

Name: Rohan Vinayak Chaudhari

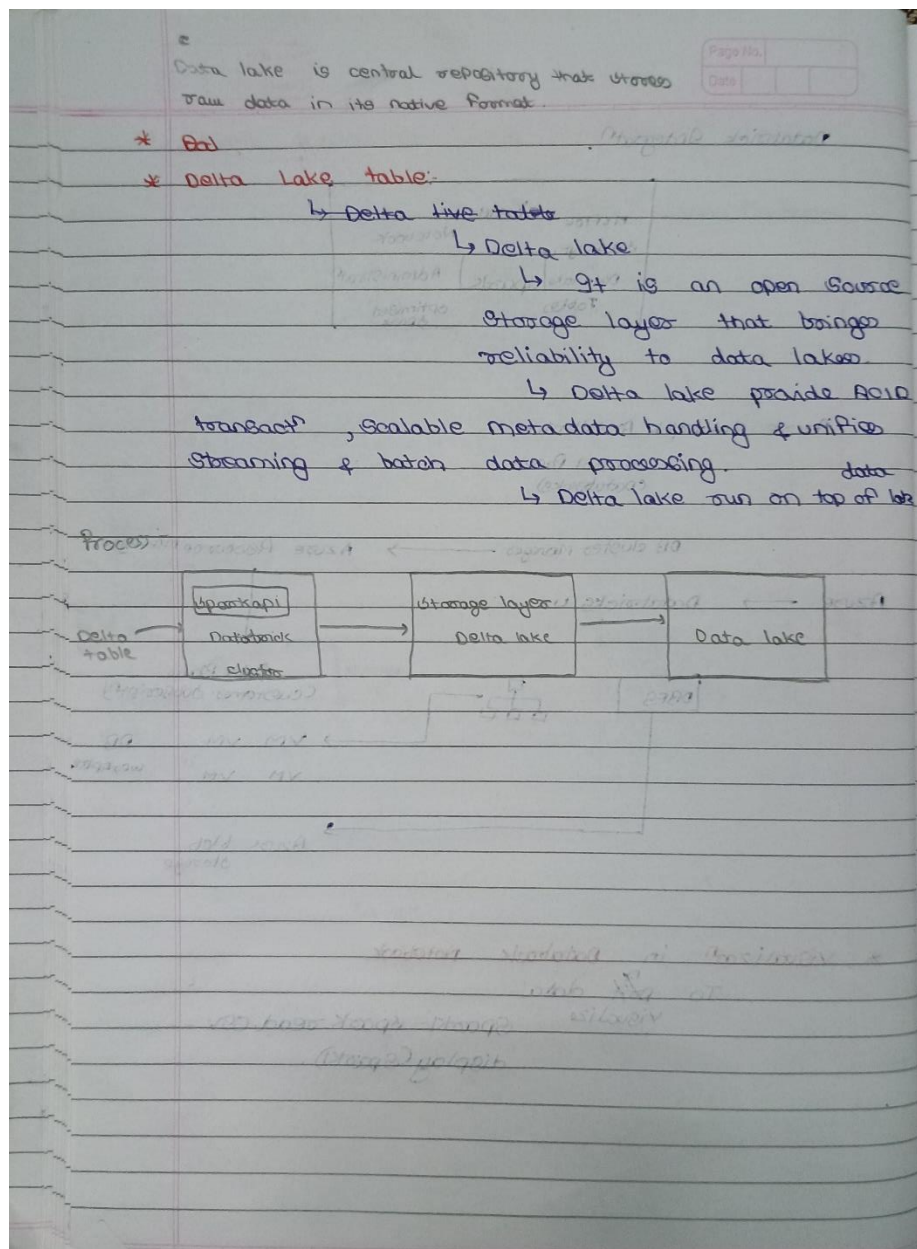
Batch: Data Engineering

Date: 14/02/2024

Topic: Azure DataBricks

Solution:

1. Azure Databricks (Delta lakes):



Creating delta lake table with SQL:

The screenshot shows the Databricks SQL Editor interface. The top bar includes the Microsoft Azure logo, the Databricks logo, a search bar, and the text "CTRL + P". The user is logged in as "azuser1076_mml.local@ih1t1.onmicr...". The left sidebar shows the "Workspace" tab selected, with a list of items including "Recents", "Catalog", "Workflows", "Compute", "SQL", "SQL Editor", "Queries", "Dashboards", "Alerts", "Query History", and "SQL Warehouses". The main editor area is titled "Deltalake_Sq 2024-02-14 15:19:02" and shows a "Create table" command in a code cell. The command is: `CREATE TABLE delta.`/tmp/delta-table` USING DELTA AS SELECT col1 as id FROM VALUES 0,1,2,3,4;`. Below the code, it shows the execution results: "(6) Spark Jobs", "._sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]", and "Query returned no results". A message indicates: "This result is stored as PySpark data frame ._sqldf and in the IPython output cache as Out[1]. Learn more". The command took 18.26 seconds to execute.

```
1 %sql
2 CREATE TABLE delta.`/tmp/delta-table` USING DELTA AS SELECT col1 as id FROM VALUES 0,1,2,3,4;
```

