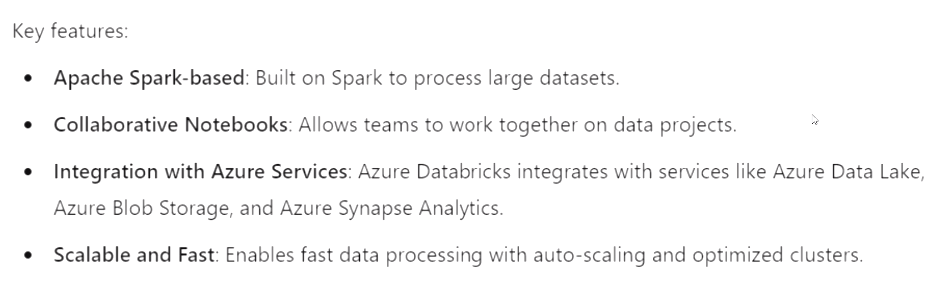
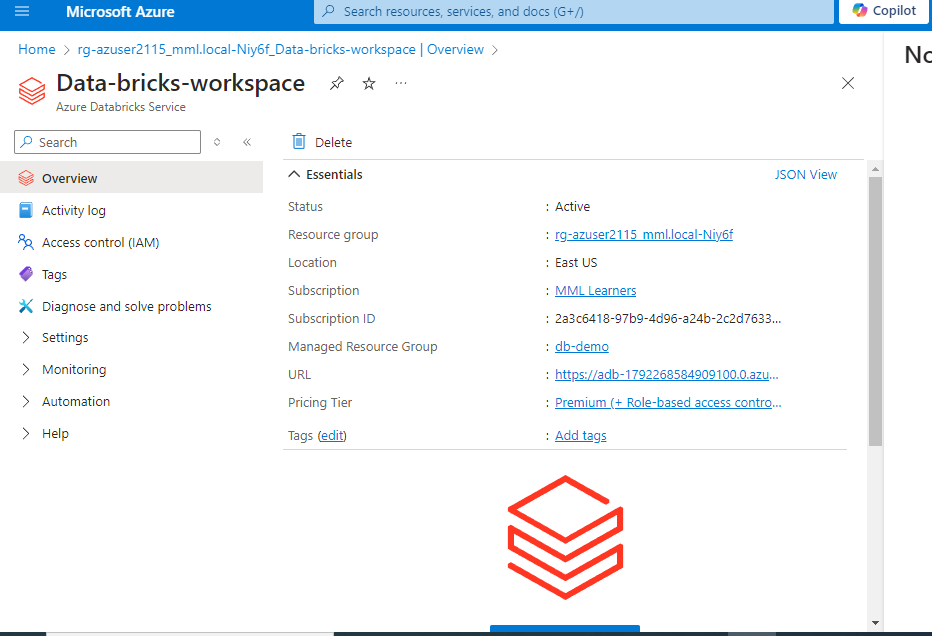
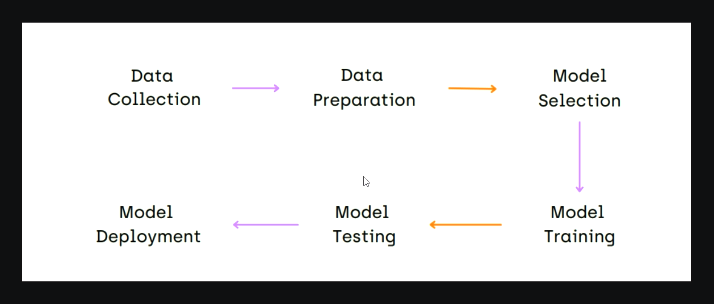
**DATABRICKS**

