Assignment: Delta Lake Concepts

SaiPrabath Chowdary S

12:54 PM (3s) 2: Load the given CSV and JSON

```
# copying files to dbfs
employee_csv_path = 'file:/Workspace/Shared/assignment17sep/employees.csv'
new_employee_csv_path = 'file:/Workspace/Shared/assignment17sep/NewEmployeeData.csv'
products_path = 'file:/Workspace/Shared/assignment17sep/products.json'

dbutils.fs.cp(employee_csv_path, 'dbfs:/FileStore/assignment17sep/employees.csv')
dbutils.fs.cp(new_employee_csv_path, 'dbfs:/FileStore/assignment17sep/NewEmployeeData.csv')
dbutils.fs.cp(products_path, 'dbfs:/FileStore/assignment17sep/products.json')
location = 'dbfs:/FileStore/assignment17sep/'

# loading csv and json
employees_df = spark.read.csv(f"{location}employees.csv", header=True, inferSchema=True)

from pyspark.sql.types import StructType, StructField, StringType, DoubleType

schema = StructType([
    StructField("ProductID", StringType(), True),
    StructField("ProductName", StringType(), True),
    StructField("Category", StringType(), True),
    StructField("Price", DoubleType(), True)
])

products_df = spark.read.schema(schema).json(f"{location}products.json")

employees_df.show()
products_df.show()
```

(4) Spark Jobs

employees_df: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, EmployeeName: string ... 3 more fields]

products_df: pyspark.sql.dataframe.DataFrame = [ProductID: string, ProductName: string ... 2 more fields]

```
+----------+------------+-----------+-----------+------+
|EmployeeID|EmployeeName| Department|JoiningDate|Salary|
+----------+------------+-----------+-----------+------+
|       101|        John|         HR| 2023-01-10| 50000|
|       102|       Alice|    Finance| 2023-02-15| 70000|
|       103|        Mark|Engineering| 2023-03-20| 85000|
|       104|        Emma|      Sales| 2023-04-01| 55000|
|       105|        Liam|  Marketing| 2023-05-12| 60000|
+----------+------------+-----------+-----------+------+

+---------+-----------+-----------+-----+
|ProductID|ProductName|   Category|Price|
+---------+-----------+-----------+-----+
|     P101|     Laptop|Electronics|1200.0|
|     P102|      Phone|Electronics| 800.0|
|     P103|     Tablet|Electronics| 600.0|
|     P104|    Monitor|Electronics| 300.0|
|     P105|      Mouse|Accessories|  25.0|
+---------+-----------+-----------+-----+
```

12:54 PM (11s) 3: converting into delta

```
employees_df.write.format("delta").save(f"{location}delta/employees")
products_df.write.format("delta").save(f"{location}delta/products")
```

(8) Spark Jobs

12:56 PM (11s)

4: Merge and Upsert (SCD)

```python
new_employee_df = spark.read.csv(f"{location}NewEmployeeData.csv", header=True, inferSchema=True)
new_employee_df.write.format("delta").mode("append").save(f"{location}delta/employees")
print("New data appended to Delta table successfully.")

# Create a temporary view for SQL operations
new_employee_df.createOrReplaceTempView("new_employee_data")

print("Merging new data into Delta table...")

delta_table_path = 'dbfs:/FileStore/assignment17sep/delta/employees'
spark.sql(f"""
MERGE INTO delta.`{delta_table_path}` AS target
USING new_employee_data AS source
ON target.EmployeeID = source.EmployeeID
WHEN MATCHED THEN UPDATE SET
    target.Salary = source.Salary
WHEN NOT MATCHED THEN INSERT (EmployeeID,EmployeeName,Department,JoiningDate,Salary)
    VALUES (source.EmployeeID,source.EmployeeName,source.Department,source.JoiningDate,source.Salary)
""")

print("Data merged successfully.")
```

(18) Spark Jobs

new_employee_df: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, EmployeeName: string ... 3 more fields]

```
New data appended to Delta table successfully.
Merging new data into Delta table...
Data merged successfully.
```

12:57 PM (35s)

