Name: Rohan Vinayak Chaudhari

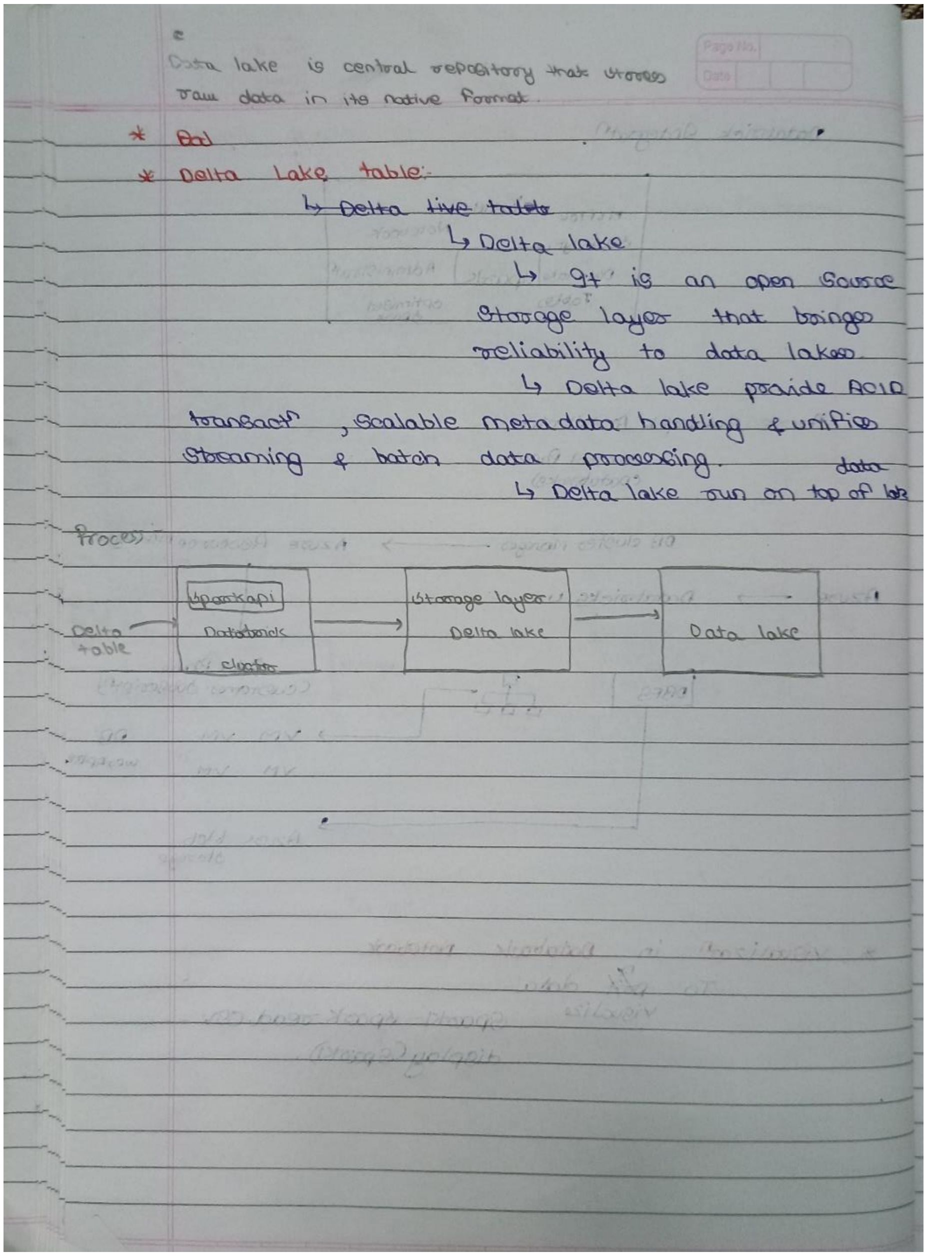
Batch: Data Engineering

Date:14/02/2024

Topic: Azure DataBricks

Solution:

1.Azure Databricks(Delta lakes):



Data lake is central repository that stores raw data in its native format.

* ~~Bal~~

* Delta Lake table:-

↳ ~~Delta live table~~

↳ Delta lake

↳ It is an open Source Storage layer that brings reliability to data lakes.

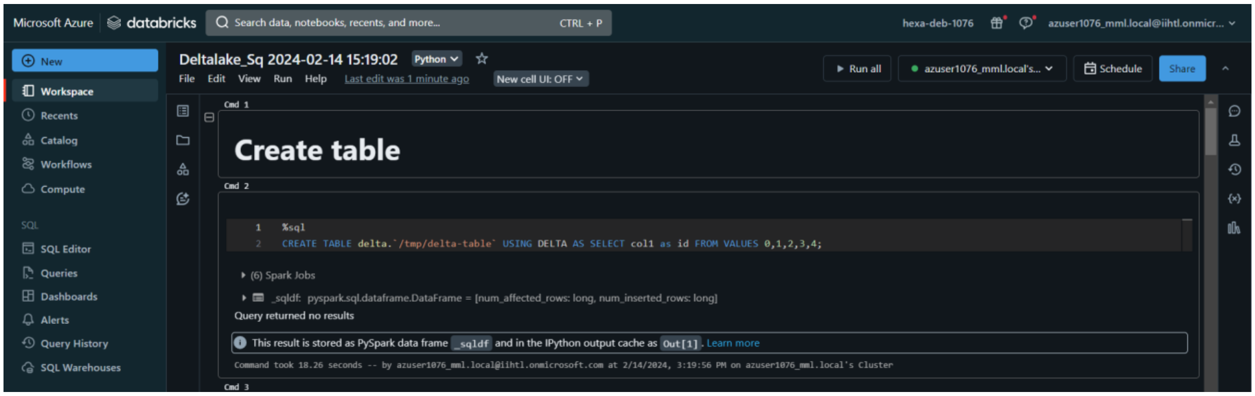
↳ Delta lake provide ACID transact^n, Scalable metadata handling & unifies Streaming & batch data processing.

↳ Delta lake run on top of data lake

Process

Delta table
Sparkapi
Databricks
cluster
Storage layer
Delta lake
Data lake

Creating delta lake table with SQL:

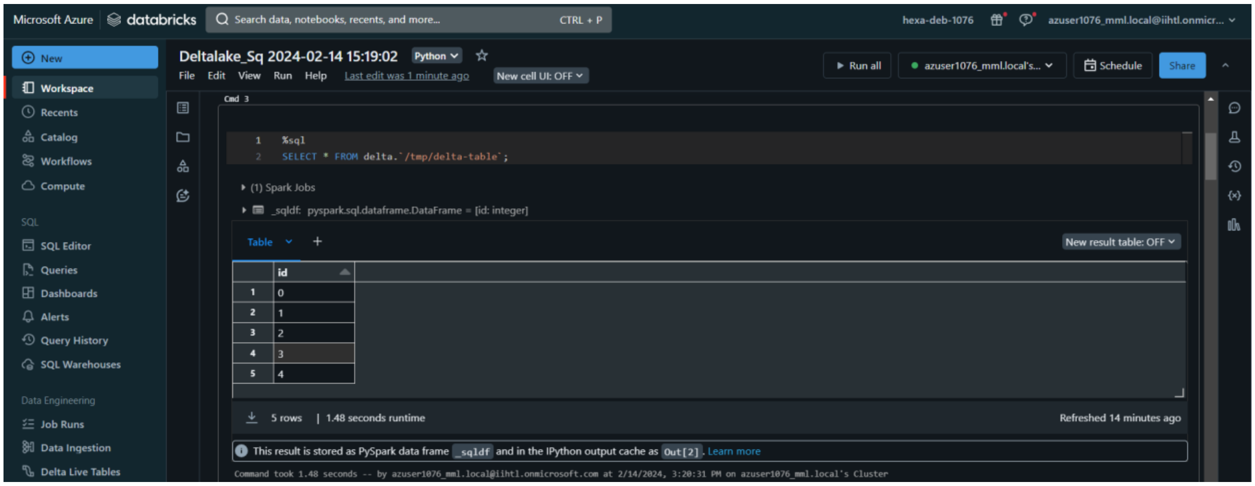


Create table

```
%sql
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@iihtl.onmicrosoft.com at 2/14/2024, 3:19:56 PM on azuser1076_mml.local's Cluster
```

Viewing the data:



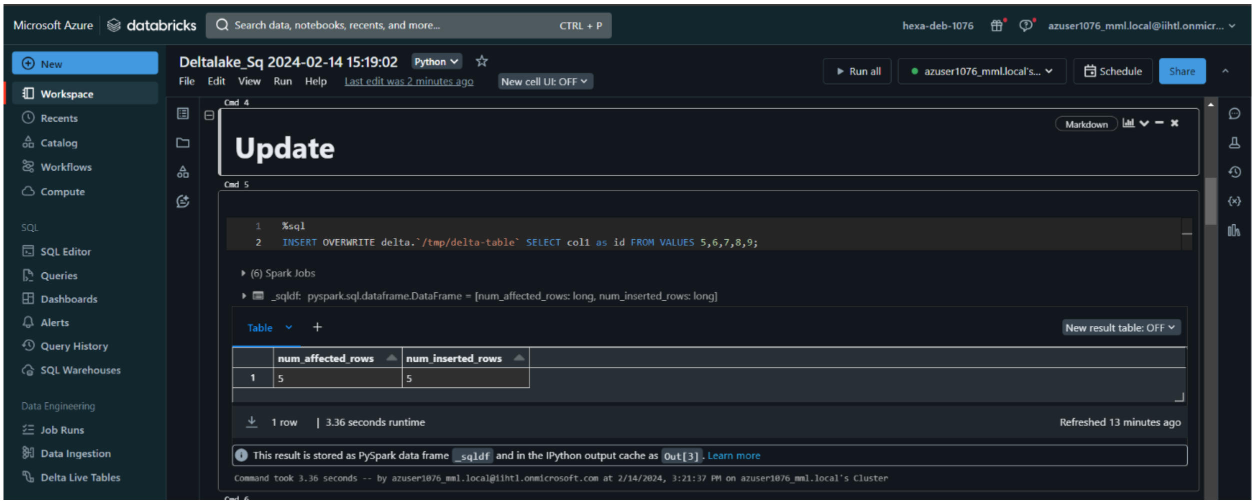
```
%sql
SELECT * FROM delta.`/tmp/delta-table`;
```

```
(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                Refreshed 14 minutes ago
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@iihtl.onmicrosoft.com at 2/14/2024, 3:20:31 PM on azuser1076_mml.local's Cluster
```

Updating data with overwrite:



Update

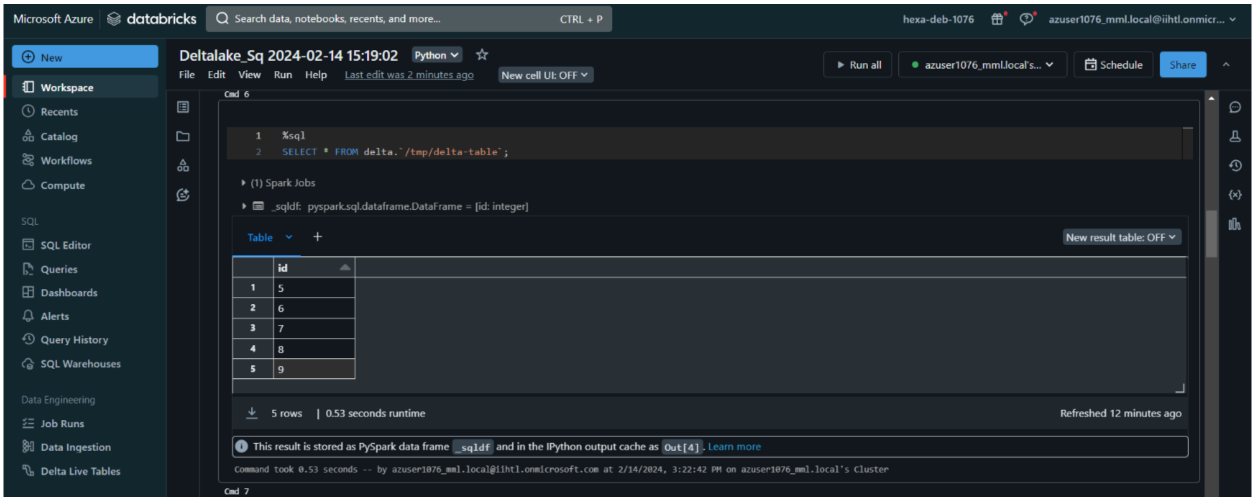
```
%sql
INSERT OVERWRITE delta.`/tmp/delta-table` SELECT col1 as id FROM VALUES 5,6,7,8,9;
```

```
(6) Spark Jobs
_sqldf:  pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]
```

	num_affected_rows	num_inserted_rows
1	5	5

```
1 row  | 3.36 seconds runtime                                   Refreshed 13 minutes ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[3]. Learn more
Command took 3.36 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:21:37 PM on azuser1076_mml.local's Cluster
```

Getting the Updated Data:



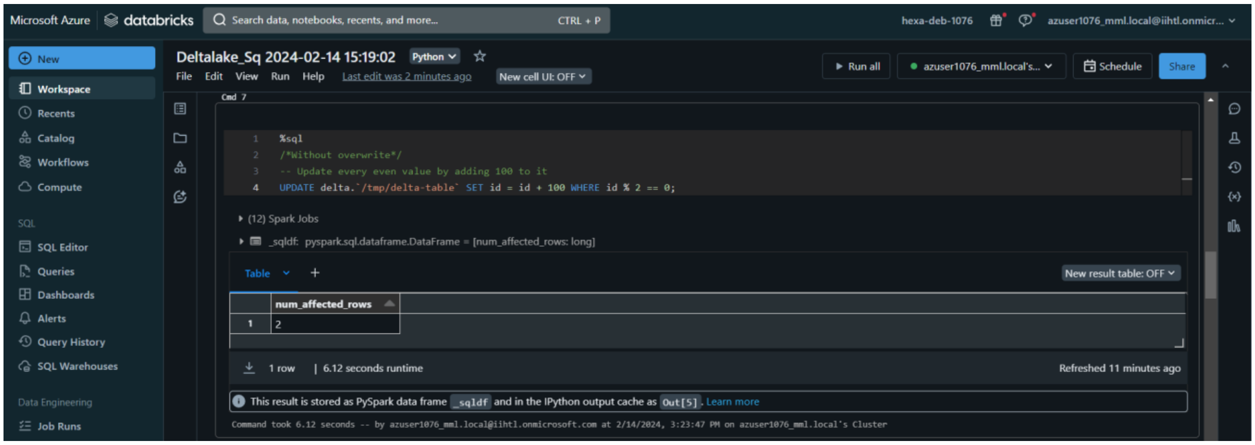
```
%sql
SELECT * FROM delta.`/tmp/delta-table`;
```

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

	id
1	5
2	6
3	7
4	8
5	9

```
5 rows  | 0.53 seconds runtime                                  Refreshed 12 minutes ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[4]. Learn more
Command took 0.53 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:22:42 PM on azuser1076_mml.local's Cluster
```

Updating Data without Overwrite:



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

```
(12) Spark Jobs
_sqldf:  pyspark.sql.dataframe.DataFrame = [num_affected_rows: long]
```

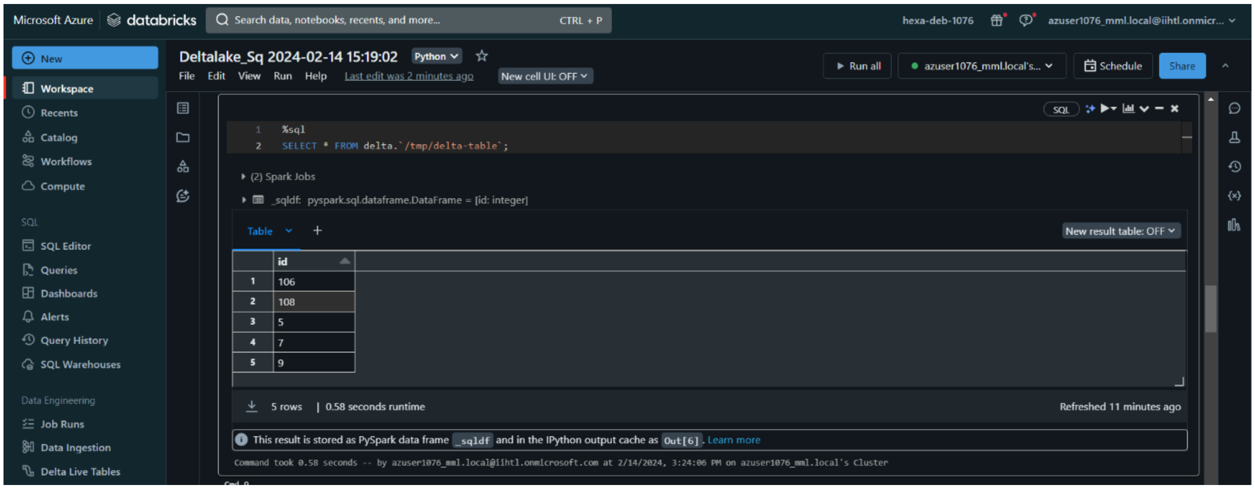
	num_affected_rows
1	2

1 row | 6.12 seconds runtime   Refreshed 11 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[5]`. Learn more

```
Command took 6.12 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:23:47 PM on azuser1076_mml.local's Cluster
```

Viewing the updated data:



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

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

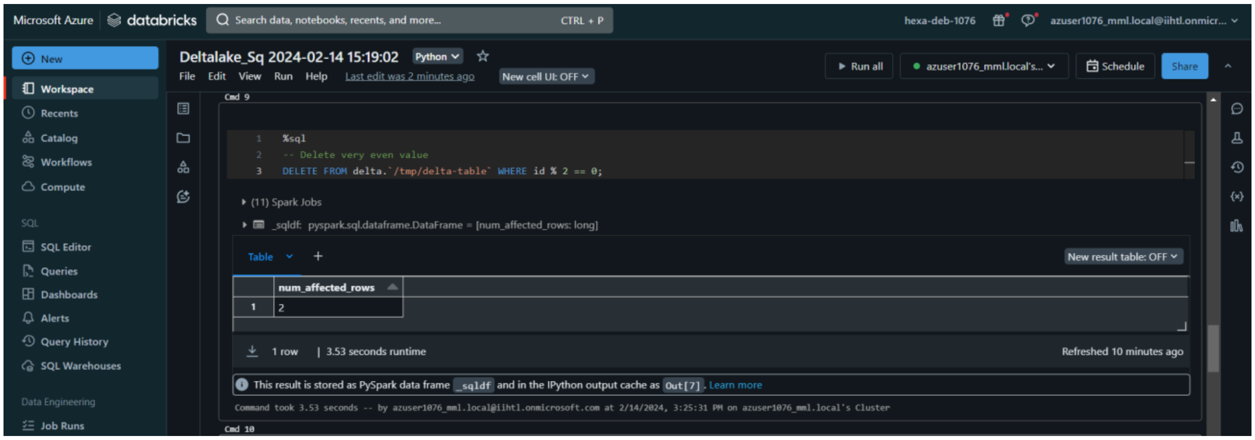
	id
1	106
2	108
3	5
4	7
5	9

5 rows | 0.58 seconds runtime   Refreshed 11 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[6]`. Learn more

```
Command took 0.58 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:24:06 PM on azuser1076_mml.local's Cluster
```

Deleting Even Data:



Deltalake_Sq 2024-02-14 15:19:02

Cmd 9

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

(11) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long]

	num_affected_rows
1	2

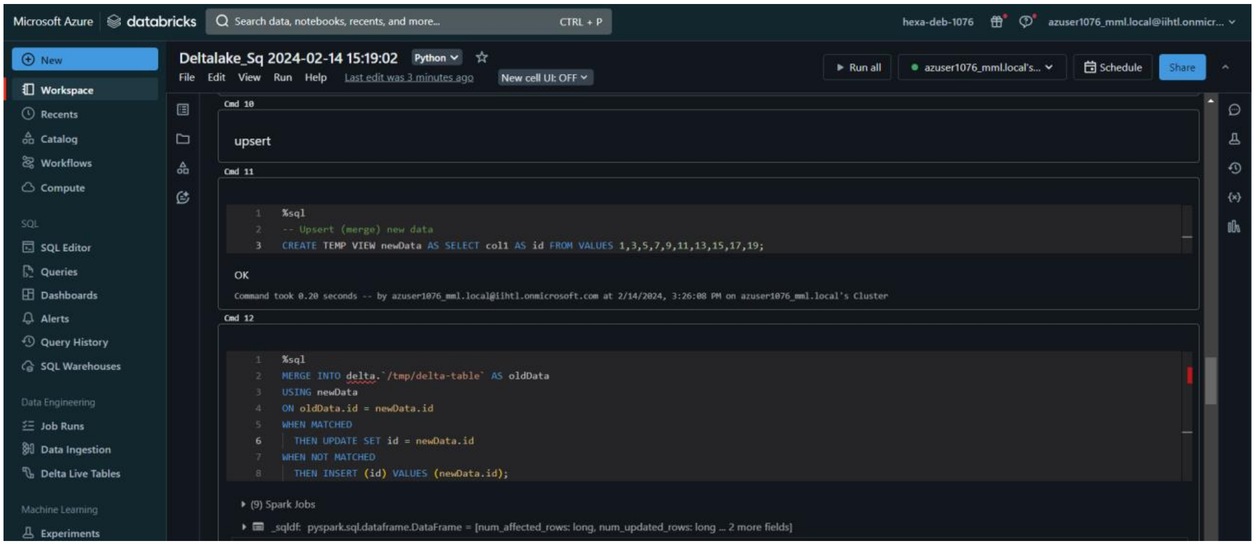
1 row | 3.53 seconds runtime Refreshed 10 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[7]`. Learn more

```
Command took 3.53 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:25:31 PM on azuser1076_mml.local's Cluster
```

Cmd 10

Upsert&Merge operations:



Deltalake_Sq 2024-02-14 15:19:02

Cmd 10

upsert

Cmd 11

```
%sql
-- Upsert (merge) new data
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@iihtl.onmicrosoft.com at 2/14/2024, 3:26:08 PM on azuser1076_mml.local's Cluster
```

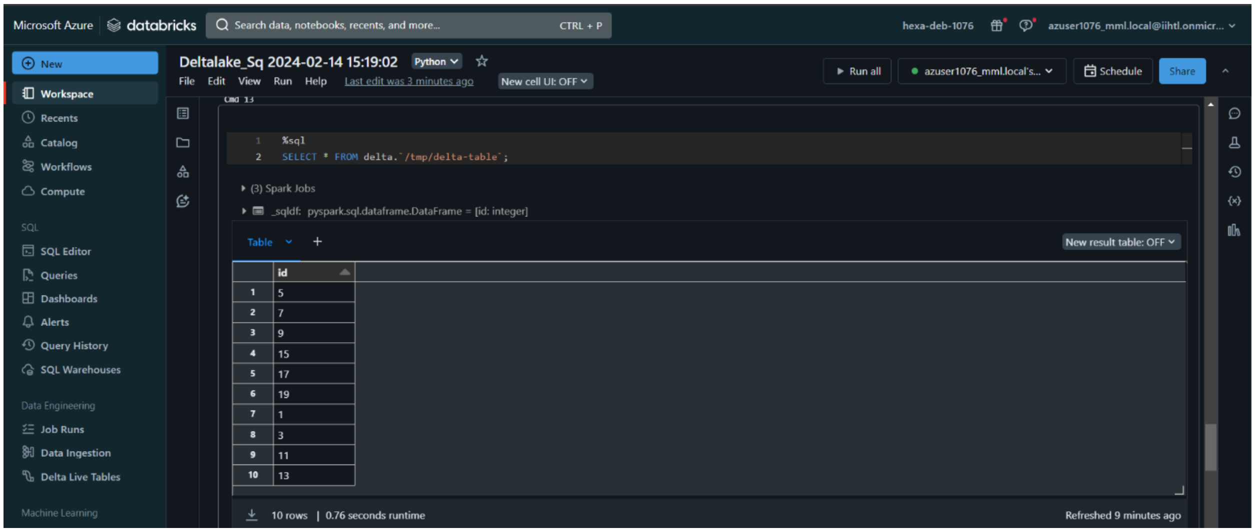
Cmd 12

```
%sql
MERGE INTO delta.`/tmp/delta-table` AS oldData
USING newData
ON oldData.id = newData.id
WHEN MATCHED
  THEN UPDATE SET id = newData.id
WHEN NOT MATCHED
  THEN INSERT (id) VALUES (newData.id);
```

(9) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_updated_rows: long ... 2 more fields]

Viewing Upserted data:



Cmd 13

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

(3) Spark Jobs

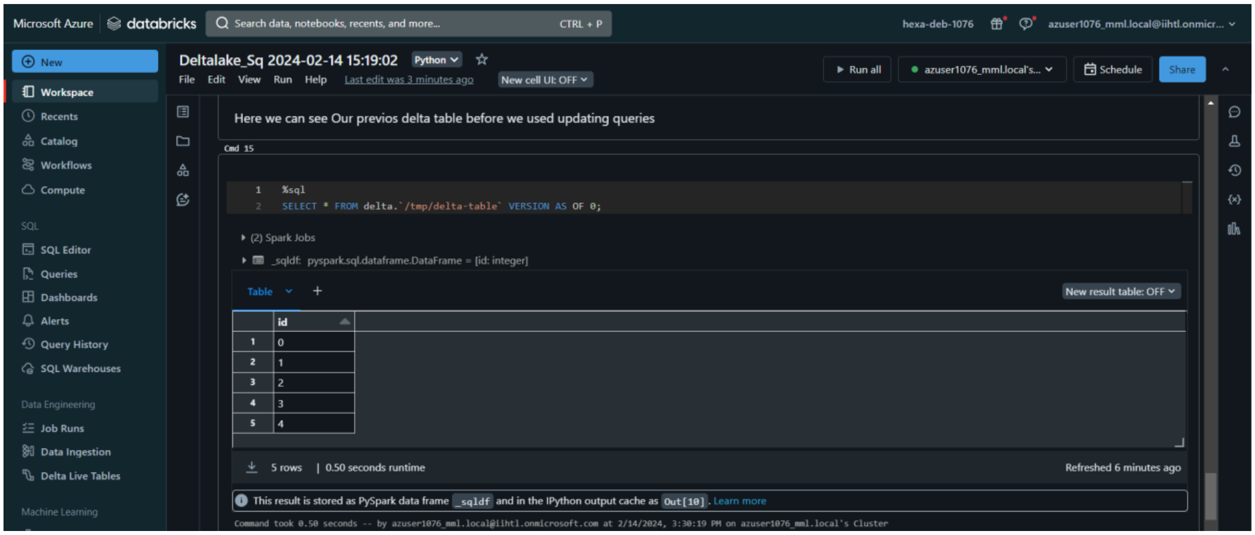
_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]

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

10 rows | 0.76 seconds runtime

Refreshed 9 minutes ago

Getting older data:



Here we can see Our previos delta table before we used updating queries

Cmd 15

```
%sql
SELECT * FROM delta.`/tmp/delta-table` VERSION AS OF 0;
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]

	id
1	0
2	1
3	2
4	3
5	4

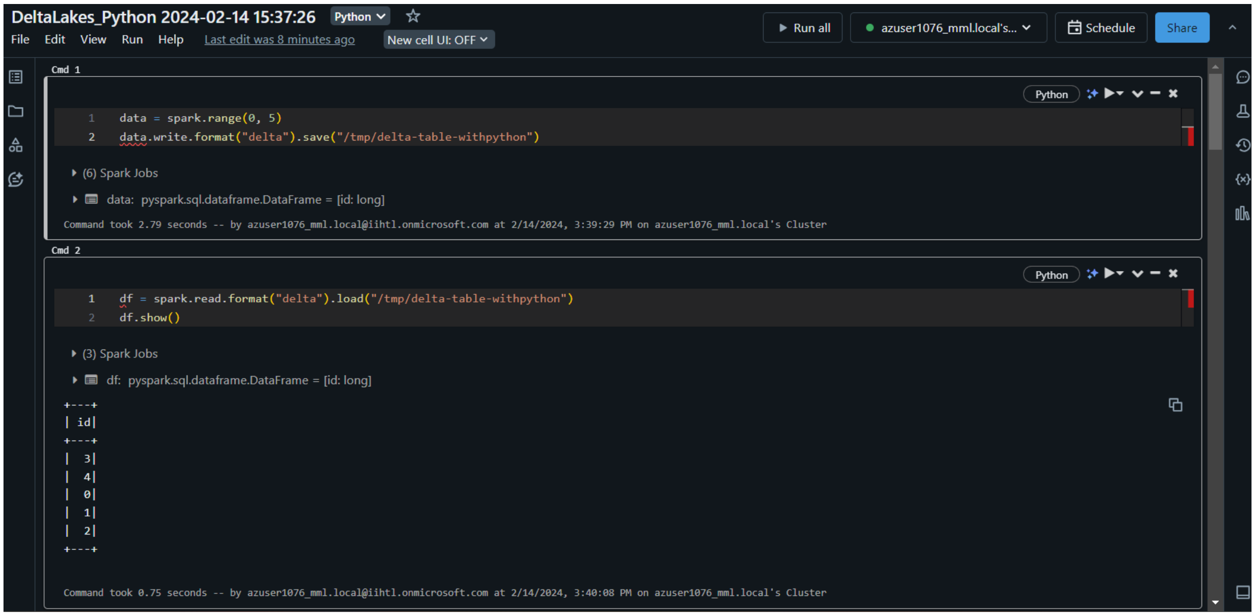
5 rows | 0.50 seconds runtime

Refreshed 6 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[10]`. Learn more

```
Command took 0.50 seconds -- by azuser1076_mml.local@iihtl.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
Python
Cmd 2
Python

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

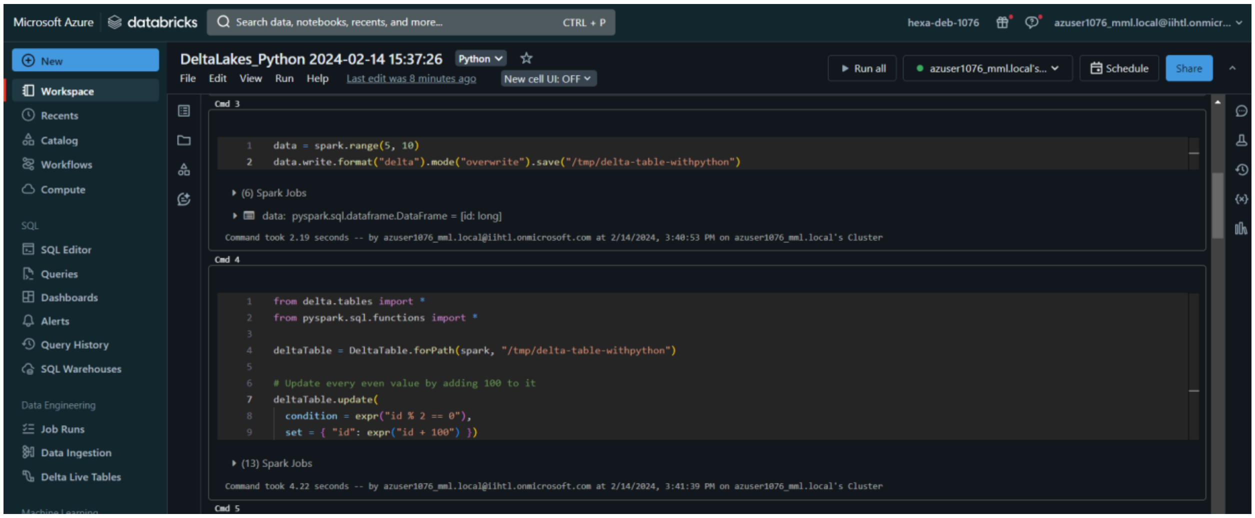
```
(6) Spark Jobs
data: pyspark.sql.dataframe.DataFrame = [id: long]
Command took 2.79 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:39:29 PM on azuser1076_mml.local's Cluster
```

```
df = spark.read.format("delta").load("/tmp/delta-table-withpython")
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@iihtl.onmicrosoft.com at 2/14/2024, 3:40:08 PM on azuser1076_mml.local's Cluster
```

Overwrite&without overwrite:



```
data = spark.range(5, 10)
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@iihtl.onmicrosoft.com at 2/14/2024, 3:40:53 PM on azuser1076_mml.local's Cluster
```

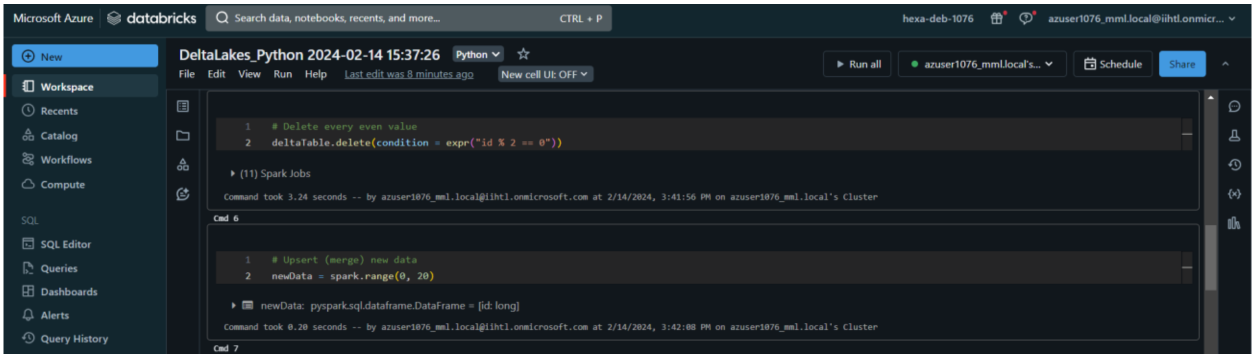
```
from delta.tables import *
from pyspark.sql.functions import *

deltaTable = DeltaTable.forPath(spark, "/tmp/delta-table-withpython")

# Update every even value by adding 100 to it
deltaTable.update(
  condition = expr("id % 2 == 0"),
  set = { "id": expr("id + 100") })
```

```
(13) Spark Jobs
Command took 4.22 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:41:39 PM on azuser1076_mml.local's Cluster
```

Delete , Upsert&retriving new data in Python:



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

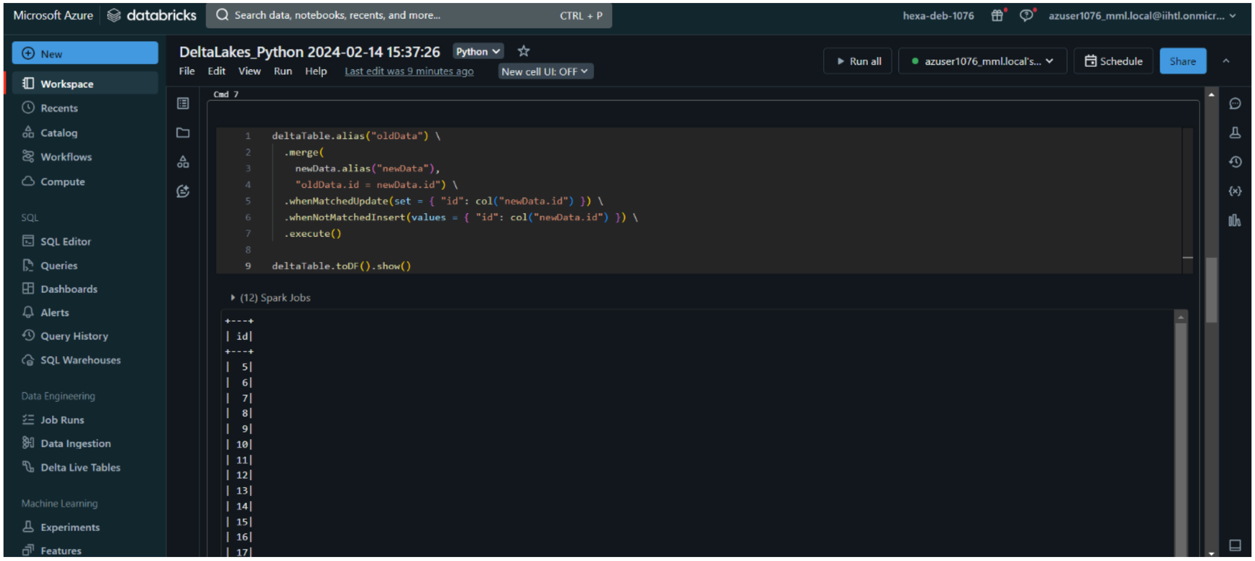
(11) Spark Jobs

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

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

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

```
Command took 0.20 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:42:08 PM on azuser1076_mml.local's Cluster
```



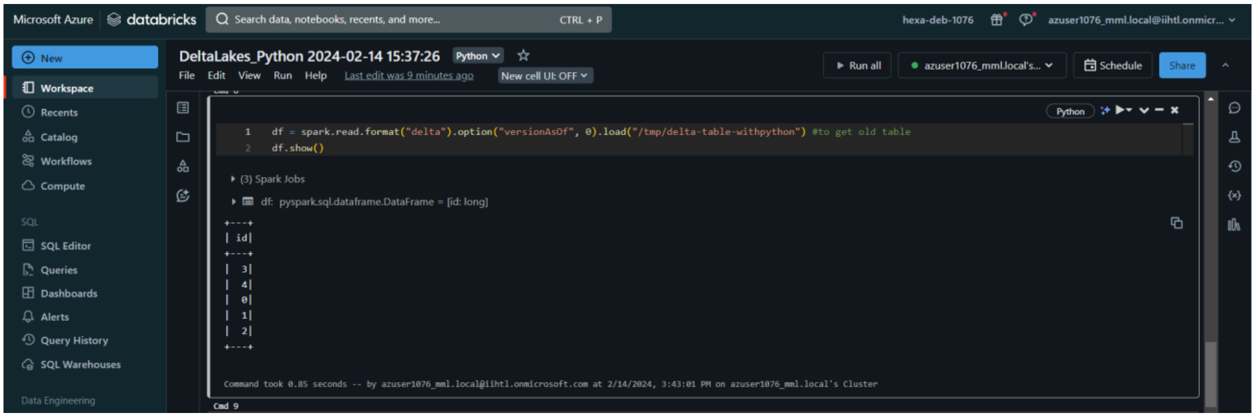
```
deltaTable.alias("oldData") \
  .merge(
    newData.alias("newData"),
    "oldData.id = newData.id") \
  .whenMatchedUpdate(set = { "id": col("newData.id") }) \
  .whenNotMatchedInsert(values = { "id": col("newData.id") }) \
  .execute()

deltaTable.toDF().show()
```

(12) Spark Jobs

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

Getting older data:



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

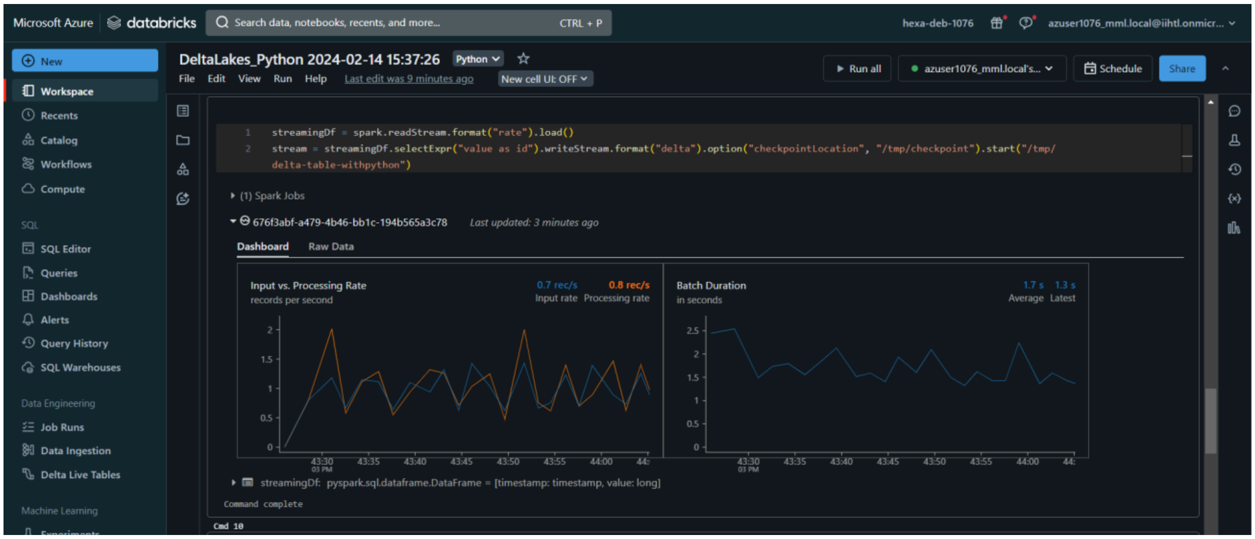
(3) Spark Jobs

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

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

```
Command took 0.85 seconds -- by azuser1076_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:43:01 PM on azuser1076_mml.local's Cluster
```

Streaming Data:



```
streamingDf = spark.readStream.format("rate").load()
stream = streamingDf.selectExpr("value as id").writeStream.format("delta").option("checkpointLocation", "/tmp/checkpoint").start("/tmp/
delta-table-withpython")
```

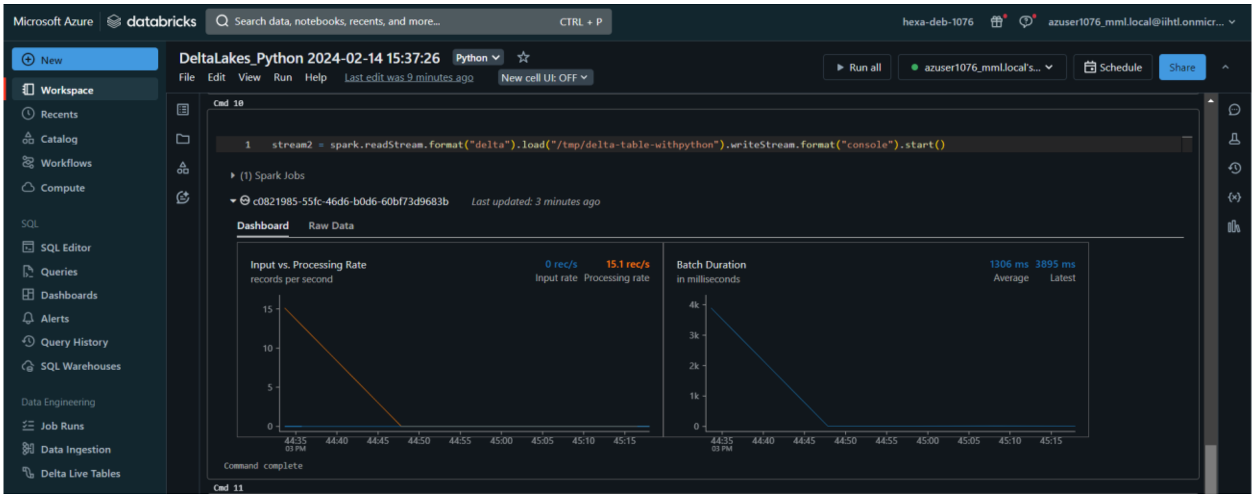
(1) Spark Jobs

676f3abf-a479-4b46-bb1c-194b565a3c78 *Last updated: 3 minutes ago*

streamingDf: pyspark.sql.dataframe.DataFrame = [timestamp: timestamp, value: long]

```
Command complete
```

```
Cmd 10
```



```
Cmd 10
```

```
stream2 = spark.readStream.format("delta").load("/tmp/delta-table-withpython").writeStream.format("console").start()
```

(1) Spark Jobs

c0821985-55fc-46d6-b0d6-60bf73d9683b *Last updated: 3 minutes ago*

```
Command complete
```

```
Cmd 11
```