DataBricks Delta Lake Assignment – 13th September

SaiPrabath Chowdary S

Task 1 :

01:31 PM (11s) 18: csv ->delta Python

```python
# task 1
sales_df = spark.read.option("header", "true").csv("file:/Workspace/Shared/sales_data2.csv")
sales_df.write.format("delta").mode("overwrite").save("/delta/sales_data2")
```

(5) Spark Jobs

sales_df: pyspark.sql.dataframe.DataFrame = [OrderID: string, OrderDate: string ... 4 more fields]

01:31 PM (4s) 19: json -> delta

```python
customer_df = spark.read.option("multiline", "true").json("file:/Workspace/Shared/customer_data.json")
customer_df.write.format("delta").mode("overwrite").save("/mnt/delta/customer_data")
```

(5) Spark Jobs

customer_df: pyspark.sql.dataframe.DataFrame = [CustomerID: string, CustomerName: string ... 2 more fields]

Task 2:

01:35 PM (4s) 20: merge

```python
# task 2

new_sales_df = spark.read.option("header", "true").csv("file:/Workspace/Shared/new_sales_data.csv")
new_sales_df.write.format("delta").mode("overwrite").save("/delta/new_sales_data")
```

(5) Spark Jobs

new_sales_df: pyspark.sql.dataframe.DataFrame = [OrderID: string, OrderDate: string ... 4 more fields]

01:39 PM (1s) 21 Python

```python
# create delta tables
spark.sql("CREATE TABLE IF NOT EXISTS sales_delta_table USING DELTA LOCATION '/delta/sales_data2'")
spark.sql("CREATE TABLE IF NOT EXISTS new_sales_delta_table USING DELTA LOCATION '/delta/new_sales_data'")
```

```
DataFrame[]
```

01:40 PM (4s) 22

```python
# merge
spark.sql('''
MERGE INTO sales_delta_table AS TARGET
USING new_sales_delta_table AS SOURCE
ON TARGET.OrderID = SOURCE.OrderID
WHEN MATCHED THEN UPDATE SET TARGET.Quantity=SOURCE.Quantity, TARGET.Price=SOURCE.Price
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 sales_delta_table").show()
```

(16) Spark Jobs

```
+-------+----------+----------+--------+--------+-----+
|OrderID| OrderDate|CustomerID| Product|Quantity|Price|
+-------+----------+----------+--------+--------+-----+
|   1001|2024-01-15|      C001|Widget A|      10|25.50|
|   1003|2024-01-16|      C001|Widget C|       8|22.50|
|   1004|2024-01-17|      C003|Widget A|      15|25.50|
|   1005|2024-01-18|      C004|Widget D|       7|30.00|
|   1006|2024-01-19|      C002|Widget B|       9|15.75|
|   1007|2024-01-20|      C005|Widget C|      12|22.50|
|   1008|2024-01-21|      C003|Widget A|      10|25.50|
|   1009|2024-01-22|      C006|Widget E|      14|20.00|
|   1010|2024-01-23|      C007|Widget F|       6|35.00|
|   1002|2024-01-16|      C002|Widget B|      10|15.75|
+-------+----------+----------+--------+--------+-----+
```

Task 3

01:44 PM (1s) 23: optimize Python

```python
# task 3
spark.sql("OPTIMIZE sales_delta_table ZORDER BY (OrderDate)")
```

(4) Spark Jobs

```
size:bigint>,inputOtherFiles:struct<numFiles:bigint,size:bigint>,inputNumZCubes:bigint,mergedFiles:struct<numFiles:bi
int,size:bigint>,numOutputZCubes:bigint>,numBins:bigint,numBatches:bigint,totalConsideredFiles:bigint,totalFilesSkippe
d:bigint,preserveInsertionOrder:boolean,numFilesSkippedToReduceWriteAmplification:bigint,numBytesSkippedToReduceWriteA
mplification:bigint,startTimeMs:bigint,endTimeMs:bigint,totalClusterParallelism:bigint,totalScheduledTasks:bigint,auto
CompactParallelismStats:struct<maxClusterActiveParallelism:bigint,minClusterActiveParallelism:bigint,maxSessionActiveP
```

Task 4:

01:52 PM (<1s) 24: descibe history

```
# task 4
spark.sql("DESCRIBE HISTORY sales_delta_table")
```

```
DataFrame[version: bigint, timestamp: timestamp, userId: string, userName: string, operation: string, operationParameters: map<string,string>, job: struct<jobId:string,jobName:string,jobRunId:string,runId:string,jobOwnerId:string,triggerType:string>, notebook: struct<notebookId:string>, clusterId: string, readVersion: bigint, isolationLevel: string, isBlindAppend: boolean, operationMetrics: map<string,string>, userMetadata: string, engineInfo: string]
```

01:54 PM (32s) 25: vaccum

```
spark.sql("VACUUM sales_delta_table RETAIN 168 HOURS")
```

(22) Spark Jobs

```
DataFrame[path: string]
```

Task 5:

02:02 PM (<1s) 26 Python

```
# task 5
version_number = 1
sales_version_df = spark.read.format("delta").option("versionAsOf", version_number).load("/delta/sales_data2")
sales_version_df.show()
```

(2) Spark Jobs

sales_version_df: pyspark.sql.dataframe.DataFrame = [OrderID: string, OrderDate: string ... 4 more fields]

```
+-------+----------+----------+--------+--------+-----+
|OrderID| OrderDate|CustomerID| Product|Quantity|Price|
+-------+----------+----------+--------+--------+-----+
|   1001|2024-01-15|      C001|Widget A|      10|25.50|
|   1003|2024-01-16|      C001|Widget C|       8|22.50|
|   1004|2024-01-17|      C003|Widget A|      15|25.50|
|   1005|2024-01-18|      C004|Widget D|       7|30.00|
|   1006|2024-01-19|      C002|Widget B|       9|15.75|
|   1007|2024-01-20|      C005|Widget C|      12|22.50|
|   1008|2024-01-21|      C003|Widget A|      10|25.50|
|   1002|2024-01-16|      C002|Widget B|      10|15.75|
|   1009|2024-01-22|      C006|Widget E|      14|20.00|
|   1010|2024-01-23|      C007|Widget F|       6|35.00|
+-------+----------+----------+--------+--------+-----+
```

02:03 PM (2s) 27

```
# Enforce schema while writing
schema = sales_df.schema
new_sales_df.write.format("delta").mode("append").option("mergeSchema", "true").save("/delta/sales_data_enforce")
```

(4) Spark Jobs

02:04 PM (25s) 28 Python

```python
# Perform vacuum to remove old data files
spark.sql("VACUUM delta.`/delta/sales_data_enforce` RETAIN 300 HOURS")
```

(21) Spark Jobs

```
DataFrame[path: string]
```

02:05 PM (<1s) 29

```python
df = spark.read.format("delta").load("/delta/sales_data_enforce")
display(df)
```

(1) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [OrderID: string, OrderDate: string ... 4 more fields]

Table

	OrderID	OrderDate	CustomerID	Product	Quantity	Price
1	1009	2024-01-22	C006	Widget E	14	20.00
2	1010	2024-01-23	C007	Widget F	6	35.00
3	1002	2024-01-16	C002	Widget B	10	15.75

3 rows | 0.28 seconds runtime Refreshed 10 minutes ago