(6) Spark Jobs
._sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]
Query returned no results

This result is stored as PySpark data frame ._sqldf and in the IPython output cache as Out[1]. Learn more

Command took 18.26 seconds -- by azuser1076_mml.local@ih1t1.onmicrosoft.com at 2/14/2024, 3:19:56 PM on azuser1076_mml.local's Cluster

Viewing the data:

The screenshot shows the Databricks SQL Editor interface. The top bar includes the Microsoft Azure logo, the Databricks logo, a search bar, and the text "CTRL + P". The user is logged in as "azuser1076_mml.local@ih1t1.onmicr...". The left sidebar shows the "Workspace" tab selected, with a list of items including "Recents", "Catalog", "Workflows", "Compute", "SQL", "SQL Editor", "Queries", "Dashboards", "Alerts", "Query History", and "SQL Warehouses". The main editor area is titled "Deltalake_Sq 2024-02-14 15:19:02" and shows a "SELECT * FROM delta.`/tmp/delta-table`;" command in a code cell. Below the code, it shows the execution results: "(1) Spark Jobs", "._sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]", and a table view of the data. The table has 5 rows and 1 column, with values 0, 1, 2, 3, and 4. A message indicates: "This result is stored as PySpark data frame ._sqldf and in the IPython output cache as Out[2]. Learn more". The command took 1.48 seconds to execute.

```
1 %sql
2 SELECT * FROM delta.`/tmp/delta-table`;
```

(1) Spark Jobs
._sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]

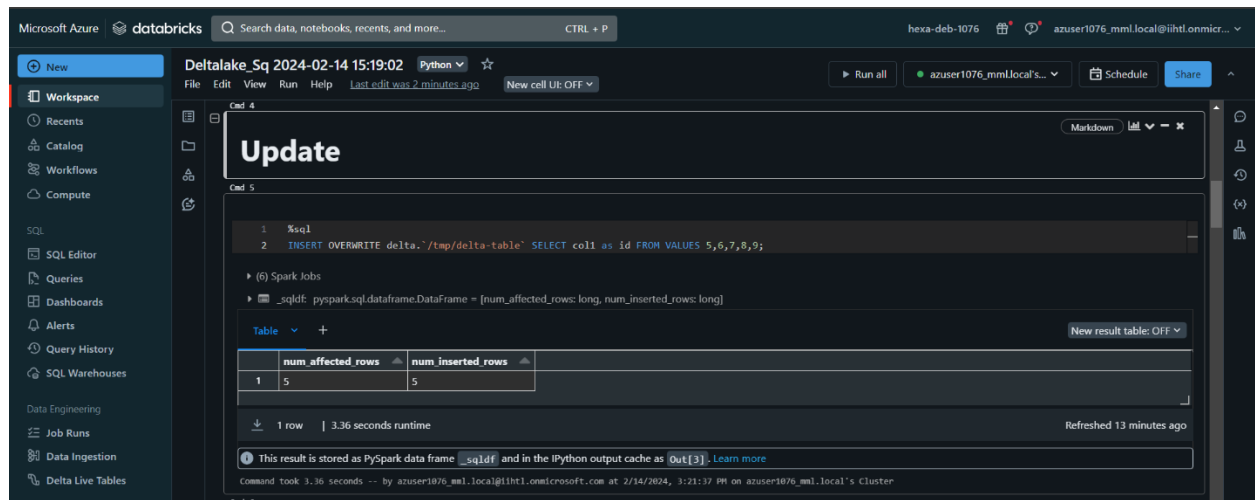
	id
1	0
2	1
3	2
4	3
5	4

5 rows | 1.48 seconds runtime

This result is stored as PySpark data frame ._sqldf and in the IPython output cache as Out[2]. Learn more

Command took 1.48 seconds -- by azuser1076_mml.local@ih1t1.onmicrosoft.com at 2/14/2024, 3:28:31 PM on azuser1076_mml.local's Cluster

Updating data with overwrite:



The screenshot shows the Databricks workspace interface. The top navigation bar includes the Microsoft Azure logo, the Databricks logo, a search bar, and the text "CTRL + P". The user profile "azuser1076_mml.local@ih1t1.onmicr..." is visible in the top right. The left sidebar contains a "New" button and a "Workspace" section with various navigation options. The main area displays a notebook titled "Deltalake_Sq 2024-02-14 15:19:02" in Python. The notebook has a cell labeled "Cmd 4" with the following SQL code:

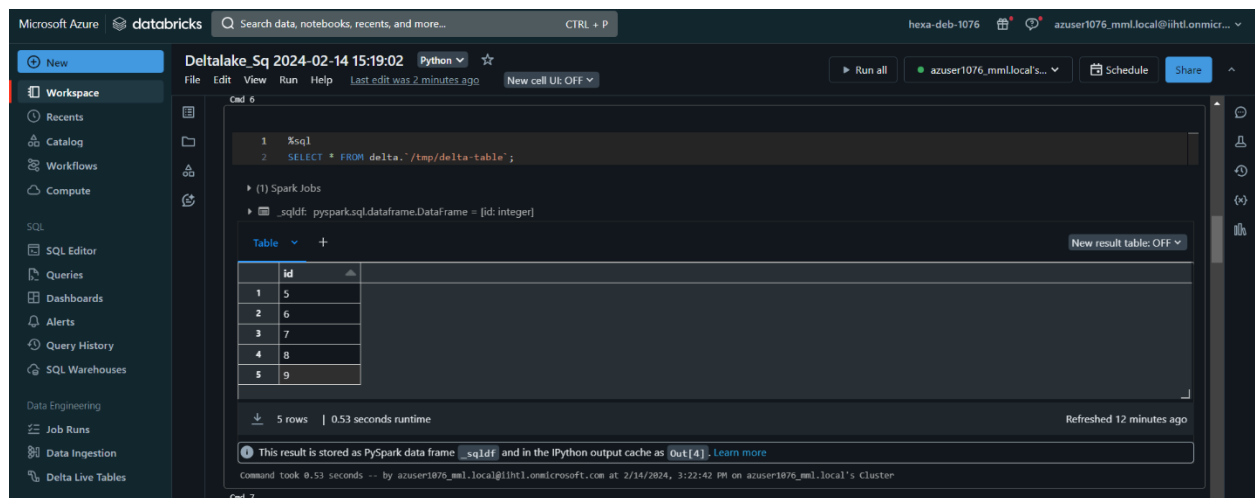
```
1 %sql
2 INSERT OVERWRITE delta.`/tmp/delta-table` SELECT col1 as id FROM VALUES 5,6,7,8,9;
```

Below the code, the execution results are shown. It indicates that 6 Spark Jobs were executed, and the _sqlidf variable is a pyspark.sql.dataframe.DataFrame with columns num_affected_rows and num_inserted_rows. The results table shows:

	num_affected_rows	num_inserted_rows
1	5	5

The table has 1 row and a runtime of 3.36 seconds. The results were refreshed 13 minutes ago. A message at the bottom states: "This result is stored as PySpark data frame _sqlidf and in the IPython output cache as Out[3]. Learn more".

Getting the Updated Data:



The screenshot shows the Databricks workspace interface. The top navigation bar includes the Microsoft Azure logo, the Databricks logo, a search bar, and the text "CTRL + P". The user profile "azuser1076_mml.local@ih1t1.onmicr..." is visible in the top right. The left sidebar contains a "New" button and a "Workspace" section with various navigation options. The main area displays a notebook titled "Deltalake_Sq 2024-02-14 15:19:02" in Python. The notebook has a cell labeled "Cmd 6" with the following SQL code:

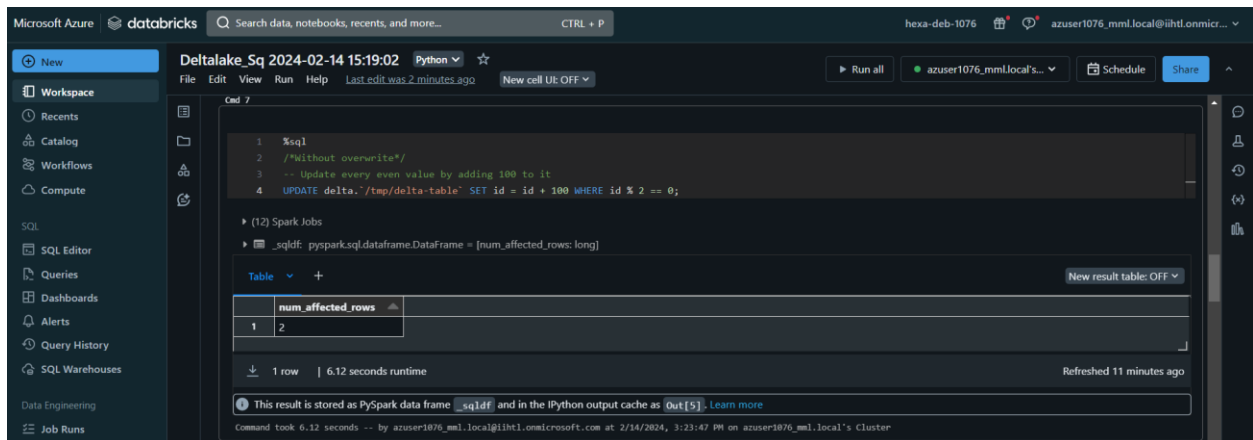
```
1 %sql
2 SELECT * FROM delta.`/tmp/delta-table`;
```

Below the code, the execution results are shown. It indicates that 1 Spark Job was executed, and the _sqlidf variable is a pyspark.sql.dataframe.DataFrame with column [id: integer]. The results table shows:

	id
1	5
2	6
3	7
4	8
5	9

The table has 5 rows and a runtime of 0.53 seconds. The results were refreshed 12 minutes ago. A message at the bottom states: "This result is stored as PySpark data frame _sqlidf and in the IPython output cache as Out[4]. Learn more".

