from pyspark.sql import SparkSession

spark=SparkSession.builder.appName("sep13").getOrCreate()

dbutils.fs.cp("File:/Workspace/Shared/sales\_data.csv","dbfs:/FileStore/sales\_data.csv")

dbutils.fs.cp("File:/Workspace/Shared/customer\_data.json","dbfs:/FileStore/customer\_data.json")

dbutils.fs.cp("File:/Workspace/Shared/new\_sales\_data.csv","dbfs:/FileStore/new\_sales\_data.csv")

df\_sales\_data=spark.read.format("csv").option("header","true").load("dbfs:/FileStore/sales\_data.csv")

df\_sales\_data.write.format("delta").mode("overwrite").save("dbfs:/FileStore/delta/delta\_sales\_data")

df\_customer\_data=spark.read.format("json").option("multiline","true").load("dbfs:/FileStore/customer\_data.json")

df\_customer\_data.write.format("delta").mode("overwrite").save("dbfs:/FileStore/delta/delta\_customer\_data")

df\_new\_sales\_data=spark.read.format("csv").option("header","true").load("dbfs:/FileStore/new\_sales\_data.csv")

df\_new\_sales\_data.write.format("delta").mode("overwrite").save("dbfs:/FileStore/delta/delta\_new\_sales\_data")

df\_sales\_data=spark.read.format("delta").load("dbfs:/FileStore/delta/delta\_sales\_data")

df\_new\_sales\_data=spark.read.format("delta").load("dbfs:/FileStore/delta/delta\_new\_sales\_data")

df\_sales\_data.createOrReplaceTempView("delta\_sales\_data")

df\_new\_sales\_data.createOrReplaceTempView("delta\_new\_sales\_data")

spark.sql

("""

MERGE INTO delta\_sales\_data AS target

USING delta\_new\_sales\_data AS source

ON target.OrderID = source.OrderID

WHEN MATCHED THEN UPDATE SET

target.Quantity = source.Quantity

WHEN NOT MATCHED THEN INSERT

(

OrderID, OrderDate, CustomerID, Product, Quantity, Price

)

VALUES (

source.OrderID, source.OrderDate, source.CustomerID, source.Product, source.Quantity, source.Price

)

""")

spark.sql("select \* from delta\_sales\_data") .show()

spark.sql("describe delta\_sales\_data").show()

spark.sql("create table if not exists delta\_sales\_table as select \* from delta\_sales\_data")

spark.sql("select \* from delta\_sales\_table").show()

spark.sql("describe delta\_sales\_table").show()

spark.sql("OPTIMIZE delta\_sales\_table ZORDER BY (OrderID)")

spark.sql("Describe History delta\_sales\_table").show(truncate=False)

spark.sql("vacuum delta\_sales\_table retain 170 hours")

spark.sql("RESTORE TABLE delta\_sales\_table TO VERSION AS OF 2;")