Assignment – Day 17

-Sarthak Niranjan Kulkarni (Maverick)

- [sarthakkul2311@gmail.com](mailto:sarthakkul2311@gmail.com) - (+91) 93256 02791

**28/11/2024 (Thursday)**

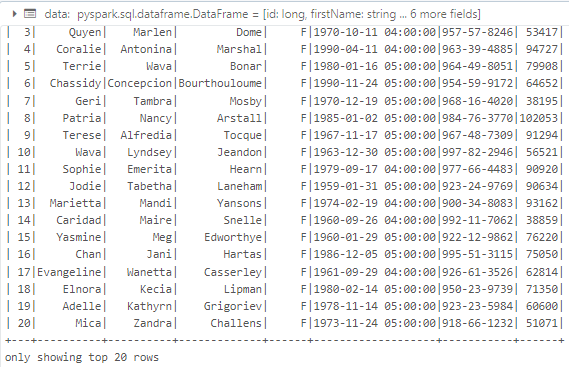
**Practice on Delta Tables: -**

1. **Loading and Displaying Data from Delta Table in Azure Databricks**

🡪 spark.table("default.export")

data = spark.read.format("delta").load("dbfs:/user/hive/warehouse/export")

data.show()



1. **Writing DataFrame to Delta Tables in Azure Databricks**

🡪 from pyspark.sql import SparkSession

spark = SparkSession.builder.appName('Delta Table Write').getOrCreate()

data = spark.createDataFrame([

(1, "Alice", 1000),

(2, "Bob", 2000),

(3, "Charlie", 3000)

], ["id", "name", "salary"])

# Write the DataFrame as Delta tables

data.write.format('delta').saveAsTable("mydata\_delta", mode="overwrite")

data.write.format('delta').saveAsTable("mydata")

1. **Loading and Displaying Data from Delta Table in Databricks**

🡪 spark.table("default.export")

data = spark.read.format("delta").load("dbfs:/user/hive/warehouse/export")

data.display()

A screenshot of a computer

Description automatically generated

A graph with colored lines

Description automatically generated

**Summary on Delta Tables: -**

In Azure Databricks, Delta tables are used to store data in a structured format that supports efficient querying and data management. The first block of code shows how to read data from an existing Delta table stored in the Databricks metastore or from a path in the DBFS (Databricks File System). Using the spark.read.format("delta") method, we load the data from the Delta table into a DataFrame and display it. This process allows us to view the content of the Delta table, which is stored in a structured format for analysis.

Next, we see how to create and write a new Delta table from a DataFrame. The data.write.format("delta").saveAsTable("mydata") command writes the data into a new Delta table named "mydata". This code demonstrates the ability to save a DataFrame into a Delta table, making it accessible for future queries and operations. We can specify the mode (like overwrite) to control how existing data is handled when writing the new data.

Finally, the data.display() method is used to show the contents of the newly written Delta table in a Databricks notebook. Overall, Delta tables provide a powerful and efficient way to store and manage data in Databricks, with built-in support for ACID transactions, versioning, and schema enforcement. This makes them ideal for data analysis and machine learning tasks where data integrity and fast querying are crucial.