5: Internals of Delta Table

```python
# History
print("Viewing Delta table history...")
history_df = spark.sql(f"DESCRIBE HISTORY delta.`{delta_table_path}`")
history_df.show(truncate=False)

# time travel
print("table before the previous merge operation.")
path = 'dbfs:/FileStore/assignment17sep/delta/employees'
df_time_travel = spark.read.format("delta").option("versionAsOf", 0).load(path)
df_time_travel.show(truncate=False)

# Vacuum
print("Vacuuming old files...")
spark.sql(f"VACUUM delta.`{delta_table_path}` RETAIN 168 HOURS")

print("Delta Table operations completed.")
```

(24) Spark Jobs

history_df: pyspark.sql.dataframe.DataFrame = [version: long, timestamp: timestamp ... 13 more fields]

df_time_travel: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, EmployeeName: string ... 3 more fields]

```
|NULL        |Databricks-Runtime/15.4.x-photon-scala2.12|
+-------+------------------+----------------+----------------------------------+---------+------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------+----+------------
----+-------------------+-----------+----------------+------------+----------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------+------------+---------------------------------------+

table before the previous merge operation.
+----------+------------+-----------+----------+------+
|EmployeeID|EmployeeName|Department |JoiningDate|Salary|
+----------+------------+-----------+----------+------+
|101       |John        |HR         |2023-01-10 |50000 |
|102       |Alice       |Finance    |2023-02-15 |70000 |
|103       |Mark        |Engineering|2023-03-20 |85000 |
|104       |Emma        |Sales      |2023-04-01 |55000 |
|105       |Liam        |Marketing  |2023-05-12 |60000 |
+----------+------------+-----------+----------+------+

Vacuuming old files...
```

✓ 12:59 PM (3s) 6: Optimize Delta Table

```
# delta table creation
spark.sql("CREATE TABLE IF NOT EXISTS delta_employee_table USING DELTA LOCATION 'dbfs:/FileStore/assignment17sep/delta/employees'")

# optimize
spark.sql("OPTIMIZE delta_employee_table")

# Zordering
spark.sql("OPTIMIZE delta_employee_table ZORDER BY Department")
```

▸ (7) Spark Jobs

```
DataFrame[path: string, metrics: struct<numFilesAdded:bigint,numFilesRemoved:bigint,filesAdded:struct<min:bigint,max:bigint,avg:double,totalFiles:bigint,totalSize:
bigint>,filesRemoved:struct<min:bigint,max:bigint,avg:double,totalFiles:bigint,totalSize:bigint>,partitionsOptimized:bigint,zOrderStats:struct<strategyName:string,
inputCubeFiles:struct<num:bigint,size:bigint>,inputOtherFiles:struct<num:bigint,size:bigint>,inputNumCubes:bigint,mergedFiles:struct<num:bigint,size:bigint>,numOut
putCubes:bigint,mergedNumCubes:bigint>,clusteringStats:struct<inputZCubeFiles:struct<numFiles:bigint,size:bigint>,inputOtherFiles:struct<numFiles:bigint,size:bigin
t>,inputNumZCubes:bigint,mergedFiles:struct<numFiles:bigint,size:bigint>,numOutputZCubes:bigint>,numBins:bigint,numBatches:bigint,totalConsideredFiles:bigint,total
FilesSkipped:bigint,preserveInsertionOrder:boolean,numFilesSkippedToReduceWriteAmplification:bigint,numBytesSkippedToReduceWriteAmplification:bigint,startTimeMs:bi
gint,endTimeMs:bigint,totalClusterParallelism:bigint,totalScheduledTasks:bigint,autoCompactParallelismStats:struct<maxClusterActiveParallelism:bigint,minClusterAct
iveParallelism:bigint,maxSessionActiveParallelism:bigint,minSessionActiveParallelism:bigint>,deletionVectorStats:struct<numDeletionVectorsRemoved:bigint,numDeletio
nVectorRowsRemoved:bigint>,numTableColumns:bigint,numTableColumnsWithStats:bigint,totalTaskExecutionTimeMs:bigint,skippedArchivedFiles:bigint,clusteringMetrics:str
uct<sizeOfTableInBytesBeforeLazyClustering:bigint,isNewMetadataCreated:boolean,isPOTriggered:boolean,numFilesSkippedWithoutStats:bigint,numFilesClassifiedToInterme
diateNodes:bigint,sizeOfFilesClassifiedToIntermediateNodesInBytes:bigint,logicalSizeOfFilesClassifiedToIntermediateNodesInBytes:bigint,numFilesClassifiedToLeafNode
s:bigint,sizeOfFilesClassifiedToLeafNodesInBytes:bigint,logicalSizeOfFilesClassifiedToLeafNodesInBytes:bigint,numThreadsForClassifier:int,clusterThresholdStrategy:
string,minFileSize:bigint,maxFileSize:bigint,nodeMinNumFilesToCompact:bigint,numIdealFiles:bigint,numClusteringTasksPlanned:int,numCompactionTasksPlanned:int,numOp
timizeBatchesPlanned:int,numLeafNodesExpanded:bigint,numLeafNodesClustered:bigint,numGetFilesForNodeCalls:bigint,numSamplingJobs:bigint,numLeafNodesCompacted:bigin
t,numIntermediateNodesCompacted:bigint,totalSizeOfDataToCompactInBytes:bigint,totalLogicalSizeOfDataToCompactInBytes:bigint,numIntermediateNodesClustered:bigint,nu
mFilesSkippedAfterExpansion:bigint,totalSizeOfFilesSkippedAfterExpansionInBytes:bigint,totalLogicalSizeOfFilesSkippedAfterExpansionInBytes:bigint,totalSizeOfDataTo
RewriteInBytes:bigint,totalLogicalSizeOfDataToRewriteInBytes:bigint,timeMetrics:struct<classifierTimeMs:bigint,optimizerTimeMs:bigint,metadataLoadTimeMs:bigint,tot
alGetFilesForNodeCallsTimeMs:bigint,totalSamplingTimeMs:bigint,metadataCreationTimeMs:bigint>,maxOptimizeBatchesInParallel:bigint,currentIteration:int,maxIteration
s:int,clusteringStrategy:string>>]
```