****

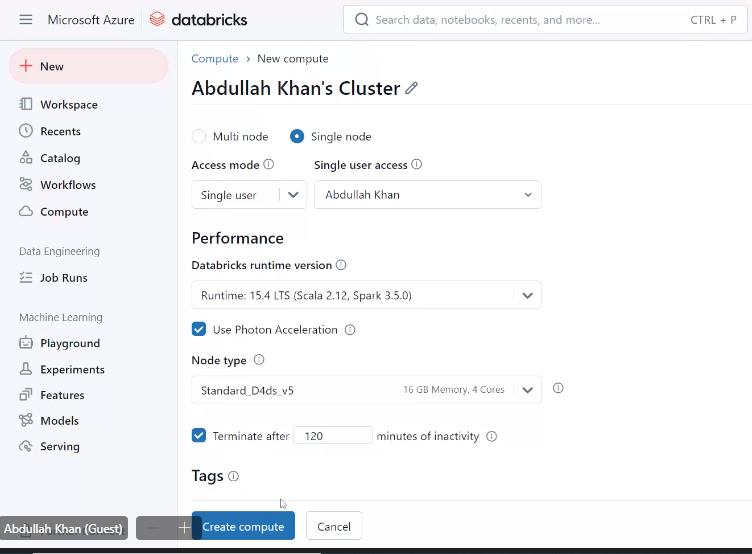
* **Cloud based data engineering tool**

****

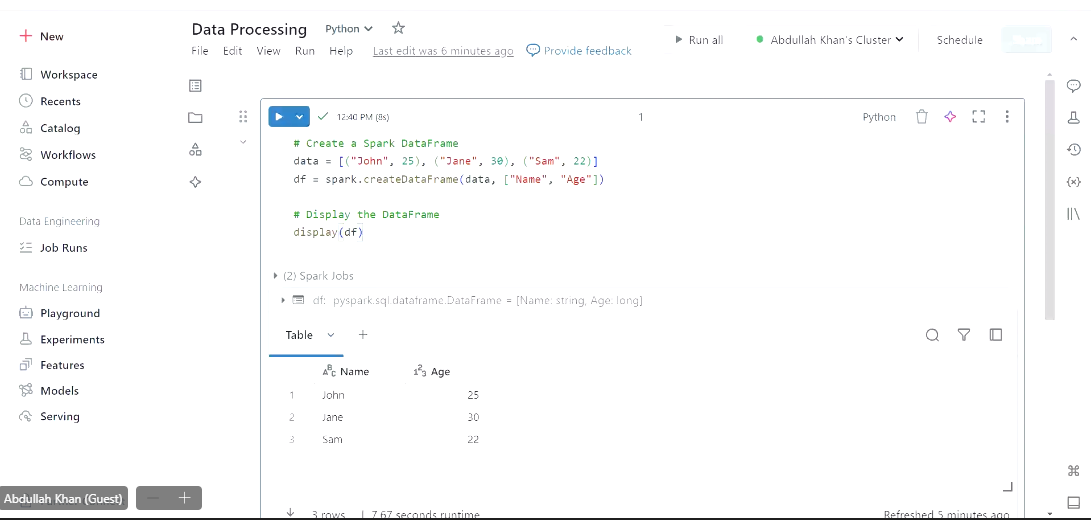
* Search for azure databricks in search bar
* Click on create resources
* Give name for resource group managed resource group and create resoursce



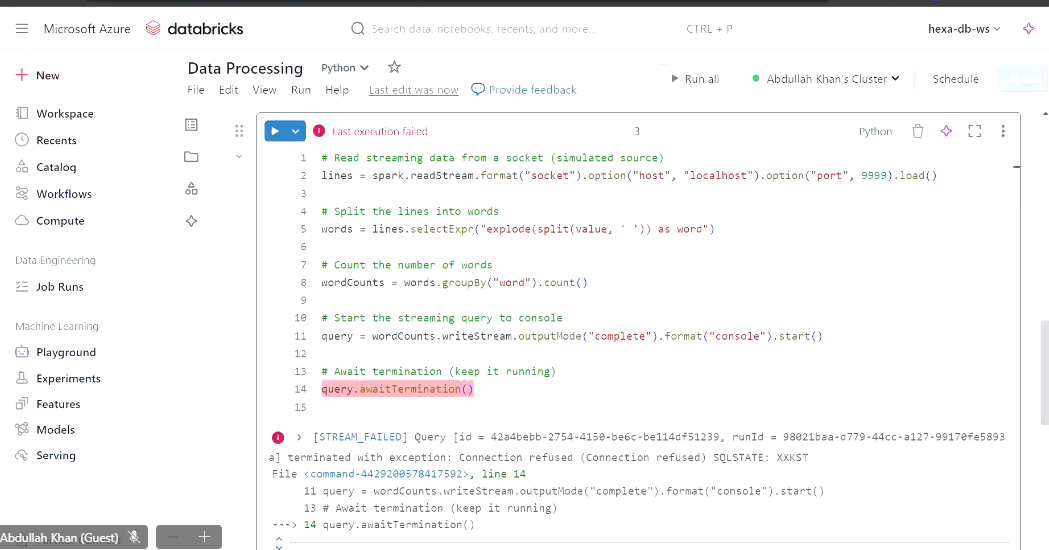
* Data bricks is about getting the data ready to feed for machine learning



* Click on launch Workspace to enter
* Click compute and create compute

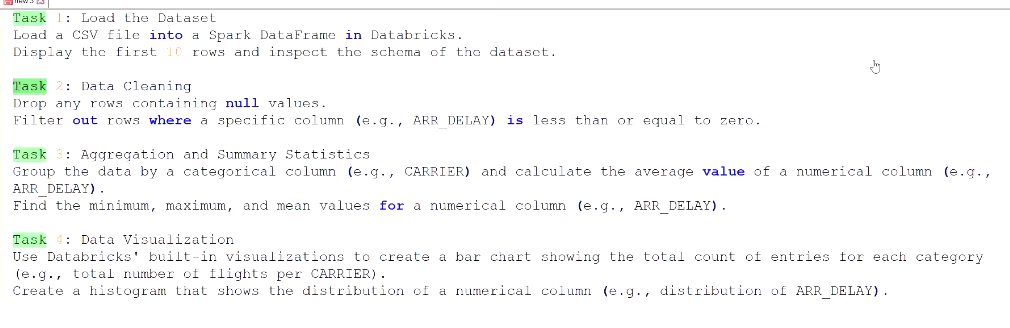


* Notebook to work pyspark codes ^



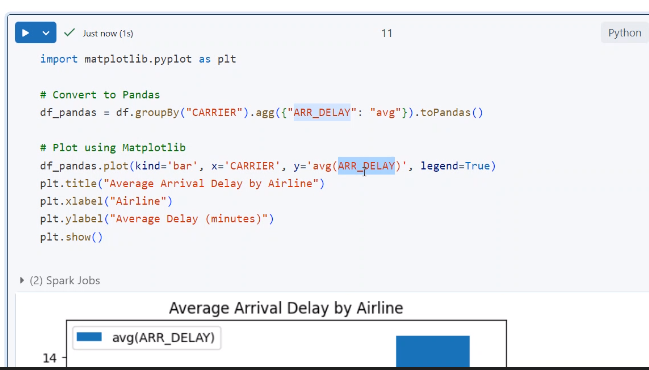
* In the above “socket” is Kafka ,it gives the live stream of data