Updating Data without Overwrite:



The screenshot shows the Databricks SQL Editor interface. The query editor contains the following SQL code:

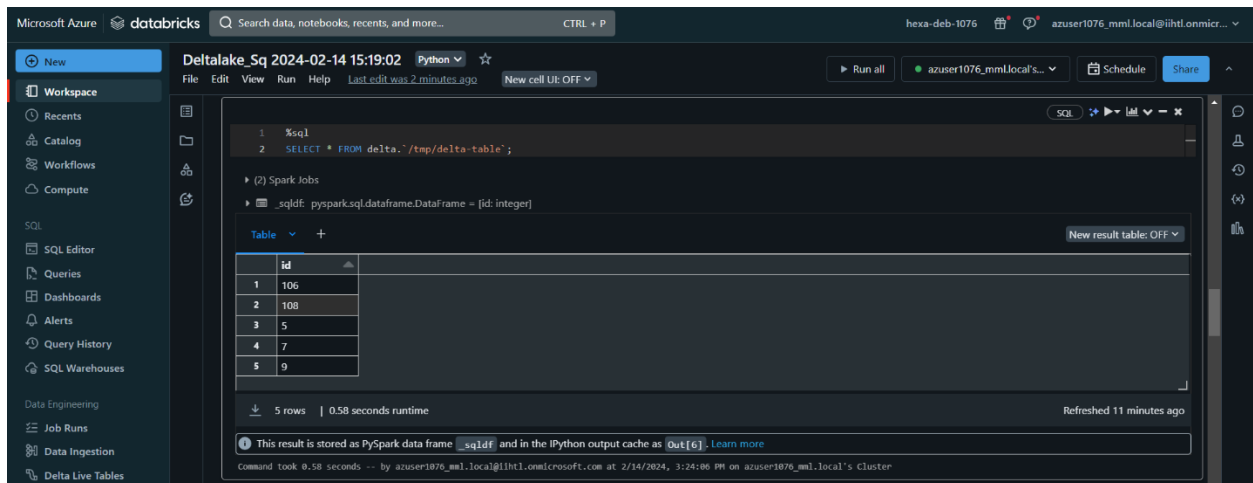
```
1 %sql
2 /*Without overwrite*/
3 -- Update every even value by adding 100 to it
4 UPDATE delta.`/tmp/delta-table` SET id = id + 100 WHERE id % 2 == 0;
```

The execution results show a table with one row:

num_affected_rows
2

The table has 1 row and a runtime of 6.12 seconds. The result is stored as a PySpark data frame `_sqldf` and in the IPython output cache as `Out[5]`.

Viewing the updated data:



The screenshot shows the Databricks SQL Editor interface. The query editor contains the following SQL code:

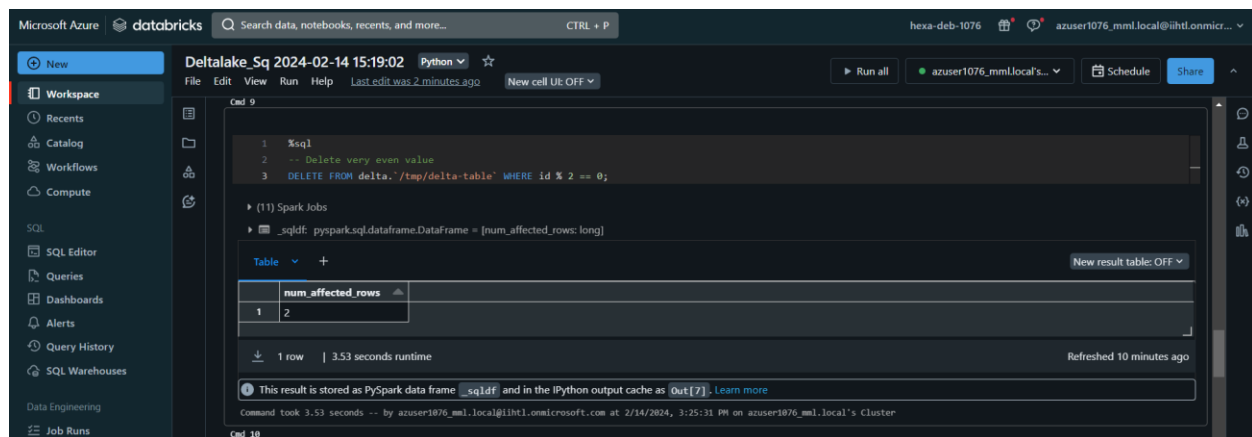
```
1 %sql
2 SELECT * FROM delta.`/tmp/delta-table`;
```

The execution results show a table with five rows:

id
1
2
3
4
5

The table has 5 rows and a runtime of 0.58 seconds. The result is stored as a PySpark data frame `_sqldf` and in the IPython output cache as `Out[6]`.

