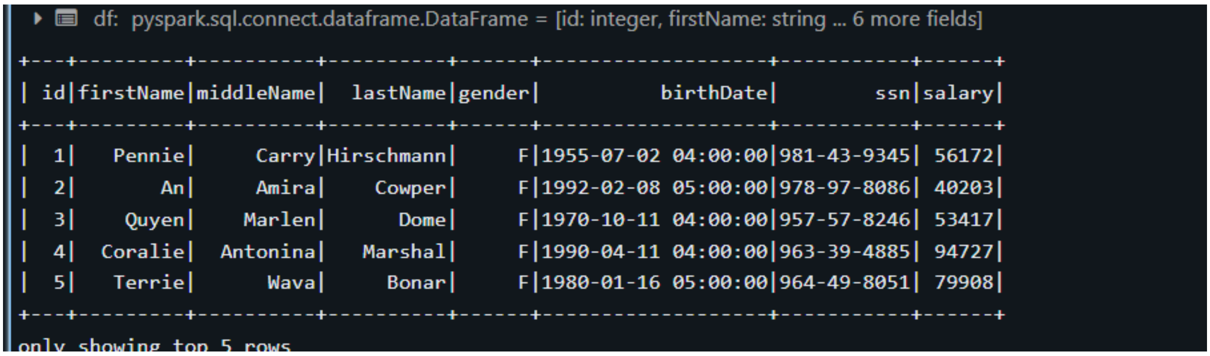
11.08.2025 – Task

Delta Lake Operations On Azure Databricks

1. Create a Delta Table

We start by loading a CSV file into a Spark DataFrame and saving it as a Delta table.

```
# Step 1: Load CSV file into a DataFrame

file_path = "/Volumes/workspace/default/delta/export.csv"

df = spark.read.format("csv") \

    .option("header", "true") \      # Use first row as header

    .option("inferSchema", "true") \ # Automatically detect data types

    .load(file_path)

df.show(5)  # Show first 5 rows of data
```



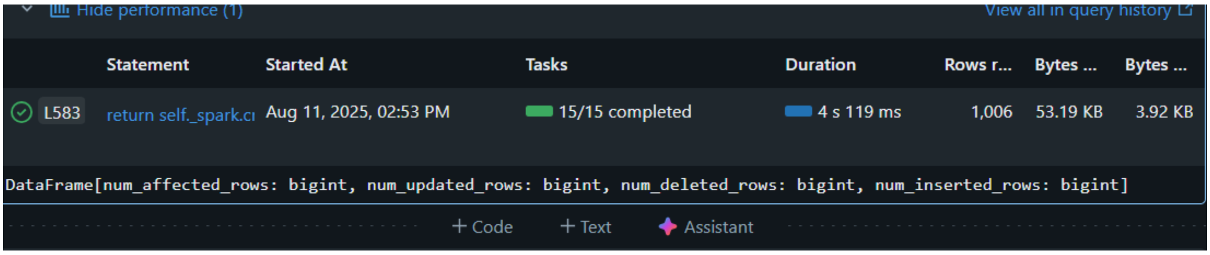
```
df:  pyspark.sql.connect.dataframe.DataFrame = [id: integer, firstName: string ... 6 more fields]

+---+---------+----------+----------+------+-------------------+-----------+------+
| id|firstName|middleName|  lastName|gender|          birthDate|        ssn|salary|
+---+---------+----------+----------+------+-------------------+-----------+------+
|  1|   Pennie|     Carry|Hirschmann|     F|1955-07-02 04:00:00|981-43-9345| 56172|
|  2|       An|     Amira|    Cowper|     F|1992-02-08 05:00:00|978-97-8086| 40203|
|  3|    Quyen|    Marlen|      Dome|     F|1970-10-11 04:00:00|957-57-8246| 53417|
|  4|  Coralie|  Antonina|   Marshal|     F|1990-04-11 04:00:00|963-39-4885| 94727|
|  5|   Terrie|      Wava|     Bonar|     F|1980-01-16 05:00:00|964-49-8051| 79908|
+---+---------+----------+----------+------+-------------------+-----------+------+
only showing top 5 rows
```

```
# Step 2: Save DataFrame as a Delta table named 'employees'
df.write.format("delta") \
    .mode("overwrite") \            # Overwrite if table exists
    .saveAsTable("employees")       # Save as managed Delta table
```