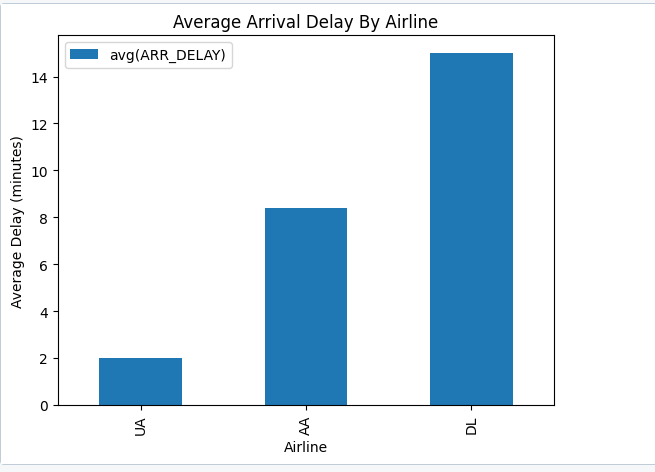
Exercise:



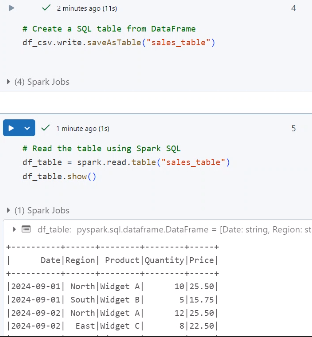
Answer : <Databricks_exercise_sep_11.ipynb>

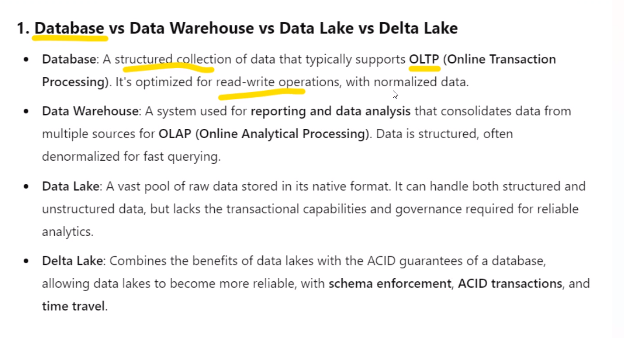
**DATA VISUALIZING USING MATPLOT**



****

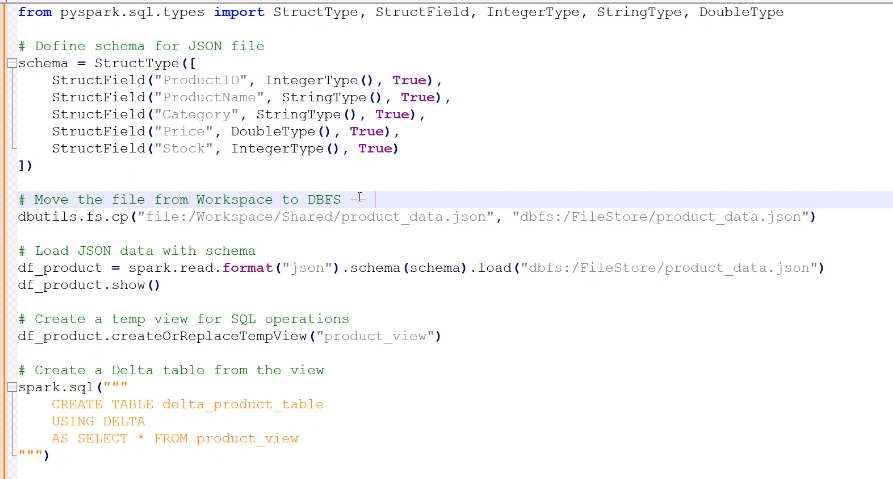
**CONVERTING CSV DATA TO TABLE(SQL) AND READING IT:**

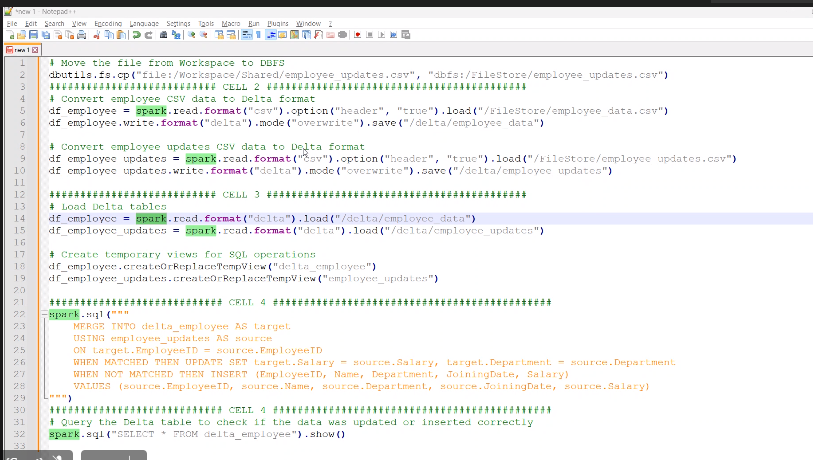
****

****

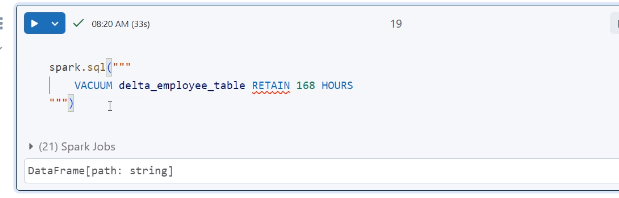
**Example – json to delta :**

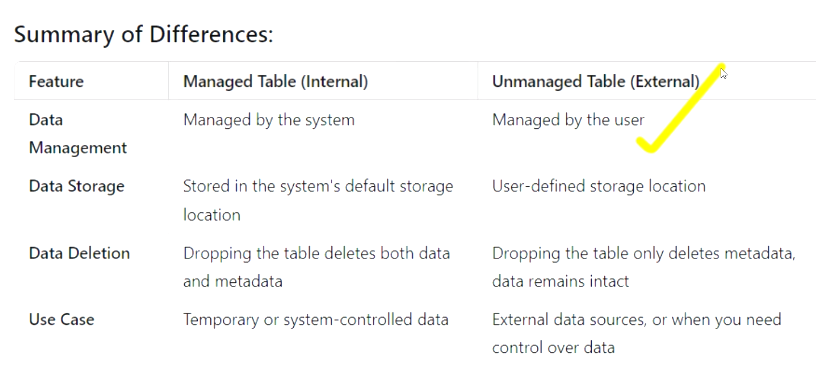
**Delata example:**

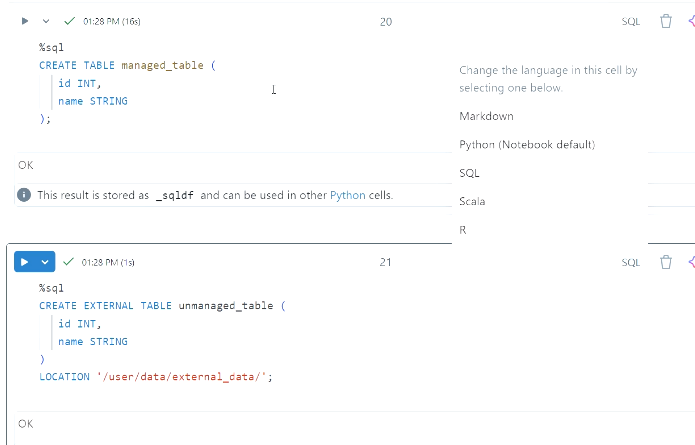
****

****

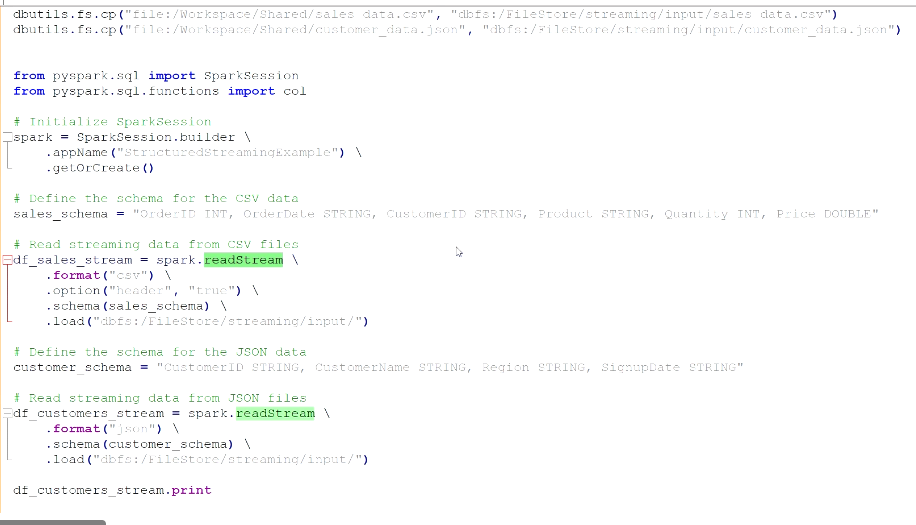
**To delete older records:**

****

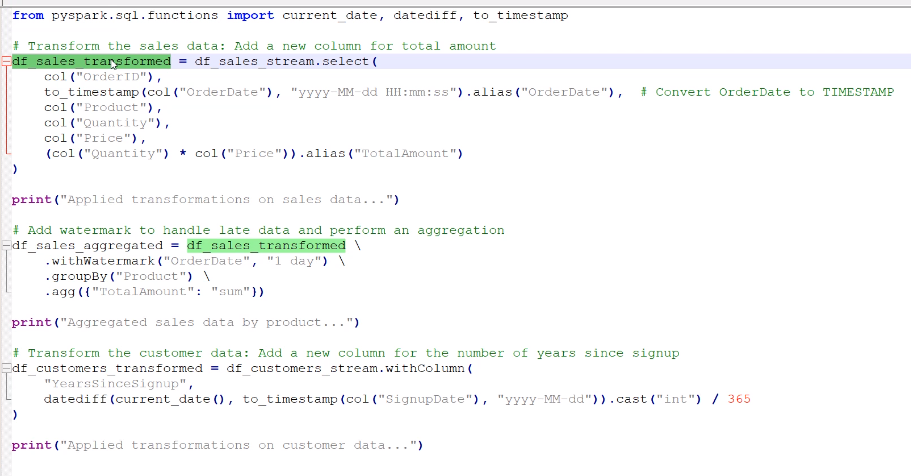
****

****

**READ STREAM : READIMG STREAMING DATA:**

****

**TRANSFORM : TRANSFORMING THE STREAMING DATA**

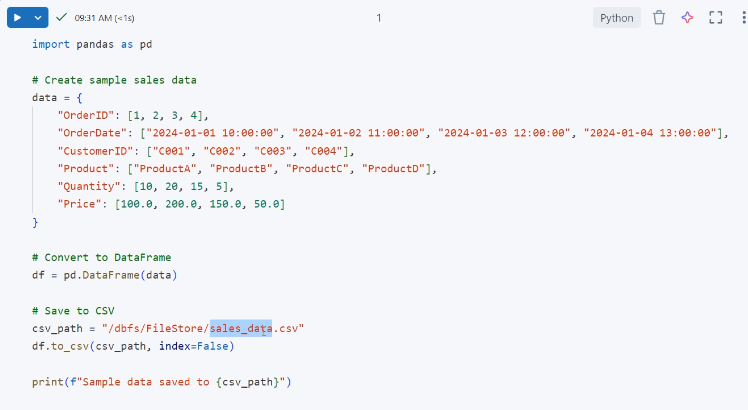
****

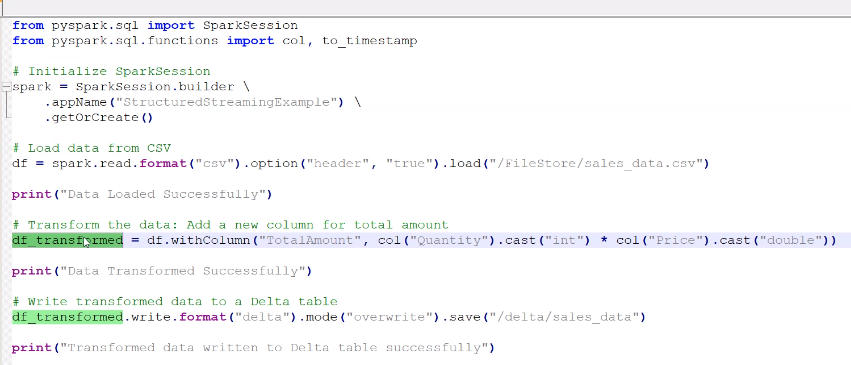
**WRITE STREAM : WRITING THE STREAMING DATA**

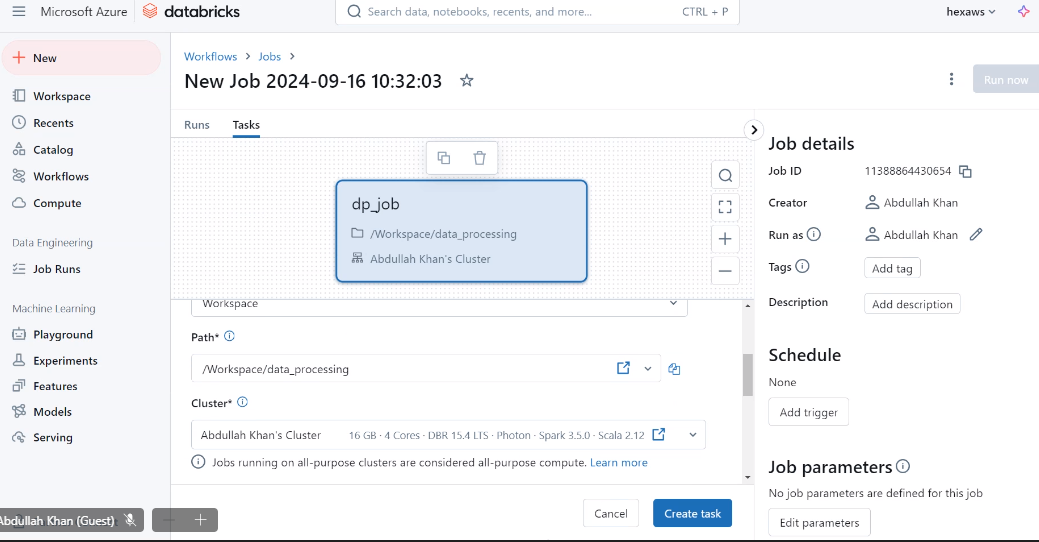
****

**JOB:**

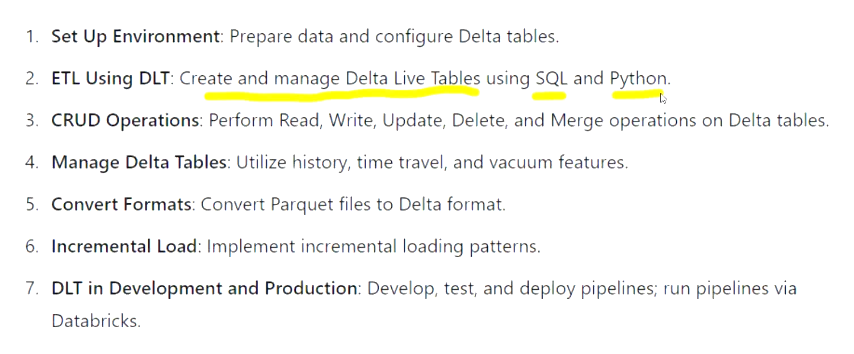
* It will be triggered at particular time.