✓ 12:59 PM (1s) 7: Time Travel with Delta Table

```
# Check the history to find the version numbers
spark.sql("DESCRIBE HISTORY delta_employee_table;").show()

# Retrieve the Delta table as it was at a specific version
spark.sql("SELECT * FROM delta_employee_table VERSION AS OF 0;").show()
```

▸ (3) Spark Jobs

```
erializable|        false|{numFiles -> 1, n...|        NULL|Databricks-Runtim...|
|      1|2024-09-12 12:01:24|8379095214579684|azuser2109_mml.lo...|    WRITE|{mode -> Overwrit...|NULL|{3714719545832430}|0911-102451-o8x6anfh|          0|WriteS
erializable|        false|{numFiles -> 1, n...|        NULL|Databricks-Runtim...|
|      0|2024-09-12 12:01:04|8379095214579684|azuser2109_mml.lo...|    WRITE|{mode -> Overwrit...|NULL|{3714719545832430}|0911-102451-o8x6anfh|       NULL|WriteS
erializable|        false|{numFiles -> 1, n...|        NULL|Databricks-Runtim...|
+-------+-------------------+----------------+--------------------+---------+--------------------+----+------------------+--------------------+-----------+------
----------+-------------+--------------------+------------+--------------------+

+----------+-------------+----------+-----------+------+
|EmployeeID|         Name|Department|JoiningDate|Salary|
+----------+-------------+----------+-----------+------+
|      1001|     John Doe|        HR| 2021-01-15| 55000|
|      1002|   Jane Smith|        IT| 2020-03-10| 62000|
|      1003|Emily Johnson|   Finance| 2019-07-01| 70000|
|      1004|Michael Brown|        HR| 2018-12-22| 54000|
|      1005| David Wilson|        IT| 2021-06-25| 58000|
|      1006|  Linda Davis|   Finance| 2020-11-15| 67000|
|      1007| James Miller|        IT| 2019-08-14| 65000|
|      1008|Barbara Moore|        HR| 2021-03-29| 53000|
+----------+-------------+----------+-----------+------+
```

✓ 01:00 PM (25s) 8: Vacuum Delta Table

```
# retain last 7 days record only
spark.sql("VACUUM delta_employee_table RETAIN 168 HOURS")
```

▸ (21) Spark Jobs

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