2. Upsert (Merge) Data into the Delta Table

We create new data to update existing rows or insert new ones, then perform a merge (upsert) operation.

```
from delta.tables import DeltaTable

# Prepare new data for upsert

new_data = [

    (3, "Quyen", "Marlen", "Dome", "F", "1970-10-11 04:00:00", "957-57-8246", 55000),  # Update salary for id=3

    (6, "John", "M", "Doe", "M", "1985-05-10 04:00:00", "123-45-6789", 60000)          # New employee with id=6

]

columns = ["id", "firstName", "middleName", "lastName", "gender", "birthDate", "ssn", "salary"]

new_df = spark.createDataFrame(new_data, columns)


# Load Delta table and perform merge

deltaTable = DeltaTable.forName(spark, "employees")

deltaTable.alias("old").merge(

    new_df.alias("new"),

    "old.id = new.id"  # Match records on 'id'

).whenMatchedUpdate(set =

  {

    "firstName": "new.firstName",

    "middleName": "new.middleName",

    "lastName": "new.lastName",

    "gender": "new.gender",
    "birthDate": "new.birthDate",
    "ssn": "new.ssn",
    "salary": "new.salary"
  }
).whenNotMatchedInsert(values =
  {
    "id": "new.id",
    "firstName": "new.firstName",
    "middleName": "new.middleName",
    "lastName": "new.lastName",
    "gender": "new.gender",
    "birthDate": "new.birthDate",
    "ssn": "new.ssn",
    "salary": "new.salary"
  }
).execute()
```



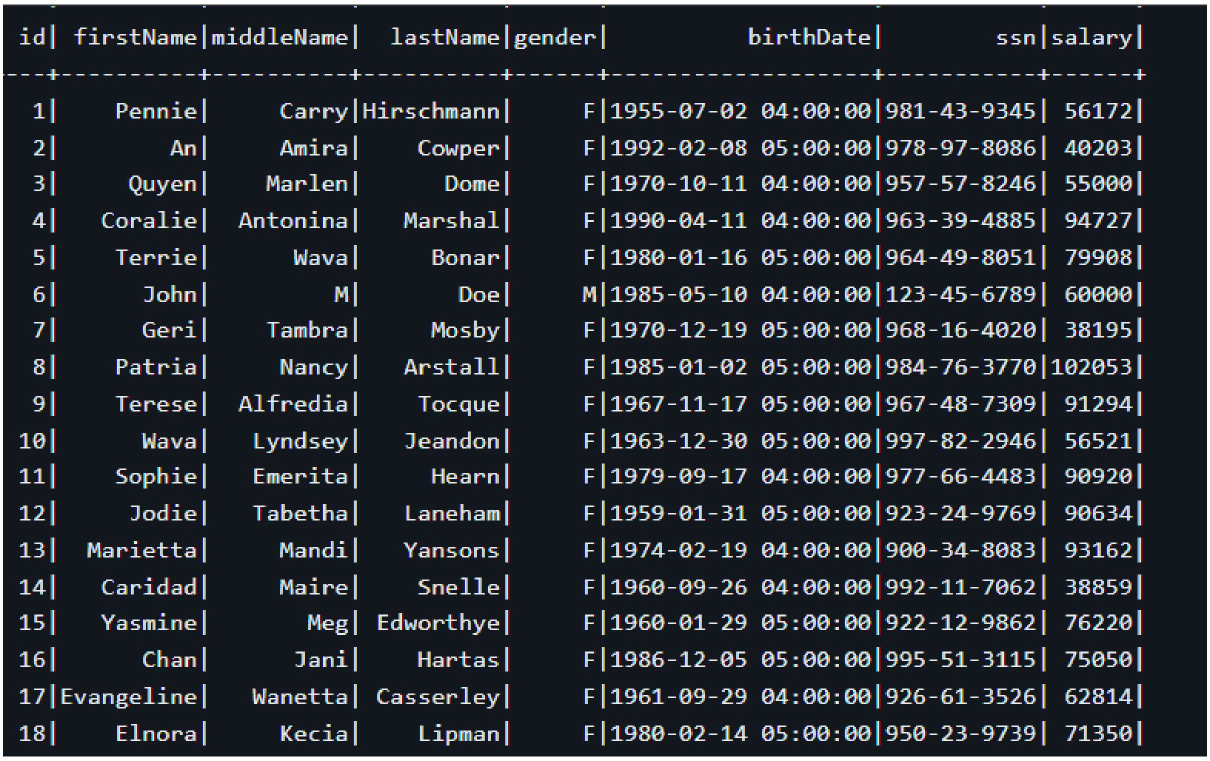
	Statement	Started At	Tasks	Duration	Rows r...	Bytes ...	Bytes ...
✓ L583	return self._spark.cı	Aug 11, 2025, 02:53 PM	🟩 15/15 completed	4 s 119 ms	1,006	53.19 KB	3.92 KB

```
DataFrame[num_affected_rows: bigint, num_updated_rows: bigint, num_deleted_rows: bigint, num_inserted_rows: bigint]
```

3. Read Data from the Delta Table

To verify the upsert operation, we read and display all data from the Delta table.