Deleting Even Data:

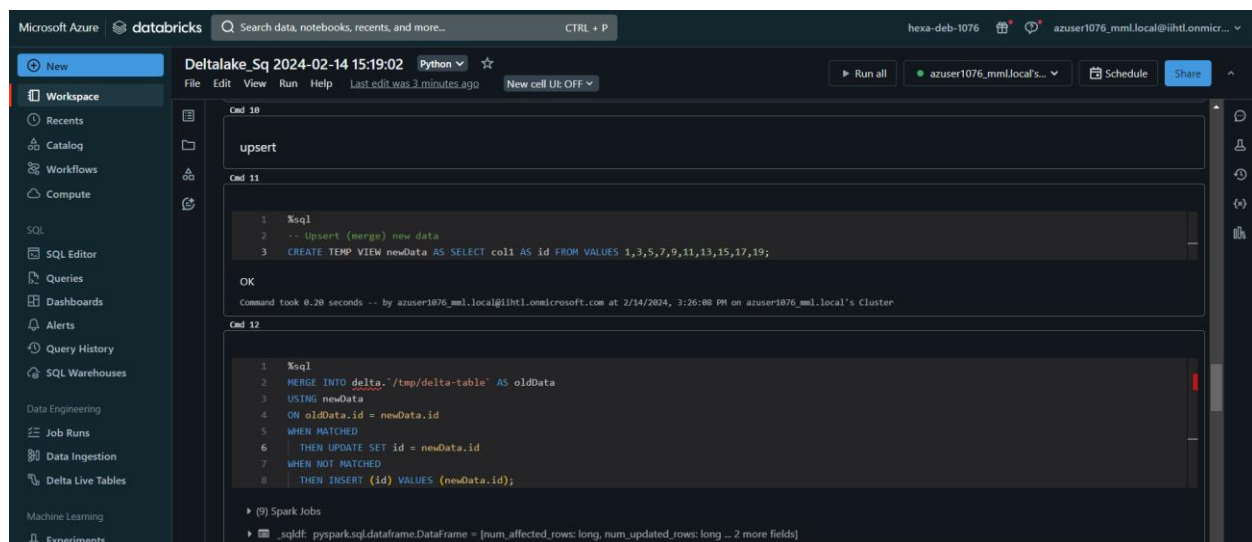


The screenshot shows the Databricks workspace interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, SQL, SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses, Data Engineering, and Job Runs. The main area displays a notebook titled "Deltalake_Sq 2024-02-14 15:19:02" with a Python language setting. The notebook content shows a SQL query in a code cell:

```
1 %sql
2 -- Delete very even value
3 DELETE FROM delta.`/tmp/delta-table` WHERE id % 2 == 0;
```

Below the code cell, the execution results are shown. It indicates that 1 Spark Job was executed, and a DataFrame with 1 row and 1 column (num_affected_rows) was returned. The value in the row is 2. The runtime was 3.53 seconds. A message at the bottom states: "This result is stored as PySpark data frame _sqlddf and in the IPython output cache as Out[7]. Learn more".

Upsert&Merge operations:



The screenshot shows the Databricks workspace interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, SQL, SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, Delta Live Tables, Machine Learning, and Experiments. The main area displays a notebook titled "Deltalake_Sq 2024-02-14 15:19:02" with a Python language setting. The notebook content shows two SQL queries in code cells:

```
Cmd 10
upsert

Cmd 11
1 %sql
2 -- Upsert (merge) new data
3 CREATE TEMP VIEW newData AS SELECT col1 AS id FROM VALUES 1,3,5,7,9,11,13,15,17,19;

OK
Command took 0.20 seconds -- by azuser1076_mml.local@iitl.onmicrosoft.com at 2/14/2024, 3:26:08 PM on azuser1076_mml.local's Cluster

Cmd 12
1 %sql
2 MERGE INTO delta.`/tmp/delta-table` AS oldData
3 USING newData
4 ON oldData.id = newData.id
5 WHEN MATCHED
6 THEN UPDATE SET id = newData.id
7 WHEN NOT MATCHED
8 THEN INSERT (id) VALUES (newData.id);

(9) Spark Jobs
_sqlddf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_updated_rows: long ... 2 more fields]
```

Viewing Upserted data:

The screenshot shows the Databricks SQL Editor interface. The top bar indicates the workspace is 'Deltalake_Sq' and the session is '2024-02-14 15:19:02'. The left sidebar contains navigation options like 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute', 'SQL', 'SQL Editor', 'Queries', 'Dashboards', 'Alerts', 'Query History', 'SQL Warehouses', 'Data Engineering', 'Job Runs', 'Data Ingestion', 'Delta Live Tables', and 'Machine Learning'. The main editor area shows a SQL query: `SELECT * FROM delta.`/tmp/delta-table`;`. Below the query, a Spark job is shown with the output: `_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]`. A table view displays 10 rows of data:

	id
1	5
2	7
3	9
4	15
5	17
6	19
7	1
8	3
9	11
10	13

The table view indicates 10 rows and a runtime of 0.76 seconds. The bottom right corner shows 'Refreshed 9 minutes ago'.

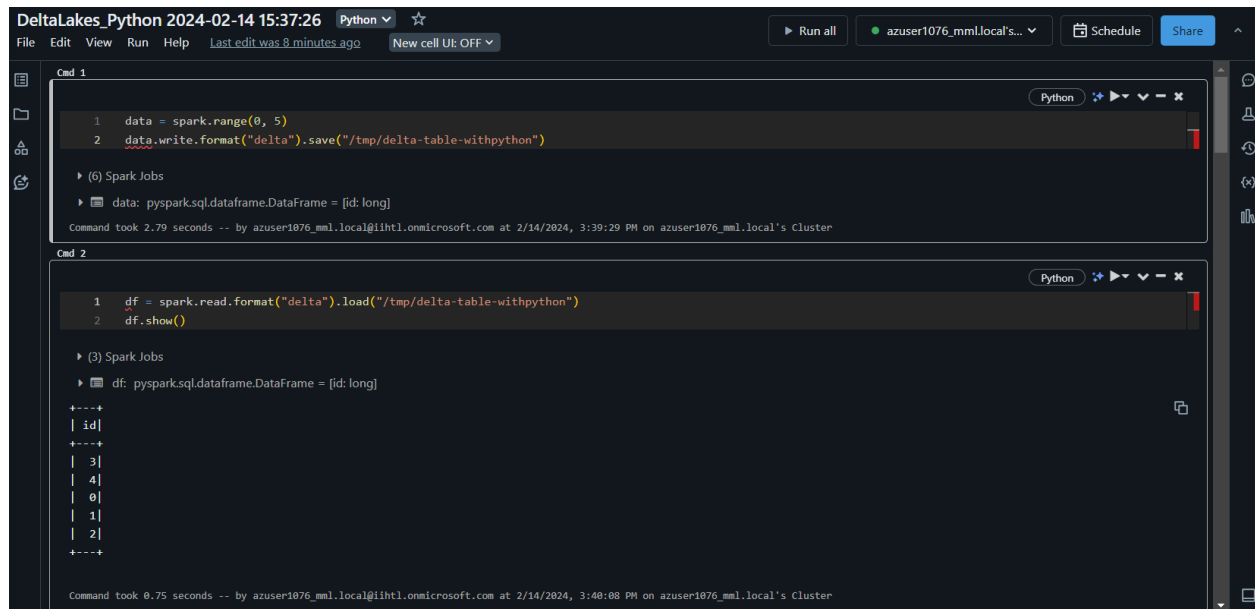
Getting older data:

The screenshot shows the Databricks SQL Editor interface. The top bar indicates the workspace is 'Deltalake_Sq' and the session is '2024-02-14 15:19:02'. The left sidebar contains navigation options like 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute', 'SQL', 'SQL Editor', 'Queries', 'Dashboards', 'Alerts', 'Query History', 'SQL Warehouses', 'Data Engineering', 'Job Runs', 'Data Ingestion', 'Delta Live Tables', and 'Machine Learning'. The main editor area shows a SQL query: `SELECT * FROM delta.`/tmp/delta-table` VERSION AS OF 0;`. Below the query, a Spark job is shown with the output: `_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]`. A table view displays 5 rows of data:

	id
1	0
2	1
3	2
4	3
5	4

The table view indicates 5 rows and a runtime of 0.50 seconds. The bottom right corner shows 'Refreshed 6 minutes ago'. Below the table view, a message states: 'This result is stored as PySpark data frame `_sqldf` and in the Python output cache as `Out[10]`. [Learn more](#)'. The bottom status bar shows 'Command took 0.50 seconds ... by azuser1076_mml.local@ih1.onmicrosoft.com at 2/14/2024, 3:30:19 PM on azuser1076_mml.local's Cluster'.

Delta lakes Working with Python:



DeltaLakes_Python 2024-02-14 15:37:26 Python

File Edit View Run Help Last edit was 8 minutes ago New cell UI: Off

Run all azuser1076_mml.local's... Schedule Share

Cmd 1

```
1 data = spark.range(0, 5)
2 data.write.format("delta").save("/tmp/delta-table-withpython")
```

▶ (6) Spark Jobs

data: pyspark.sql.dataframe.DataFrame = [id: long]

Command took 2.79 seconds -- by azuser1076_mml.local@iitl.onmicrosoft.com at 2/14/2024, 3:39:29 PM on azuser1076_mml.local's Cluster

Cmd 2

```
1 df = spark.read.format("delta").load("/tmp/delta-table-withpython")
2 df.show()
```

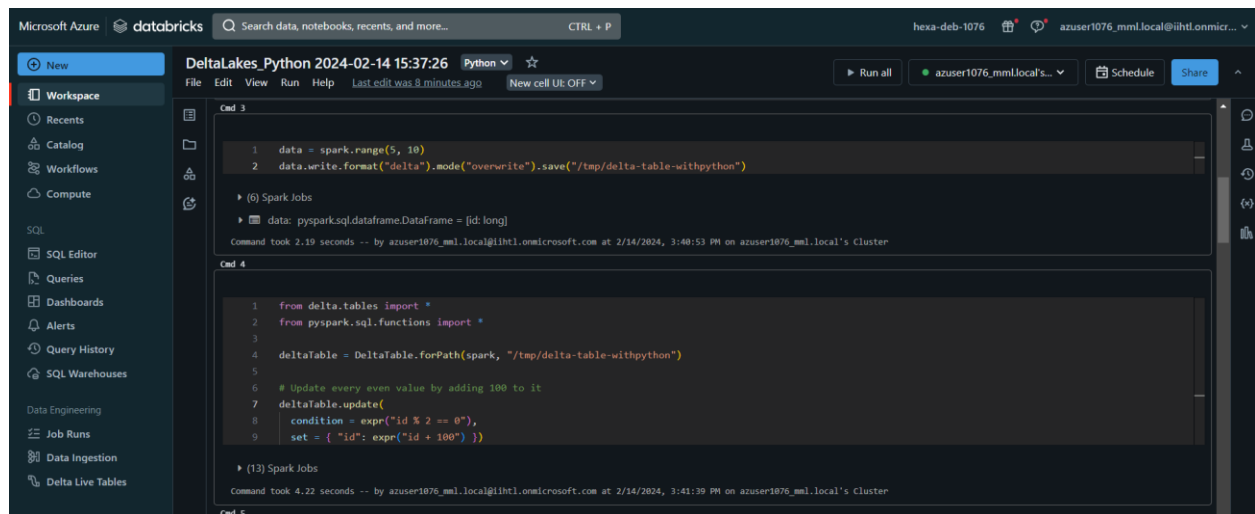
▶ (3) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [id: long]

```
+----+
| id |
+----+
| 3   |
| 4   |
| 0   |
| 1   |
| 2   |
+----+
```

