

NAME : AKULA SHARATH CHANDRA

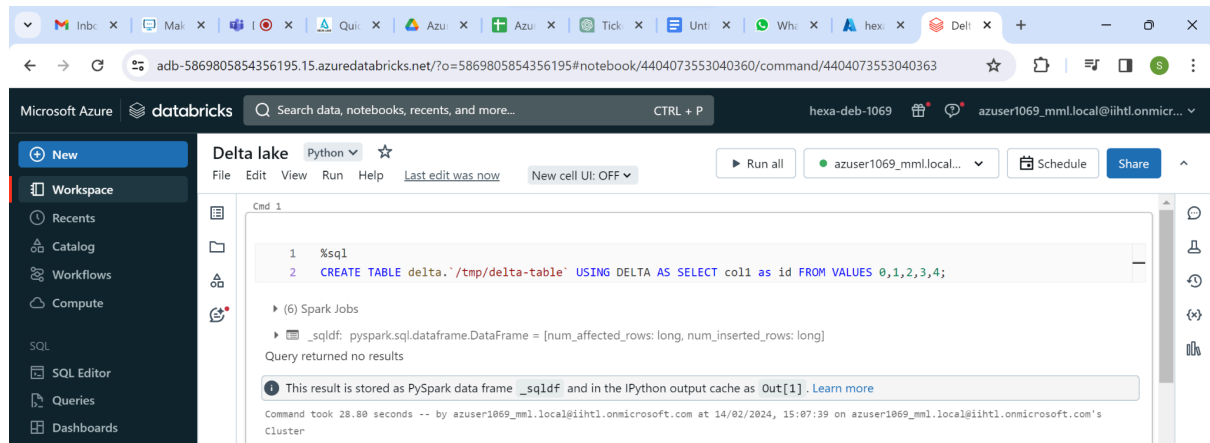
BATCH : DATA ENGINEERING

DATE : 14-02-2024

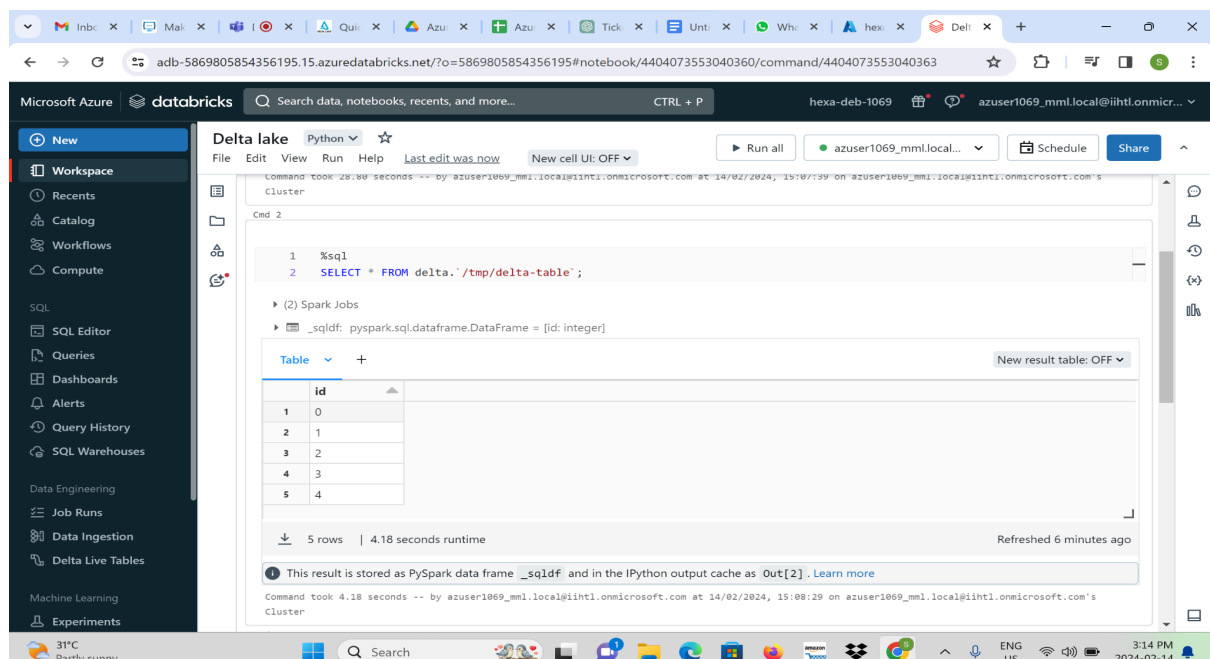
TOPICS :

- 1. Creating a delta table and passing values**
- 2. Creating and visualising the stream data Create a table.**

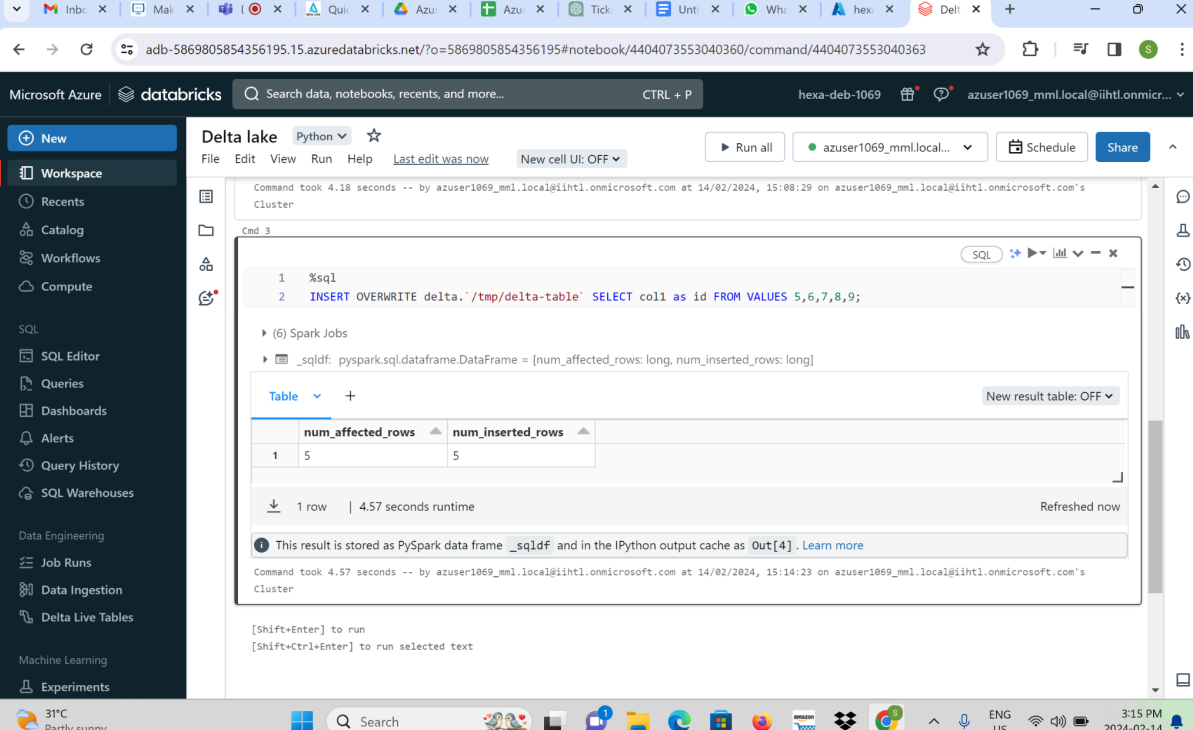
To create a Delta table, write a DataFrame out in the delta format. You can use existing Spark SQL code and change the format from parquet, csv, json, and so on, to delta.



Read data You read data in your Delta table by specifying the path to the files: "/tmp/delta-table":



Update table data Delta Lake supports several operations to modify tables using standard DataFrame APIs. This example runs a batch job to overwrite the data in the table:



The screenshot displays the Databricks Delta Lake interface. The top navigation bar includes the Microsoft Azure logo, the Databricks logo, a search bar, and a user profile dropdown. The left sidebar contains a navigation menu with options like 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 workspace area shows a Delta lake table with a Python notebook editor. The command executed is: `1 %sql`
`2 INSERT OVERWRITE delta.`/tmp/delta-table` SELECT col1 as id FROM VALUES 5,6,7,8,9;` The output shows a table with two columns: `num_affected_rows` and `num_inserted_rows`. The first row shows 5 for both. The command took 4.18 seconds to execute. Below the table, a message states: "This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[4]`. [Learn more](#)". The bottom status bar shows the system temperature as 31°C and the time as 3:15 PM on 2024-02-14.

	num_affected_rows	num_inserted_rows
1	5	5

Conditional update without overwrite Delta Lake provides programmatic APIs to conditional update, delete, and merge (upsert) data into tables. Here are a few examples.

Microsoft Azure databricks Search data, notebooks, recents, and more... CTRL + P hexa-deb-1069 azuser1069_mml.local@iihtl.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 Experiments

Delta lake Python ☆ File Edit View Run Help Last edit was 1 minute a... New cell UI: OFF Interrupt azuser1069_mml.local... Schedule Share

Cmd 4

```
1 %sql
2 -- Update every even value by adding 100 to it
3 UPDATE delta.`/tmp/delta-table` SET id = id + 100 WHERE id % 2 == 0;
```

(13) Spark Jobs

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

num_affected_rows
2

1 row | 9.71 seconds runtime Refreshed 5 minutes ago

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

Command took 9.71 seconds -- by azuser1069_mml.local@iihtl.onmicrosoft.com at 14/02/2024, 15:31:36 on azuser1069_mml.local@iihtl.onmicrosoft.com's Cluster

Cmd 5

```
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;
```

Microsoft Azure databricks Search data, notebooks, recents, and more... CTRL + P hexa-deb-1069 azuser1069_mml.local@iihtl.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 Experiments

Delta lake Python ☆ File Edit View Run Help Last edit was 2 minutes... New cell UI: OFF Interrupt azuser1069_mml.local... Schedule Share

Cmd 5

```
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 0.13 seconds -- by azuser1069_mml.local@iihtl.onmicrosoft.com at 14/02/2024, 15:32:43 on azuser1069_mml.local@iihtl.onmicrosoft.com's Cluster

The screenshot shows the Databricks Delta Lake interface. 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, Machine Learning, and Experiments. The main area displays a SQL query in a cell:

```
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
```

Below the query, the results are shown as a table with 15 Spark Jobs. The first job is expanded, showing the following table:

	num_affected_rows	num_updated_rows	num_deleted_rows	num_inserted_rows
1	10	3	0	7

The table is refreshed 4 minutes ago. Below the table, a message states: "This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[6]. Learn more".

You can query previous snapshots of your Delta table by using time travel. If you want to access the data that you overwrote, you can query a snapshot of the table before you overwrote the first set of data using the versionAsOf option.

The screenshot shows the Databricks Delta Lake interface. 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, Machine Learning, and Experiments. The main area displays a SQL query in a cell:

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

Below the query, the results are shown as a table with 2 Spark Jobs. The first job is expanded, showing the following table:

	id
1	106
2	108
3	1
4	3
5	5
6	7
7	9

The table is refreshed 4 minutes ago. Below the table, a message states: "This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[7]. Learn more".

Write a stream of data to a table You can also write to a Delta table using Structured Streaming. The Delta Lake transaction log guarantees exactly-once processing, even when there are other streams or batch queries running concurrently against the table. By default, streams run in append mode, which adds new records to the table:

The screenshot shows a Databricks notebook interface. The top navigation bar includes the Databricks logo, a search bar, and user information. The left sidebar contains navigation links for 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 two code cells. Cell 8, titled 'Delta lake', contains Python code to write a stream to a Delta table. It uses `spark.readStream` to load data from a source named 'rate', selects the 'value' column as 'id', and writes it to a Delta table named 'delta' in the '/tmp/' directory. The code also sets the 'checkpointLocation' to '/tmp/delta-table'. Below the code, a message indicates the stream has stopped with an exception: `org.apache.spark.sql.AnalysisException: Failed to merge fields 'id' and 'id'. Failed to merge incompatible data types IntegerType and LongType`. Cell 9 contains Python code to read a stream of changes from the 'delta' table and write it to a console. The bottom status bar shows the temperature as 31°C, the time as 3:38 PM on 2024-02-14, and the user as azuser1069_mml.local@iihtl.onmicr...

```
1 #Write a stream of data to a table
2 streamingDf = spark.readStream.format("rate").load()
3 stream = streamingDf.selectExpr("value as id").writeStream.format("delta").option("checkpointLocation", "/tmp/
checkpoint").start("/tmp/delta-table")

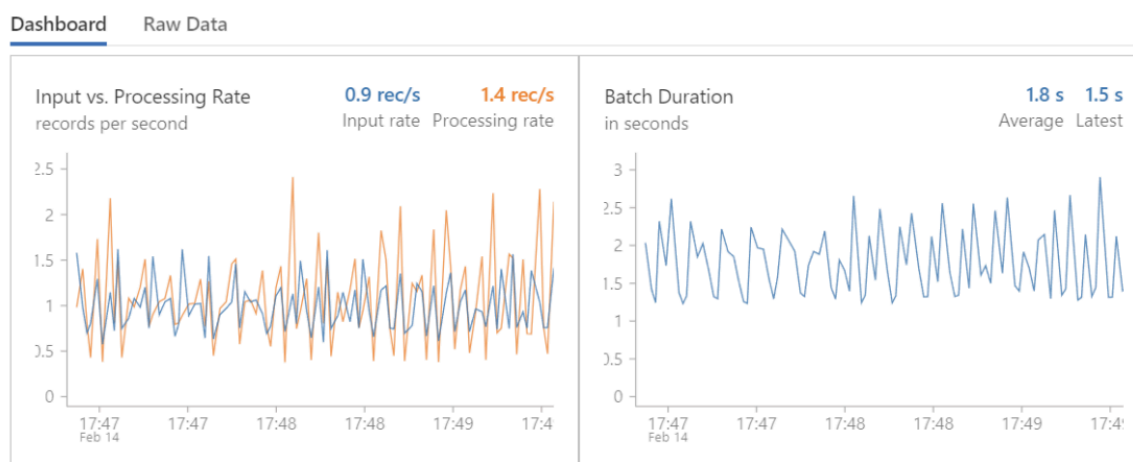
Stream stopped...
org.apache.spark.sql.AnalysisException: Failed to merge fields 'id' and 'id'. Failed to merge incompatible data types IntegerT
ype and LongType

streamingDf: pyspark.sql.dataframe.DataFrame
  timestamp: timestamp
  value: long

Command complete
```

```
1 #Read a stream of changes from a table
2 stream2 = spark.readStream.format("delta").load("/tmp/delta-table").writeStream.format("console").start()

Cancel ***
(1) Spark Jobs
762bfda8-785e-4e9f-a5c4-5df29bcb8e1d Last updated: 5 seconds ago
```



Read a stream of changes from a table While the stream is writing to the Delta table, you can also read from that table as streaming source. For example, you can start another streaming query that prints all the changes made to the Delta table. You can specify which version Structured Streaming should start from by providing the `startingVersion` or `startingTimestamp` option to get changes from that point onwards. See Structured Streaming for details.

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

▼  f0345a3d-3e3d-4d14-8ddf-a387dab47765 Last updated: 15 hours ago

Dashboard Raw Data