spark.sql("SELECT * FROM employees ORDER BY id").show()



```
 id| firstName|middleName|  lastName|gender|          birthDate|        ssn|salary|
---+----------+----------+----------+------+-------------------+-----------+------+
  1|    Pennie|     Carry|Hirschmann|     F|1955-07-02 04:00:00|981-43-9345| 56172|
  2|        An|     Amira|    Cowper|     F|1992-02-08 05:00:00|978-97-8086| 40203|
  3|     Quyen|    Marlen|      Dome|     F|1970-10-11 04:00:00|957-57-8246| 55000|
  4|   Coralie|  Antonina|   Marshal|     F|1990-04-11 04:00:00|963-39-4885| 94727|
  5|    Terrie|      Wava|     Bonar|     F|1980-01-16 05:00:00|964-49-8051| 79908|
  6|      John|         M|       Doe|     M|1985-05-10 04:00:00|123-45-6789| 60000|
  7|      Geri|    Tambra|     Mosby|     F|1970-12-19 05:00:00|968-16-4020| 38195|
  8|    Patria|     Nancy|   Arstall|     F|1985-01-02 05:00:00|984-76-3770|102053|
  9|    Terese|  Alfredia|    Tocque|     F|1967-11-17 05:00:00|967-48-7309| 91294|
 10|      Wava|   Lyndsey|   Jeandon|     F|1963-12-30 05:00:00|997-82-2946| 56521|
 11|    Sophie|   Emerita|     Hearn|     F|1979-09-17 04:00:00|977-66-4483| 90920|
 12|     Jodie|   Tabetha|   Laneham|     F|1959-01-31 05:00:00|923-24-9769| 90634|
 13|  Marietta|     Mandi|   Yansons|     F|1974-02-19 04:00:00|900-34-8083| 93162|
 14|   Caridad|     Maire|    Snelle|     F|1960-09-26 04:00:00|992-11-7062| 38859|
 15|   Yasmine|       Meg| Edworthye|     F|1960-01-29 05:00:00|922-12-9862| 76220|
 16|      Chan|      Jani|    Hartas|     F|1986-12-05 05:00:00|995-51-3115| 75050|
 17|Evangeline|   Wanetta|  Casserley|    F|1961-09-29 04:00:00|926-61-3526| 62814|
 18|    Elnora|     Kecia|    Lipman|     F|1980-02-14 05:00:00|950-23-9739| 71350|
```

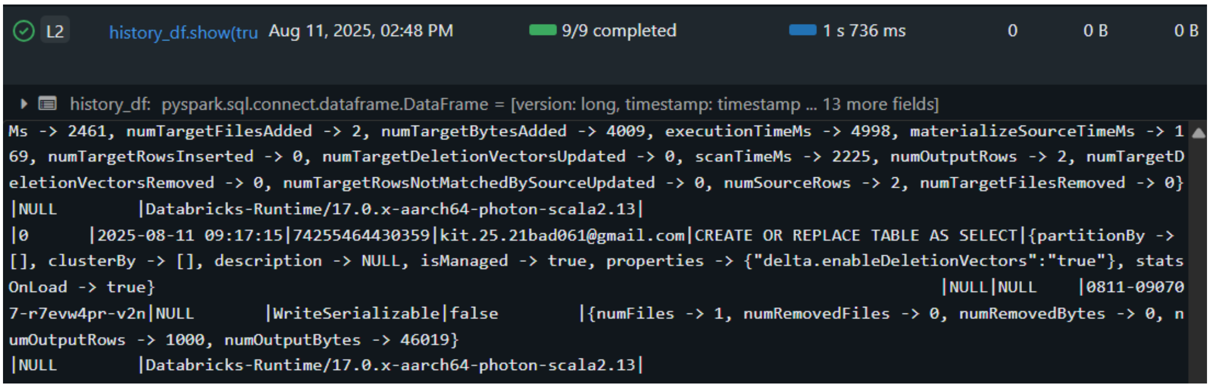
---

4. Display the Table History

Delta Lake maintains a transaction log. We can display the full history of all operations on the table.

history_df = deltaTable.history()

history_df.show(truncate=False)



```
history_df:  pyspark.sql.connect.dataframe.DataFrame = [version: long, timestamp: timestamp ... 13 more fields]
Ms -> 2461, numTargetFilesAdded -> 2, numTargetBytesAdded -> 4009, executionTimeMs -> 4998, materializeSourceTimeMs -> 1
69, numTargetRowsInserted -> 0, numTargetDeletionVectorsUpdated -> 0, scanTimeMs -> 2225, numOutputRows -> 2, numTargetD
eletionVectorsRemoved -> 0, numTargetRowsNotMatchedBySourceUpdated -> 0, numSourceRows -> 2, numTargetFilesRemoved -> 0}
|NULL        |Databricks-Runtime/17.0.x-aarch64-photon-scala2.13|
|0      |2025-08-11 09:17:15|74255464430359|kit.25.21bad061@gmail.com|CREATE OR REPLACE TABLE AS SELECT|{partitionBy ->
[], clusterBy -> [], description -> NULL, isManaged -> true, properties -> {"delta.enableDeletionVectors":"true"}, stats
OnLoad -> true}                                                                                      |NULL|NULL    |0811-09070
7-r7evw4pr-v2n|NULL        |WriteSerializable|false        |{numFiles -> 1, numRemovedFiles -> 0, numRemovedBytes -> 0, n
umOutputRows -> 1000, numOutputBytes -> 46019}
|NULL        |Databricks-Runtime/17.0.x-aarch64-photon-scala2.13|
```