Command took 0.75 seconds -- by azuser1076_mml.local@iitl.onmicrosoft.com at 2/14/2024, 3:40:08 PM on azuser1076_mml.local's Cluster

Overwrite&without overwrite:



Microsoft Azure databricks Search data, notebooks, recents, and more... CTRL + P hexa-deb-1076 azuser1076_mml.local@iitl.onmicr...

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning

DeltaLakes_Python 2024-02-14 15:37:26 Python

File Edit View Run Help Last edit was 8 minutes ago New cell UI: Off

Run all azuser1076_mml.local's... Schedule Share

Cmd 3

```
1 data = spark.range(5, 10)
2 data.write.format("delta").mode("overwrite").save("/tmp/delta-table-withpython")
```

▶ (6) Spark Jobs

data: pyspark.sql.dataframe.DataFrame = [id: long]

Command took 2.19 seconds -- by azuser1076_mml.local@iitl.onmicrosoft.com at 2/14/2024, 3:40:53 PM on azuser1076_mml.local's Cluster

Cmd 4

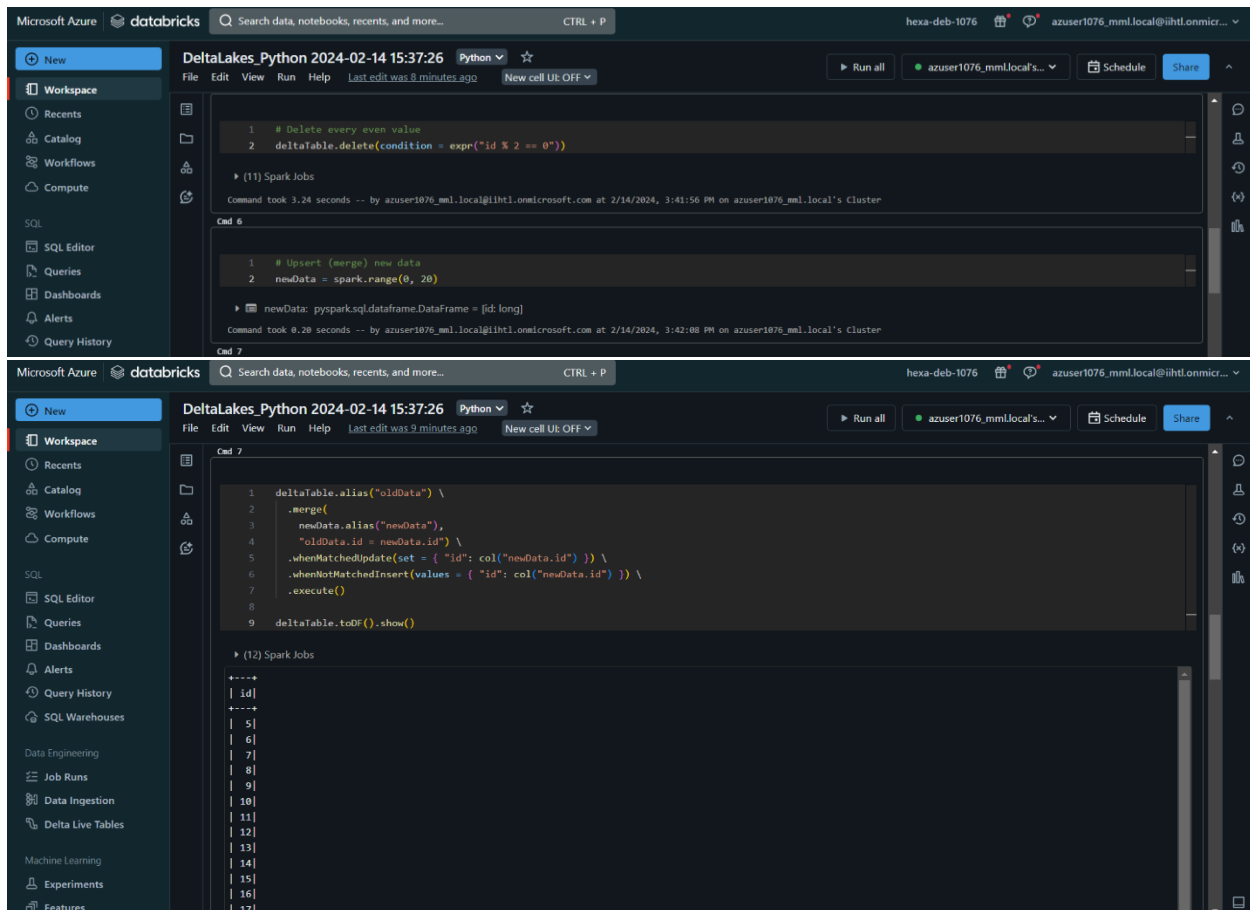
```
1 from delta.tables import *
2 from pyspark.sql.functions import *
3
4 deltaTable = DeltaTable.forPath(spark, "/tmp/delta-table-withpython")
5
6 # Update every even value by adding 100 to it
7 deltaTable.update(
8     condition = expr("id % 2 == 0"),
9     set = { "id": expr("id + 100") })
```

▶ (13) Spark Jobs

Command took 4.22 seconds -- by azuser1076_mml.local@iitl.onmicrosoft.com at 2/14/2024, 3:41:39 PM on azuser1076_mml.local's Cluster

Cmd 5

Delete , Upsert&retriving new data in Python:



The screenshot displays two Databricks notebooks. The first notebook, titled "DeltaLakes_Python 2024-02-14 15:37:26", contains two commands. Command 6 shows a delete operation on even values and an upsert operation. Command 7 shows a merge operation. The second notebook, also titled "DeltaLakes_Python 2024-02-14 15:37:26", contains a single command (Cmd 7) that performs a merge operation. The execution results for both notebooks are shown in the console.

```
1 # Delete every even value
2 deltaTable.delete(condition = expr("id % 2 == 0"))

> (11) Spark Jobs

Command took 3.24 seconds -- by azuser1076_mml.local@ihti1.onmicrosoft.com at 2/14/2024, 3:41:56 PM on azuser1076_mml.local's Cluster

Cmd 6

1 # Upsert (merge) new data
2 newData = spark.range(0, 20)

> newData: pyspark.sql.dataframe.DataFrame = [id: long]

Command took 0.20 seconds -- by azuser1076_mml.local@ihti1.onmicrosoft.com at 2/14/2024, 3:42:00 PM on azuser1076_mml.local's Cluster

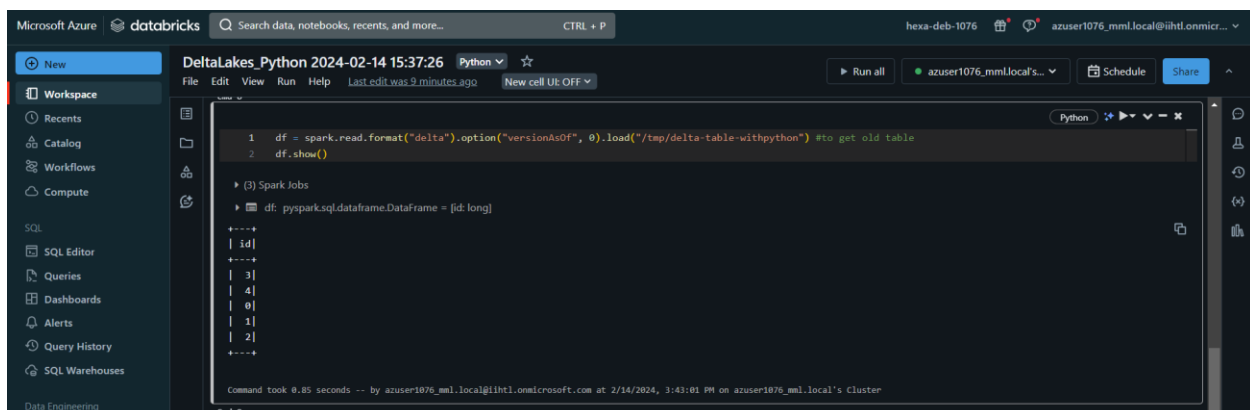
Cmd 7

1 deltaTable.alias("oldData") \
2   .merge(
3     newData.alias("newData"),
4     "oldData.id = newData.id" ) \
5     .whenMatchedUpdate(set = { "id": col("newData.id") }) \
6     .whenNotMatchedInsert(values = { "id": col("newData.id") }) \
7     .execute()
8
9 deltaTable.toDF().show()
```

Execution results for Cmd 7:

```
++---+
| id|
++---+
| 5|
| 6|
| 7|
| 8|
| 9|
|10|
|11|
|12|
|13|
|14|
|15|
|16|
|17|
```

Getting older data:



The screenshot displays a Databricks notebook titled "DeltaLakes_Python 2024-02-14 15:37:26". The notebook contains a single command (Cmd 9) that reads data from a Delta table using the 'versionAsOf' option to get older data. The execution results are shown in the console.

```
1 df = spark.read.format("delta").option("versionAsOf", 0).load("/tmp/delta-table-withpython") #to get old table
2 df.show()
```

Execution results for Cmd 9:

```
++---+
| id|
++---+
| 3|
| 4|
| 0|
| 1|
| 2|
```


Streaming Data:

