise, we would be using a sample CSV file as the source to create a delta format table in Azure Databricks. For this purpose, we first need an Azure account with the required privileges to create an Azure Databricks workspace. It is assumed that an Azure account is already in place.

Open the Azure portal, navigate to the Azure Databricks service dashboard, and click on the Create button to create a new instance. Provide the required details like subscription, resource group, pricing tier, workspace name and the region in which the instance will be created. Using the standard tier, we can proceed and create a new instance.

After having the workspace in place, we need to create a new table in Azure Databricks using an existing CSV file. We can use any CSV file with a few fields and at least one record in it. Click on the Create menu option and select the Table option, which would open the interface as shown below. Click on the **Upload File** option and upload the sample file

Later we discussed a sample demo project for clear idea.