* Workflows -> jobs -> paste path and choose cluster

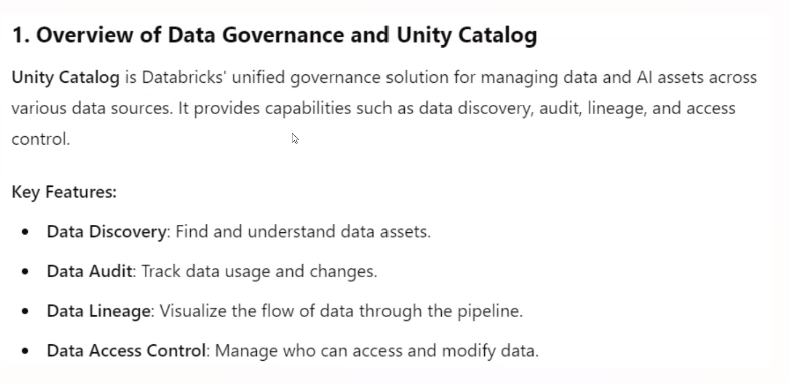


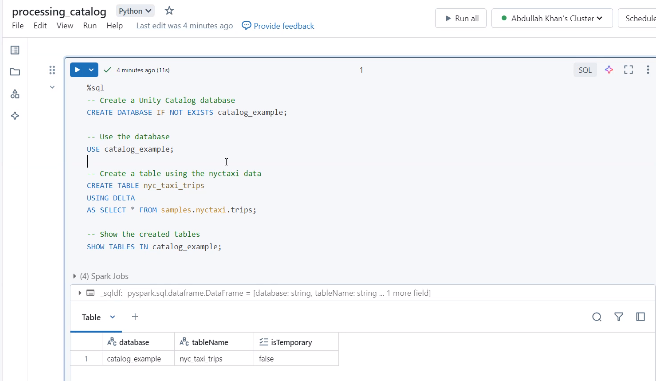




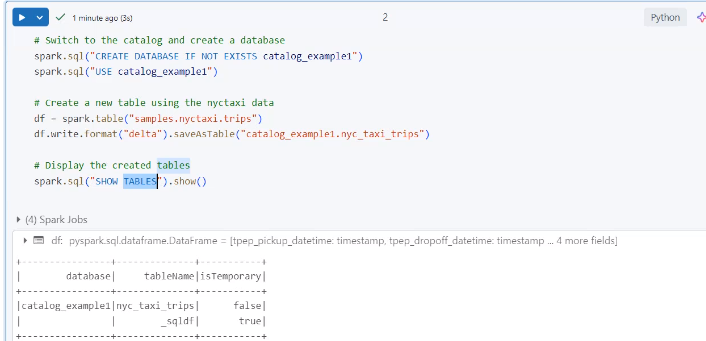
****

**UNITY CATALOG:**

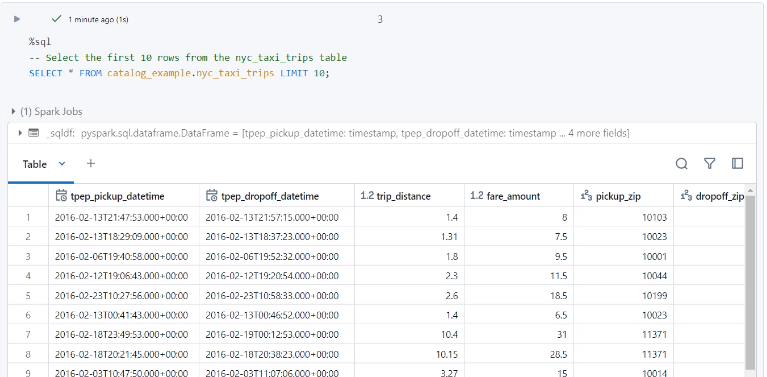
****



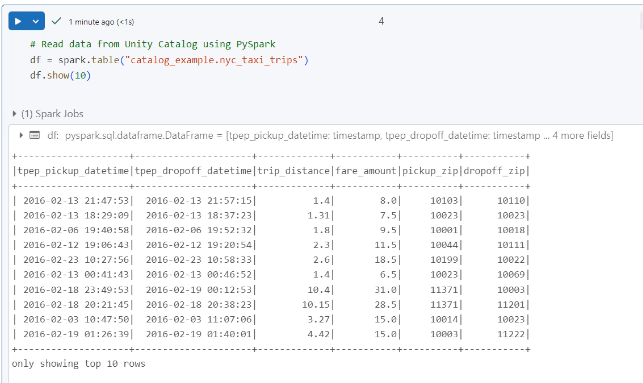
Using spark :



Sql version of selecting data in unity catalog:



Spark version of selecting data in unity catalog:





Overview of the Three-Level Namespace in Unity Catalog

Databricks Unity Catalog organizes data objects into a three-level namespace that

enables fine-grained access control and management of data objects. The three levels

are:

1. Catalog: The top-level container, which contains schemas (databases) and

tables. Each catalog represents an organizational boundary and is often used to

segregate data at the business level.

2. Schema: The second level, which contains tables and views. Schemas (also called

databases) group logically related tables and views.

3. Table/View: The lowest level of the hierarchy, representing the actual data

stored in a table or a view that refers to data.

Example:

CatalogName.SchemaName.TableName

Unity Catalog Structure Example

Catalog: financial\_data

Schema: transactions

Table: credit\_card\_payments

Table: bank\_transfers

Schema: accounts

Table: customer\_accounts

View: active\_customers\_view

Complete Example:

financial\_data.transactions.credit\_card\_payments

Creating Unity Catalog Objects in Databricks

Step 1: Create a Unity Catalog Metastore

Before creating any objects, you need to create a metastore that holds catalogs and

schemas.

# Step 1: Create a new Metastore using SQL

CREATE METASTORE my\_metastore;

1. Create Metastore via Databricks CLI or Admin Console:

In Databricks Admin Console, go to the Metastore tab.

Create a new metastore and assign it to your workspace.

Step 2: Create a Catalog

Once the metastore is set up, you can create catalogs.

-- SQL to create a Catalog

CREATE CATALOG financial\_data;

Step 3: Create a Schema (within a Catalog)

After creating a catalog, you can create schemas to organize your tables and views.

-- SQL to create a Schema

CREATE SCHEMA financial\_data.transactions;

CREATE SCHEMA financial\_data.accounts;

Step 4: Create Tables in Unity Catalog

Create tables within the schema.

-- SQL to create Tables in a Schema

CREATE TABLE financial\_data.transactions.credit\_card\_payments (

PaymentID INT,

Amount DECIMAL(10,2),

PaymentDate DATE

);

CREATE TABLE financial\_data.transactions.bank\_transfers (

TransferID INT,

Amount DECIMAL(10,2),

TransferDate DATE

);

Step 5: Create Views in Unity Catalog

You can also create views to display subsets of data in tables.

-- SQL to create a View

CREATE VIEW financial\_data.accounts.active\_customers\_view AS

SELECT CustomerID, AccountID, Status

FROM financial\_data.accounts.customer\_accounts

WHERE Status = 'Active';

Key Concepts

1. Catalog: Represents the highest level in the namespace and contains multiple

schemas. It segregates data for different organizational units.

2. Schema: Represents the second level of the namespace, containing multiple

tables and views. Schemas group tables and views that belong to the same

logical domain.

3. Table: The third level in the namespace, representing the actual dataset where

data resides. Tables can be partitioned, and data is stored in Delta Lake

format, Parquet, or other supported formats.

4. View: A virtual table generated from a SQL query. It does not physically store

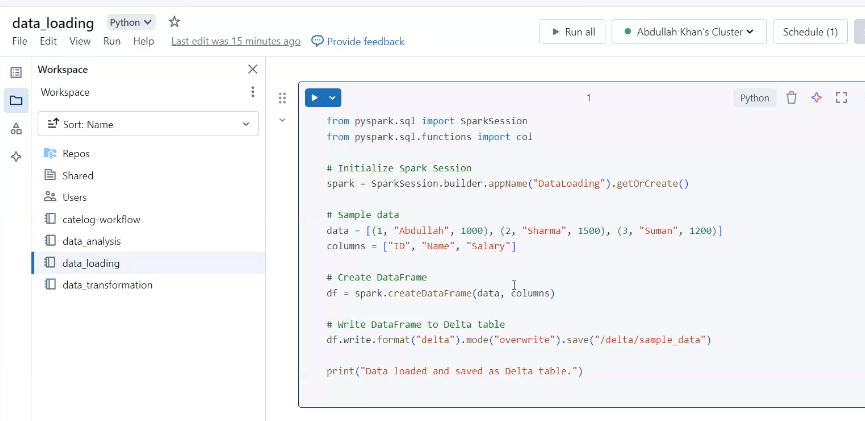
data but provides a way to query and organize subsets of data from tables.

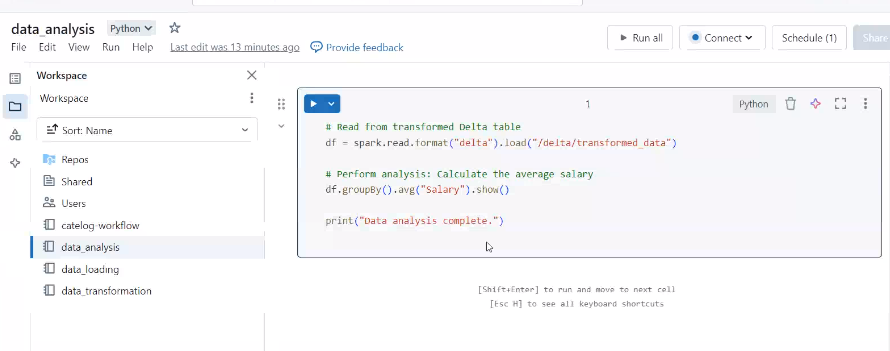
[Three Level Namespace.pdf](Three%20Level%20Namespace.pdf)

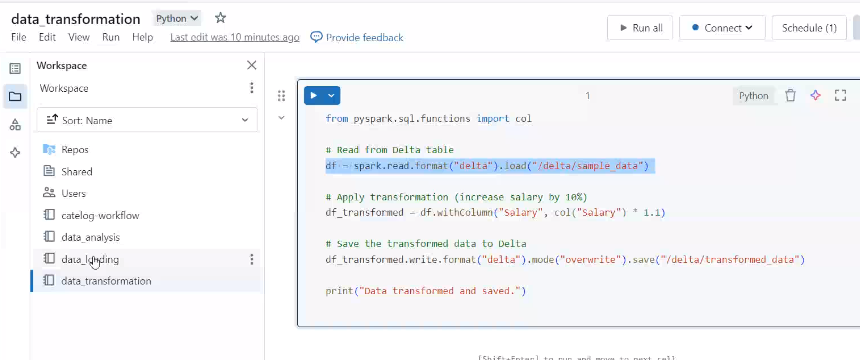
MINI PROJECT USING UNITY CATALOG:

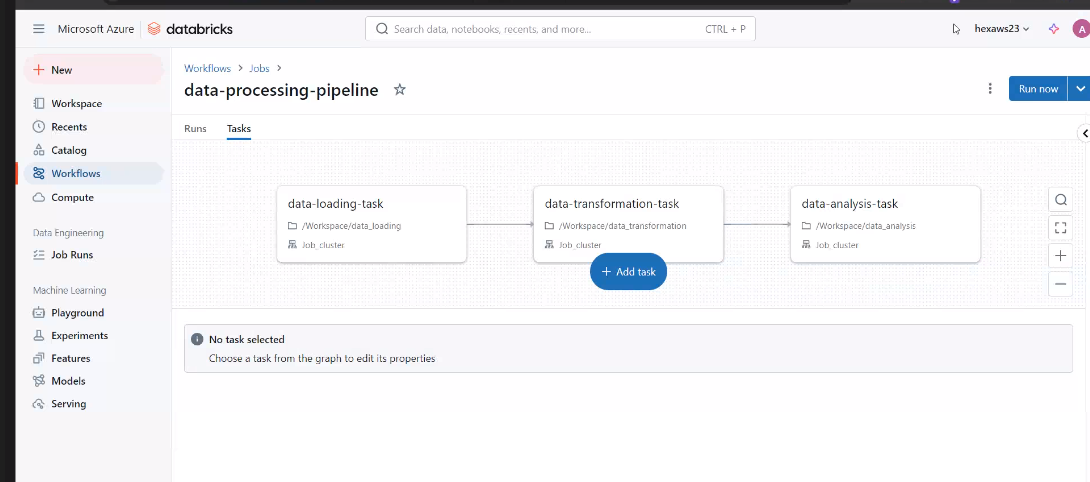
[Unity Catelog Workflow.pdf](Unity%20Catelog%20Workflow.pdf)

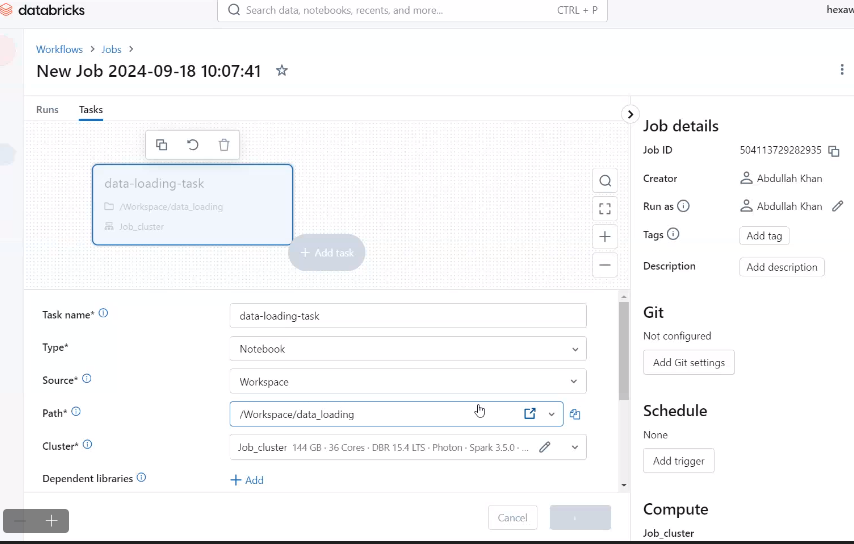
PIPELINE:

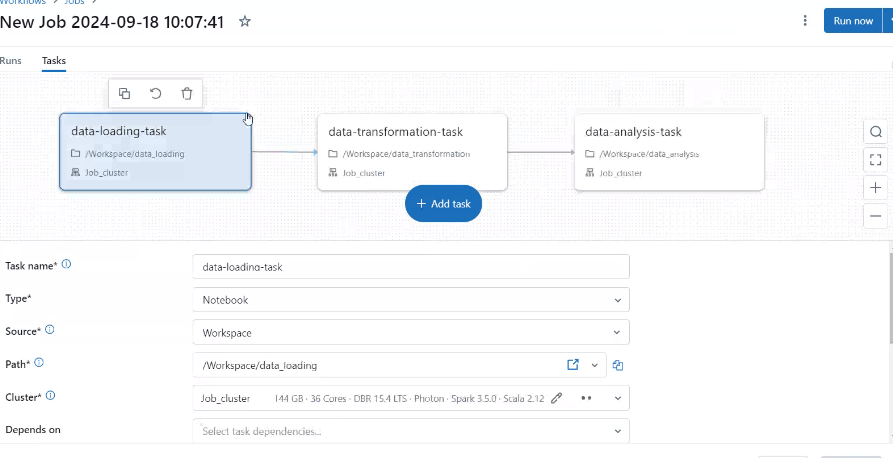










****

* THE ABOVE IS A JOB PIPELINE.
* Job can have multiple tasks.
* The above three notebooks are dependent on other notebooks.
* It will run all 3 notebooks in one go.

HANdLIG MISSING dATA:

