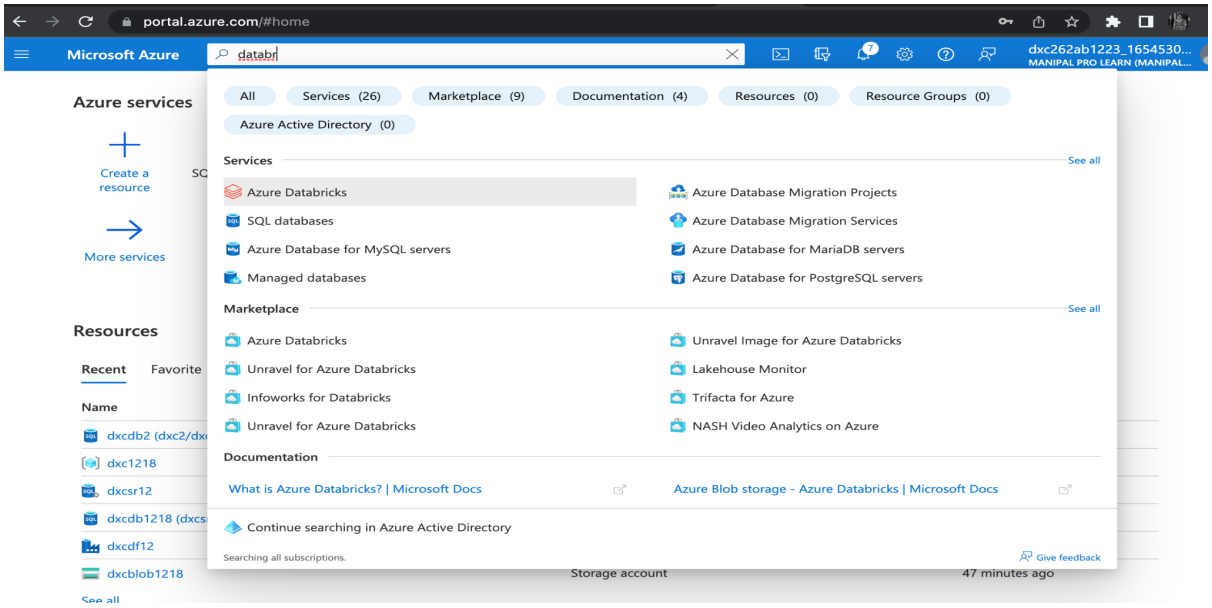
ASSIGNMENT 10

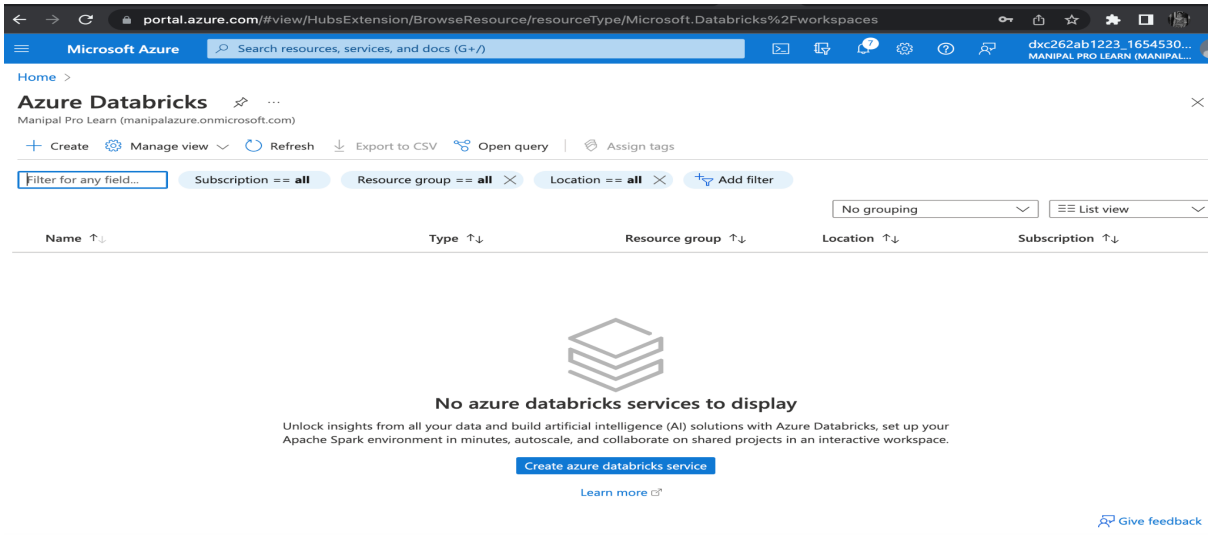
NAME – JYOTHI CHANDANA VOLETI
ROLL NUMBER – DXC-262-AB-1218
BATCH – DXC-262-ANALYTICS-B12-AZURE
COMPANY – DXC TECHNOLOGY
EMPLOYEE DOMAIN –AZURE ANALYTICS
TRAINING UNDER – MANIPAL PRO LEARN
TRAINER NAME – MR. AJAY KUMAR
DATE OF SUBMISSION – 15TH JUNE 2022
NO.OF QUESTIONS: 6

INTRODUCTION: To complete today's task we need to create Azure Databricks workspace. To create that we need to follow the steps below.

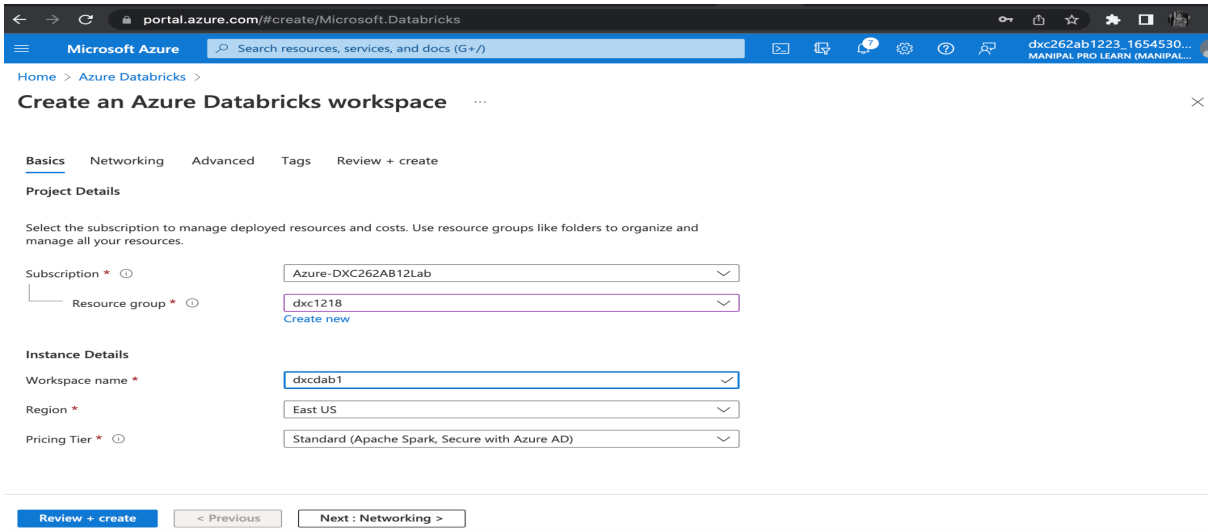
Step 1: Login to Azure portal and search for Databricks.



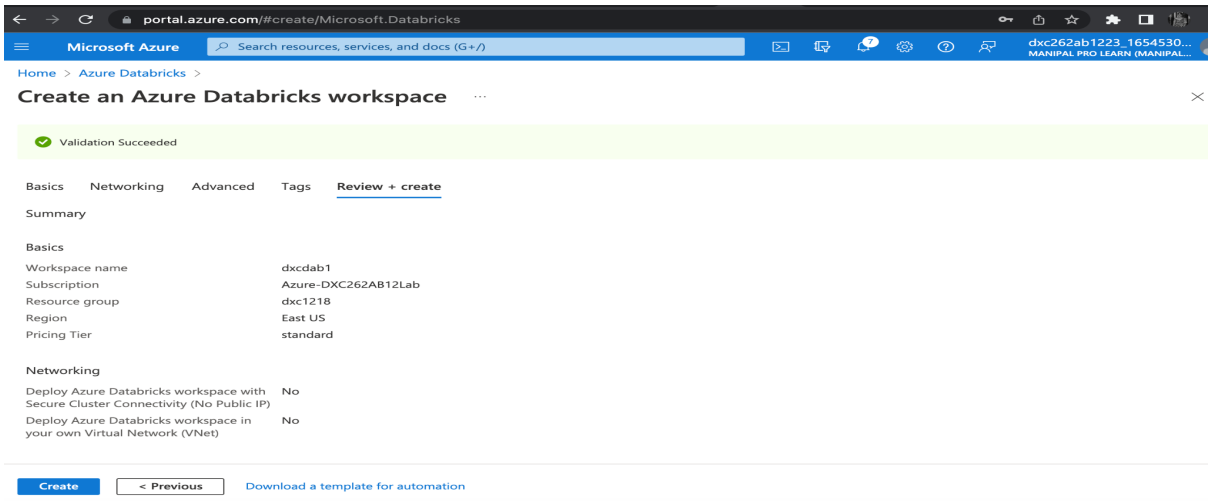
Step 2: Click on Azure Databricks and click on "+create".



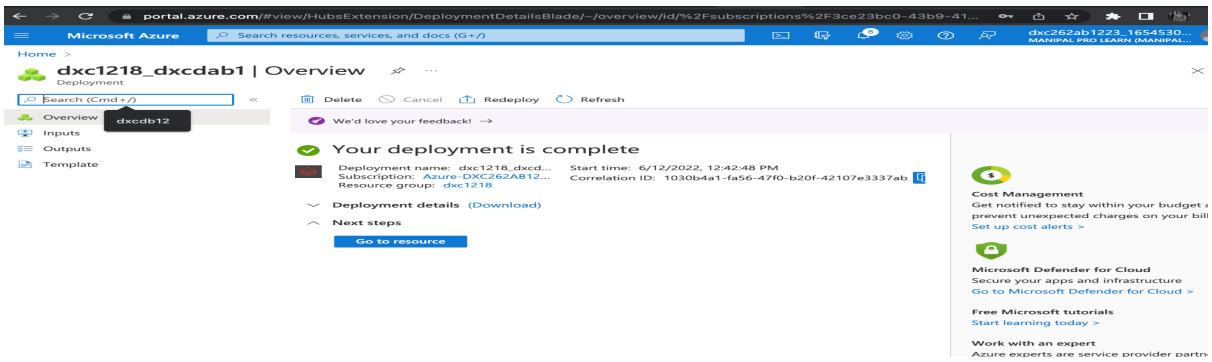
Step 3: Fill all the required fields.



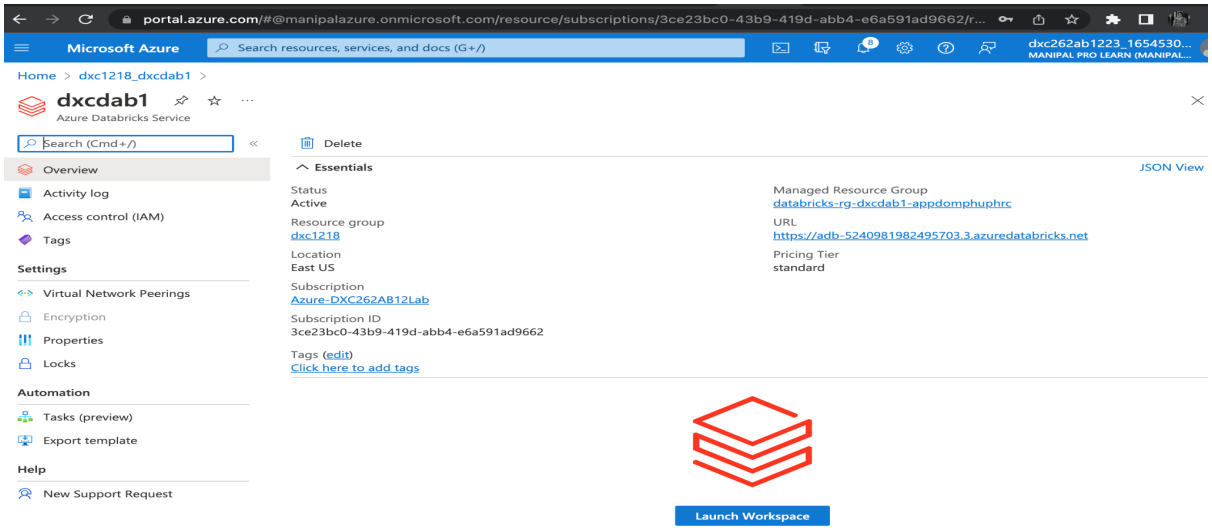
Step 4: Now click on “Review+create” to validate and then click on create at the left bottom to deploy the databricks.



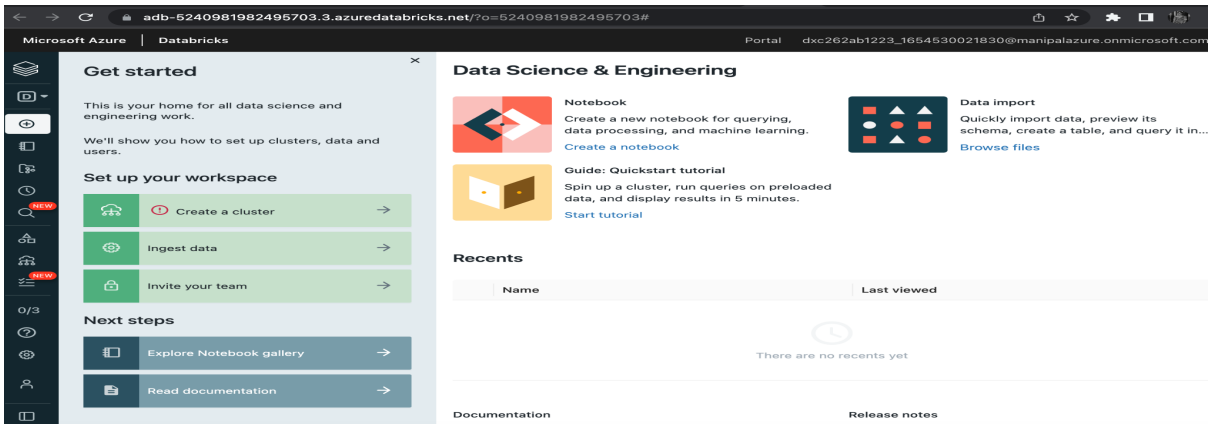
Step 5: Deployment completed.



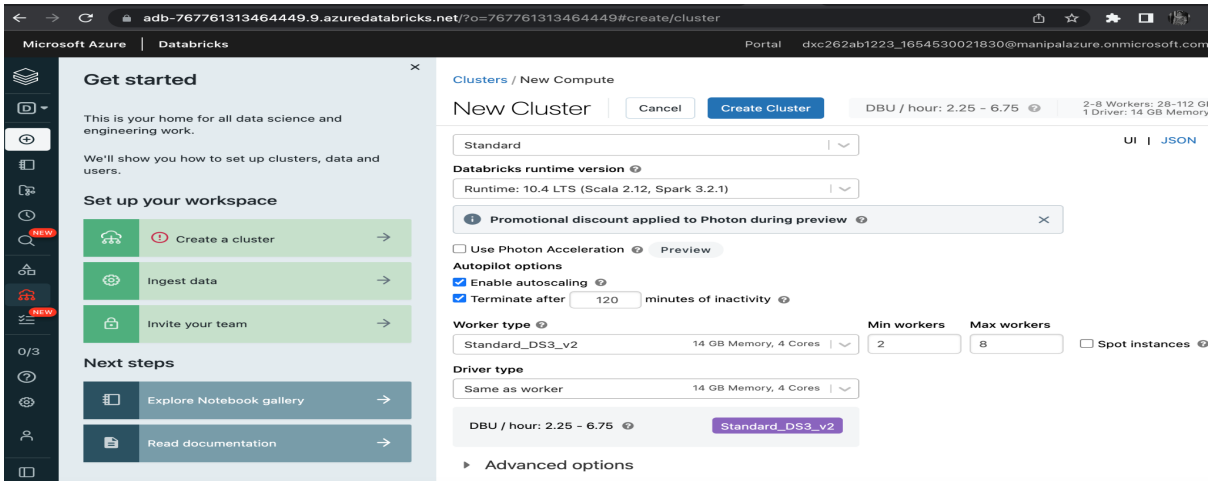
Step 6: Now click on “Go to Resource” and click on “launch workspace”.



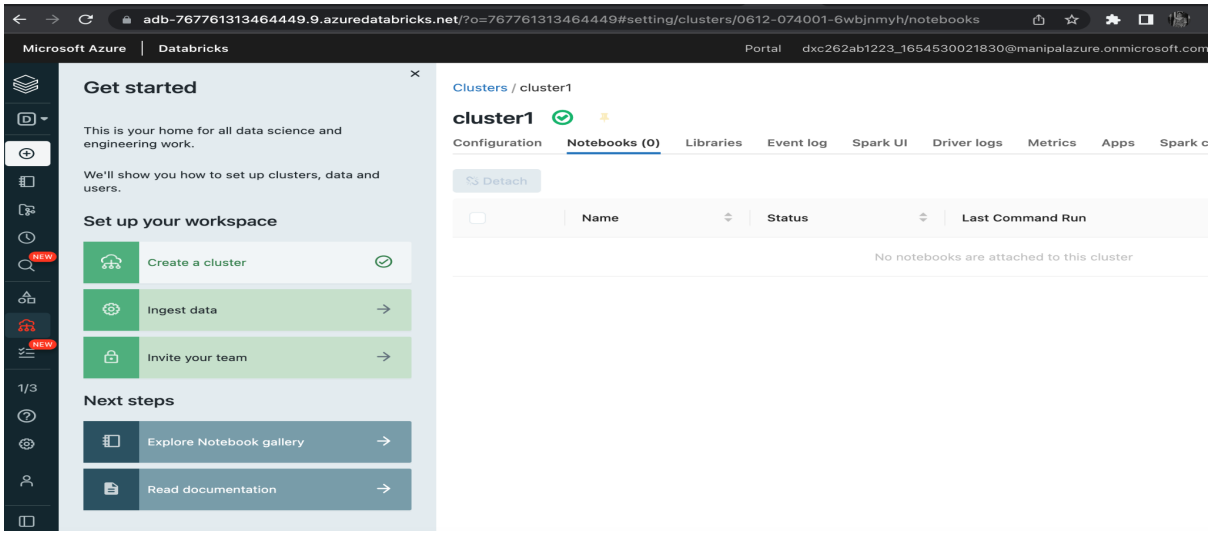
Step 7: You will sign into Azure Databricks and the Azure Databricks is created.



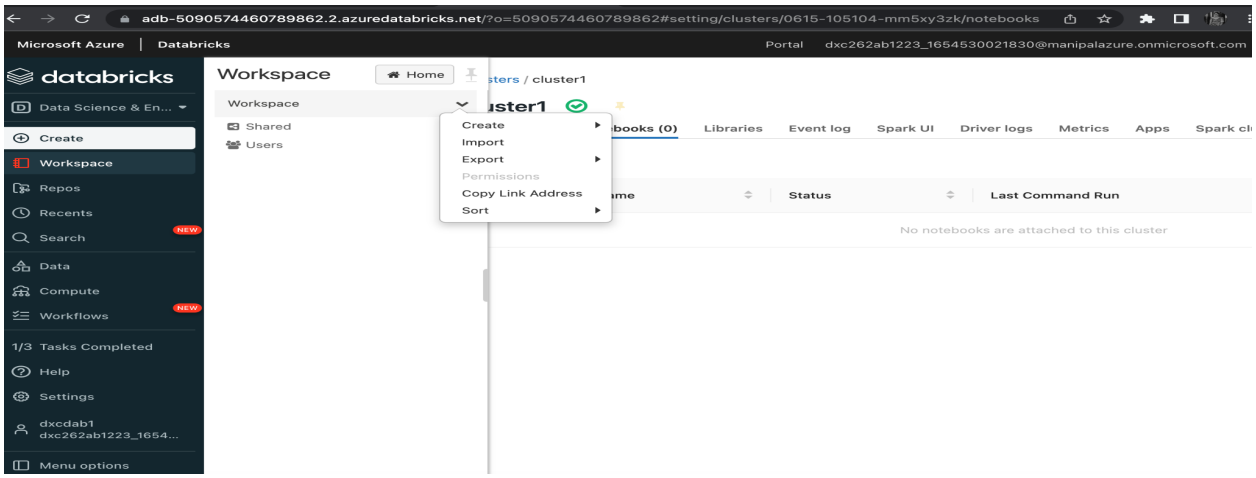
Step 8: Now create a cluster using the option compute. Give the necessary credentials for the cluster and click on create.



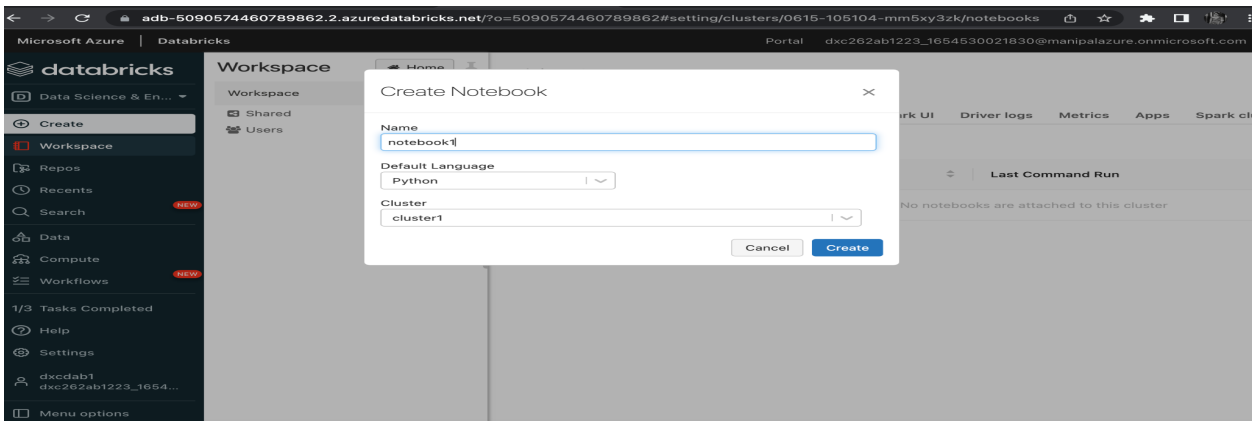
Step 9: Cluster is created.



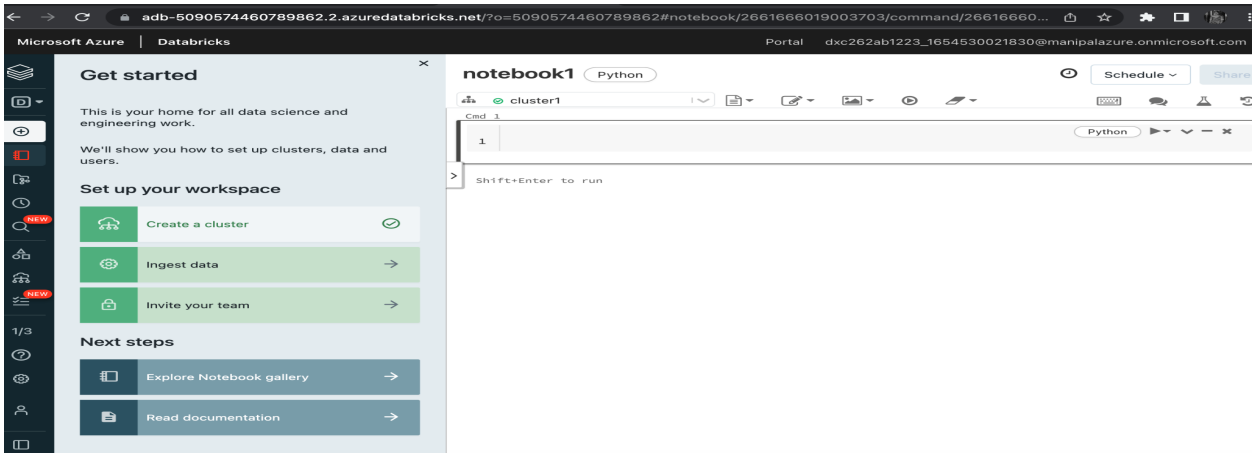
Step 10: Now create 6 notebooks for 6 questions. To create a notebook click on workspace then on create and then click on notebook.



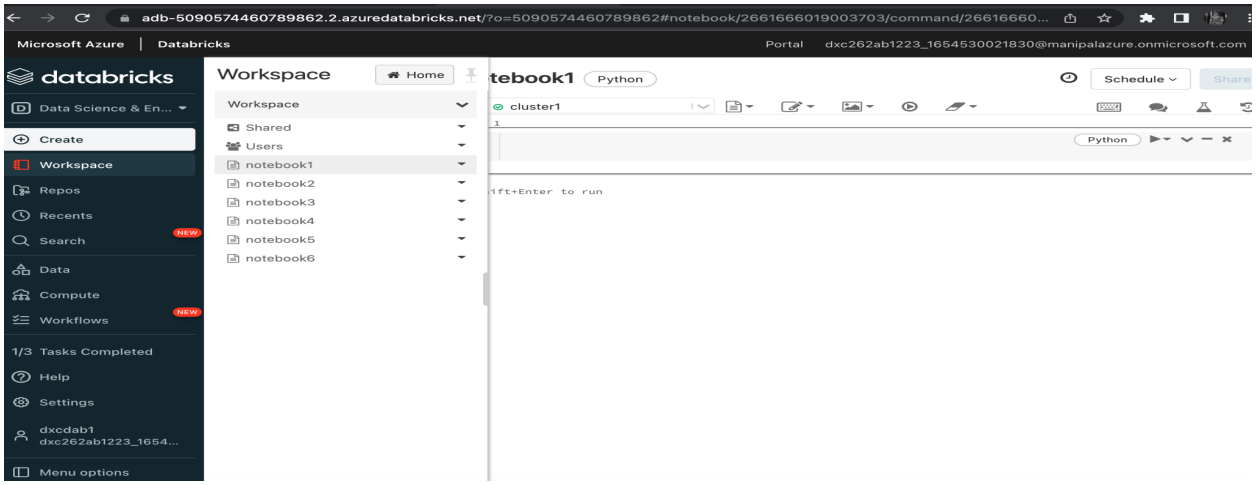
Step 11: Fill all the fields and click on create.



Step 12: Notebook is created.

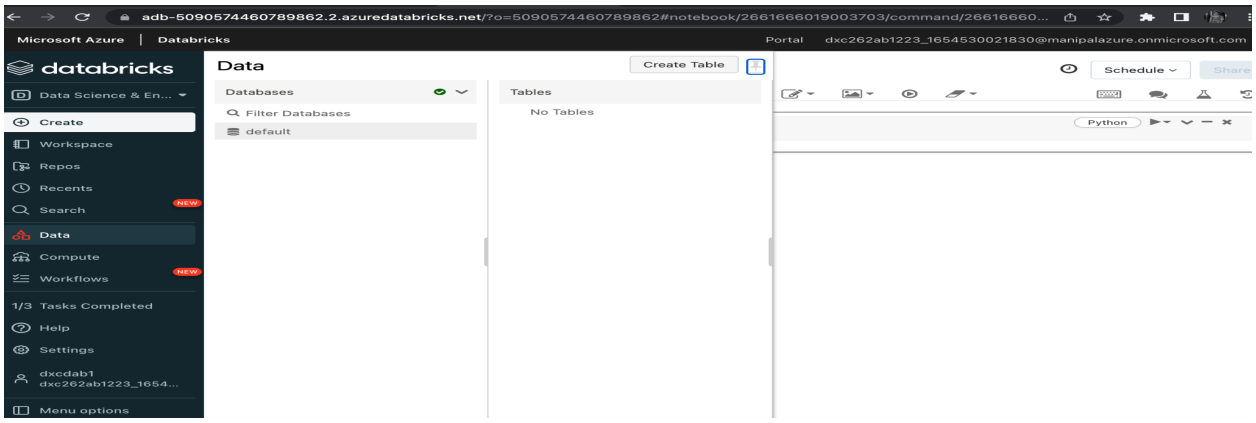


Step 13: Similarly create the remaining 5 notebooks.



All the 6 notebooks are created.

Step 14: Now upload the csv files. To upload the csv files click on data and then click on create table.



Step 15: Now browse and upload the files.

For Case 1: country-codes.csv.

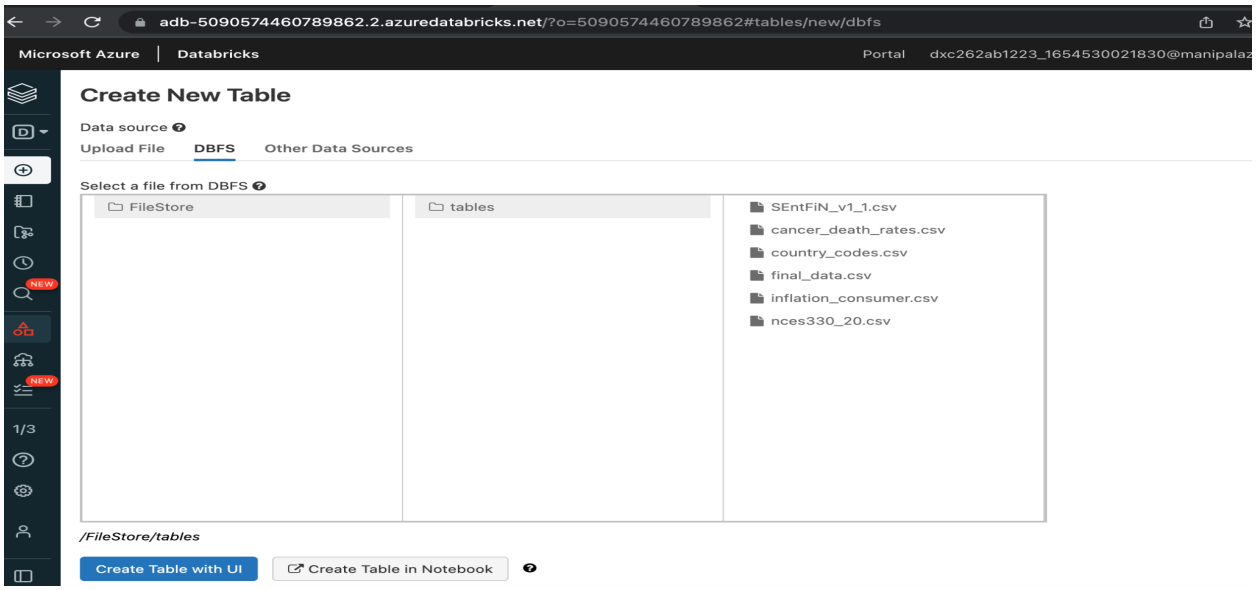
For Case 2: nces330.20.csv.

For Case 3: final_data.csv.

For Case 4: SEntFiN-v1.1.csv.

For Case 5: cancer-death-rates.csv.

For Case 6: inflation-consumer.csv.



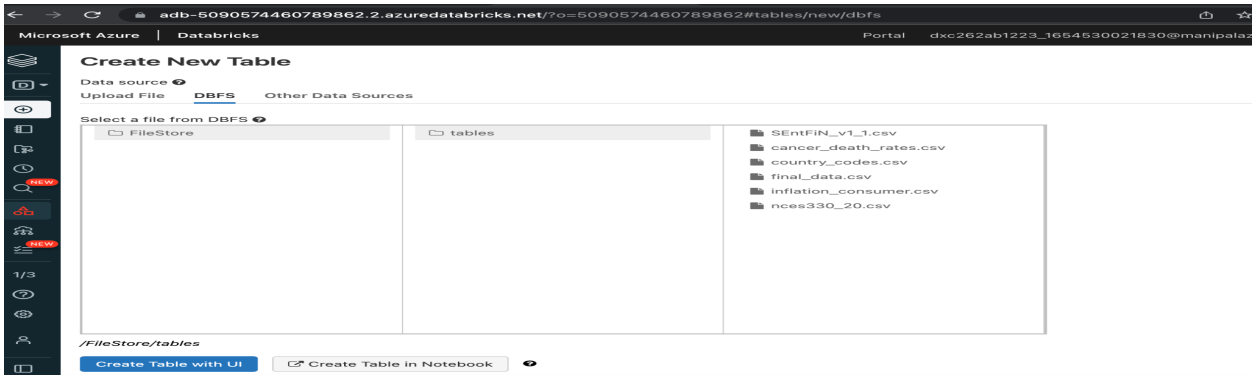
Now perform the actions that are needed in the following 6 cases.

Case 1. Using archive1.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive1.zip contains a csv file named country-codes.csv.

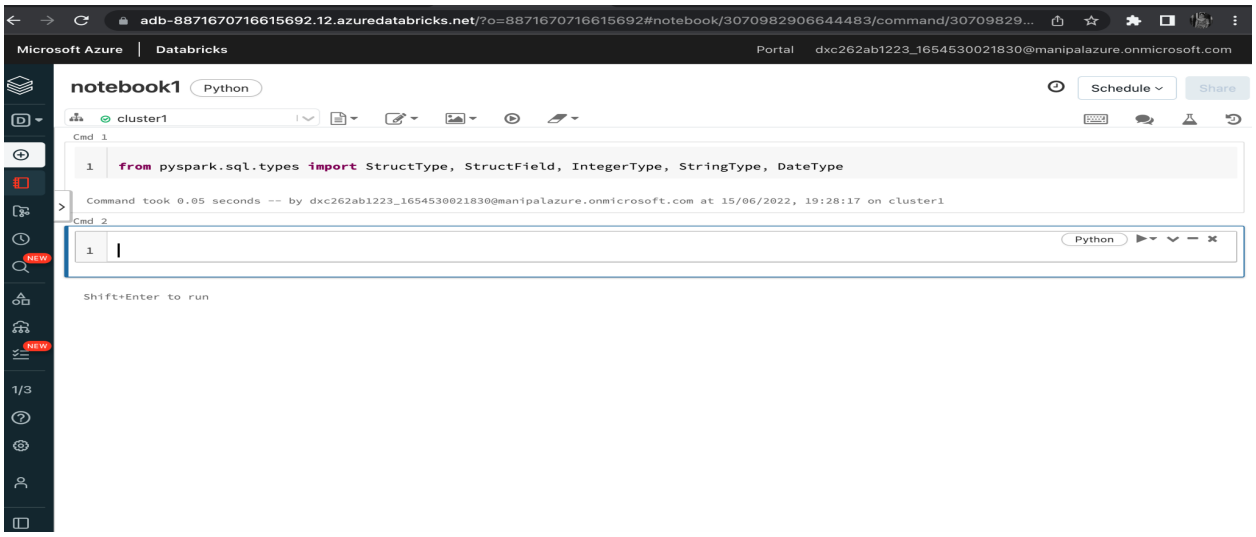
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table country-codes.csv.



Step 1: Import the required fields and features from pyspark.

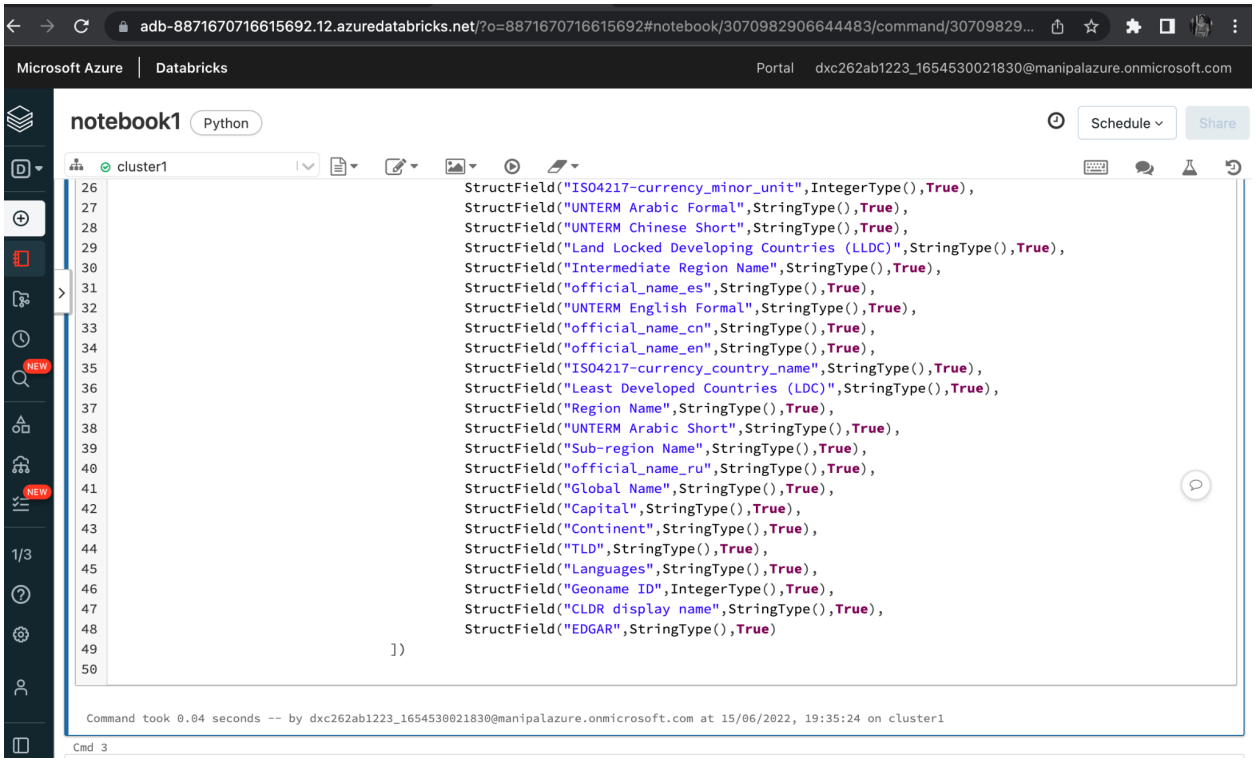
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType



Step 2:

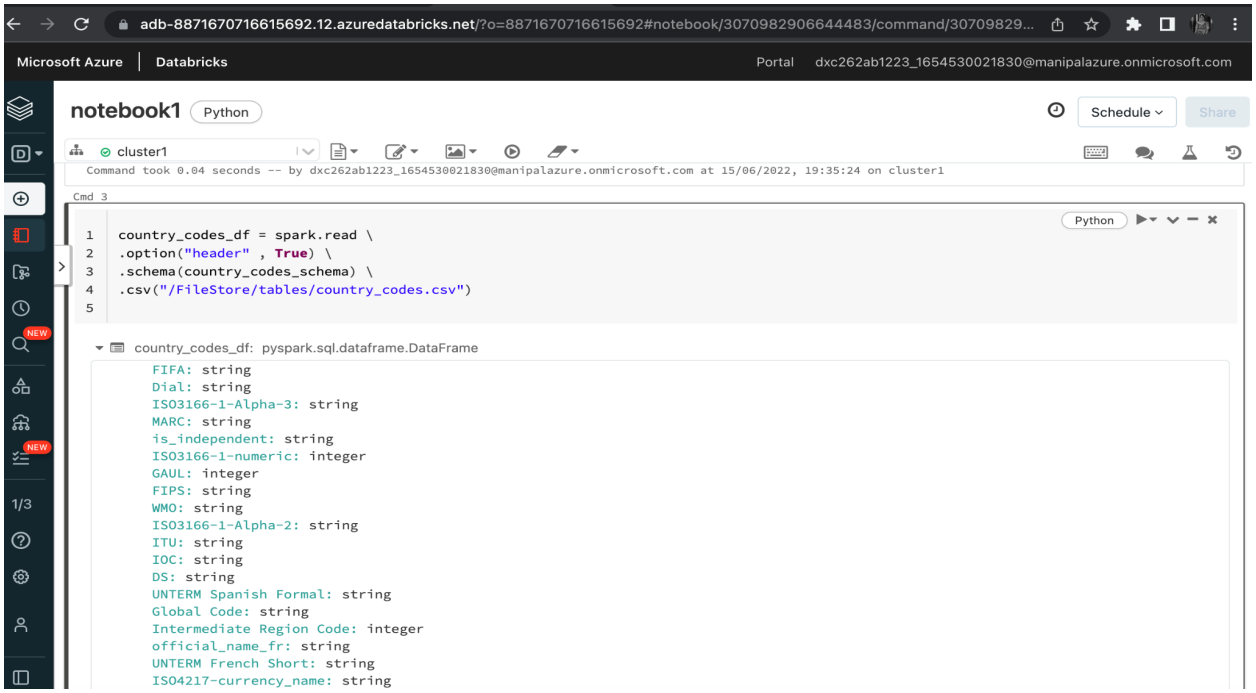
```
country_codes_schema = StructType(fields=[StructField("FIFA", StringType(),False),
                                          StructField("Dial", StringType(),True),
                                          StructField("ISO3166-1-Alpha-3",StringType(),True),
                                          StructField("MARC", StringType(),True),
                                          StructField("is_independent", StringType(),True),
                                          StructField("ISO3166-1-numeric",IntegerType(),True),
                                          StructField("GAUL", IntegerType(),True),
                                          StructField("FIPS", StringType(),True),
                                          StructField("WMO", StringType(),True),
                                          StructField("ISO3166-1-Alpha-2",StringType(),True),
                                          StructField("ITU", StringType(),True),
                                          StructField("IOC", StringType(),True),
                                          StructField("DS", StringType(),True),
                                          StructField("UNTERM Spanish Formal", StringType(),True),
                                          StructField("Global Code",StringType(),True),
                                          StructField("Intermediate Region Code",IntegerType(),True),
                                          StructField("official_name_fr",StringType(),True),
                                          StructField("UNTERM French Short",StringType(),True),
                                          StructField("ISO4217-currency_name",StringType(),True),
                                          StructField("Developed / DevelopingCountries",
StringType(),True),
                                          StructField("UNTERM Russian Formal",StringType(),True),
                                          StructField("UNTERM English Short",StringType(),True),

StructField("ISO4217-currency_alphabetic_code",StringType(),True),
                                          StructField("Small Island Developing States
(SIDS)",StringType(),True),
```

```
                    StructField("UNTERM Spanish Short",StringType(),True),

StructField("ISO4217-currency_numeric_code",IntegerType(),True),
                    StructField("UNTERM Chinese Formal",StringType(),True),
                    StructField("UNTERM French Formal",StringType(),True),
                    StructField("UNTERM Russian Short",StringType(),True),
                    StructField("M49",IntegerType(),True),
                    StructField("Sub-region Code",IntegerType(),True),
                    StructField("Region Code",IntegerType(),True),
                    StructField("official_name_ar",StringType(),True),
                    StructField("ISO4217-currency_minor_unit",IntegerType(),True),
                    StructField("UNTERM Arabic Formal",StringType(),True),
                    StructField("UNTERM Chinese Short",StringType(),True),
                    StructField("Land Locked Developing Countries
(LLDC)",StringType(),True),
                    StructField("Intermediate Region Name",StringType(),True),
                    StructField("official_name_es",StringType(),True),
                    StructField("UNTERM English Formal",StringType(),True),
                    StructField("official_name_cn",StringType(),True),
                    StructField("official_name_en",StringType(),True),

StructField("ISO4217-currency_country_name",StringType(),True),
                    StructField("Least Developed Countries
(LDC)",StringType(),True),
                    StructField("Region Name",StringType(),True),
                    StructField("UNTERM Arabic Short",StringType(),True),
                    StructField("Sub-region Name",StringType(),True),
                    StructField("official_name_ru",StringType(),True),
                    StructField("Global Name",StringType(),True),
                    StructField("Capital",StringType(),True),
                    StructField("Continent",StringType(),True),
                    StructField("TLD",StringType(),True),
                    StructField("Languages",StringType(),True),
                    StructField("Geoname ID",IntegerType(),True),
                    StructField("CLDR display name",StringType(),True),
                    StructField("EDGAR",StringType(),True)
                ])
```



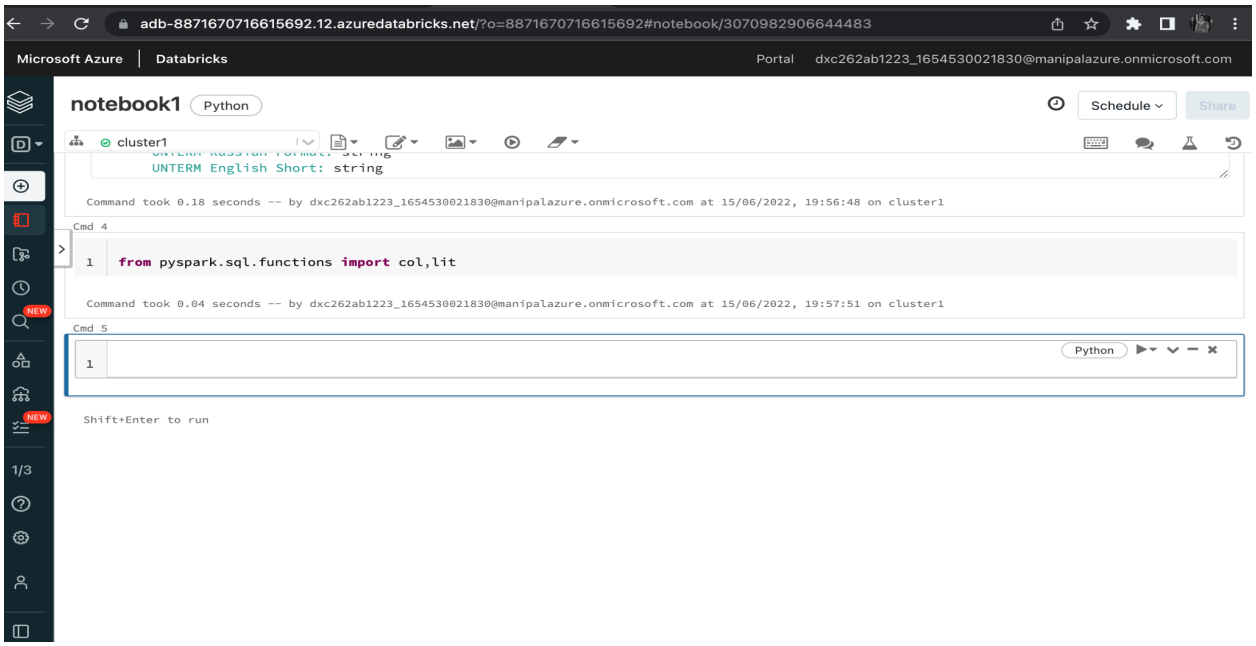
Step 3:Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

**country_codes_df = spark.read **
**.option("header" , True) **
**.schema(country_codes_schema) **
.csv("/FileStore/tables/country_codes.csv")



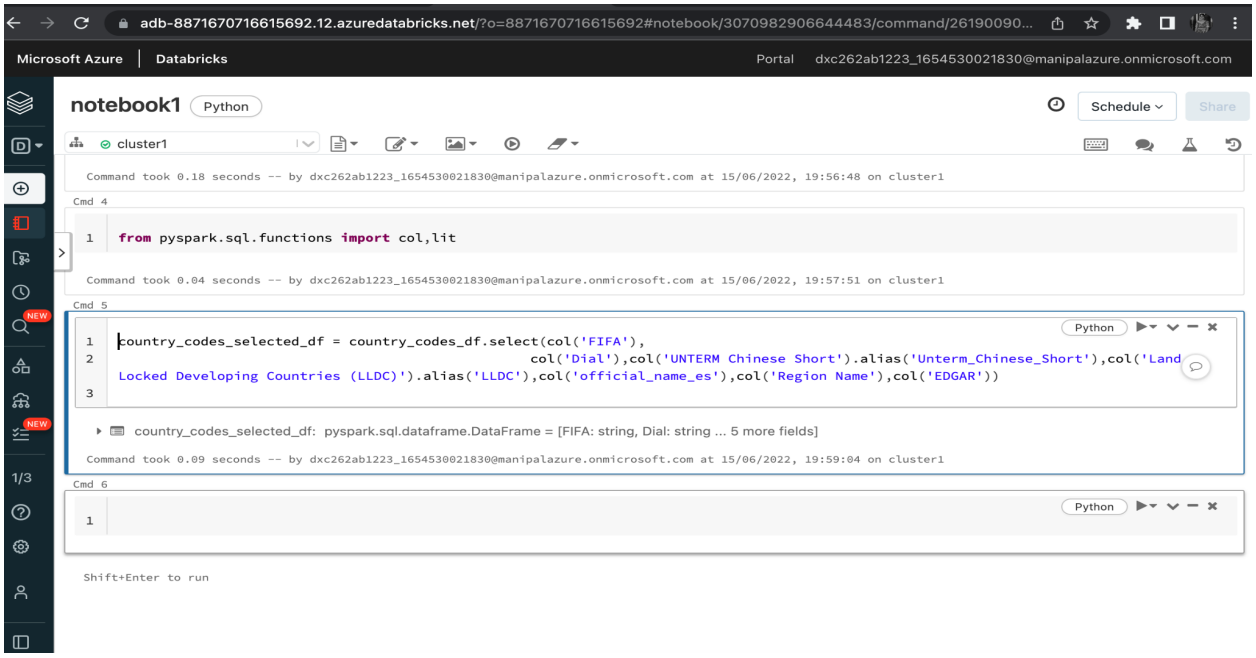
Step 4:

from pyspark.sql.functions import col,lit

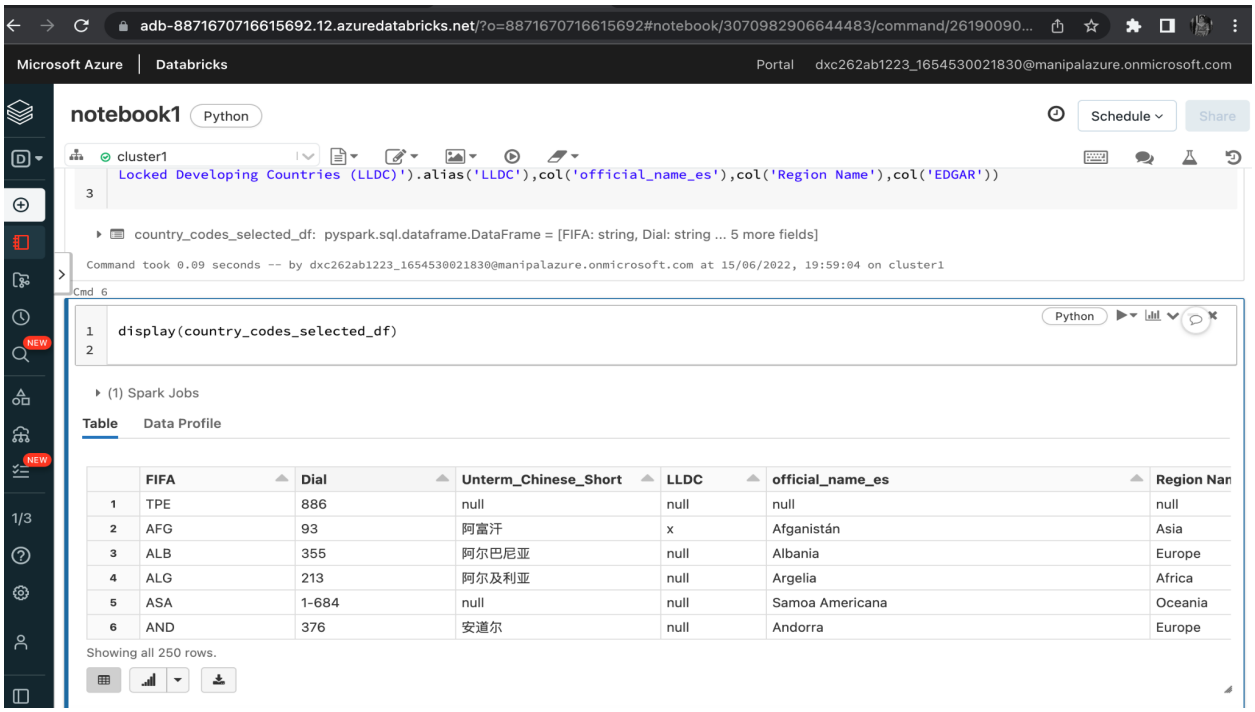


Step 5:

country_codes_selected_df = country_codes_df.select(col('FIFA'),
col('Dial'),col('UNTERM Chinese
Short').alias('Unterm_Chinese_Short'),col('Land Locked Developing Countries (LLDC)').alias('LLDC'),col('official_name_es'),col('Region Name'),col('EDGAR'))



Step 6:
display(country_codes_selected_df)

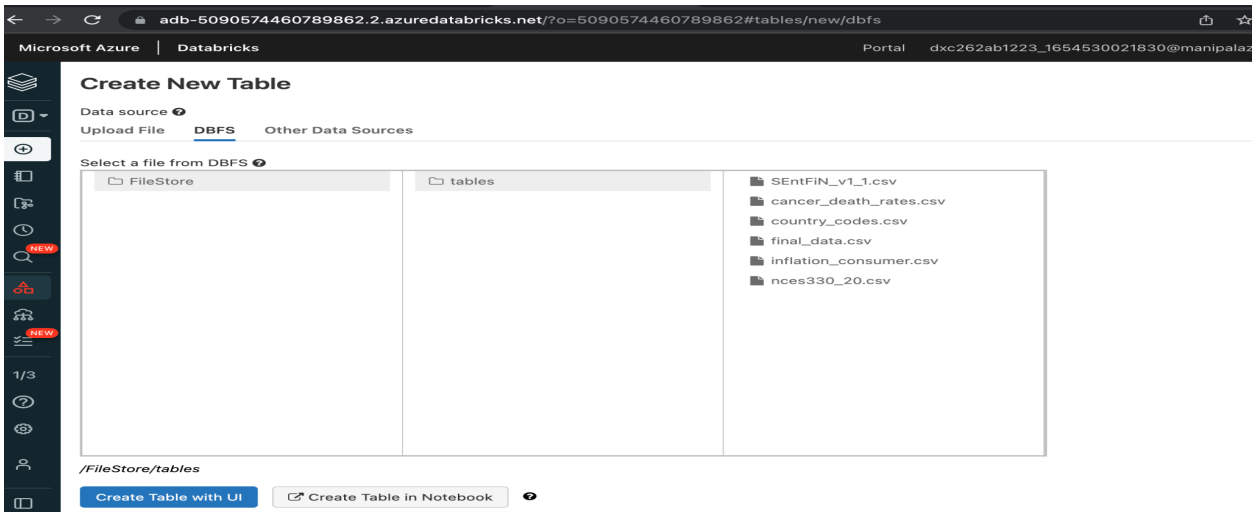


Case 2. Using archive2.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive2.zip contains a csv file named nces330.20.csv.

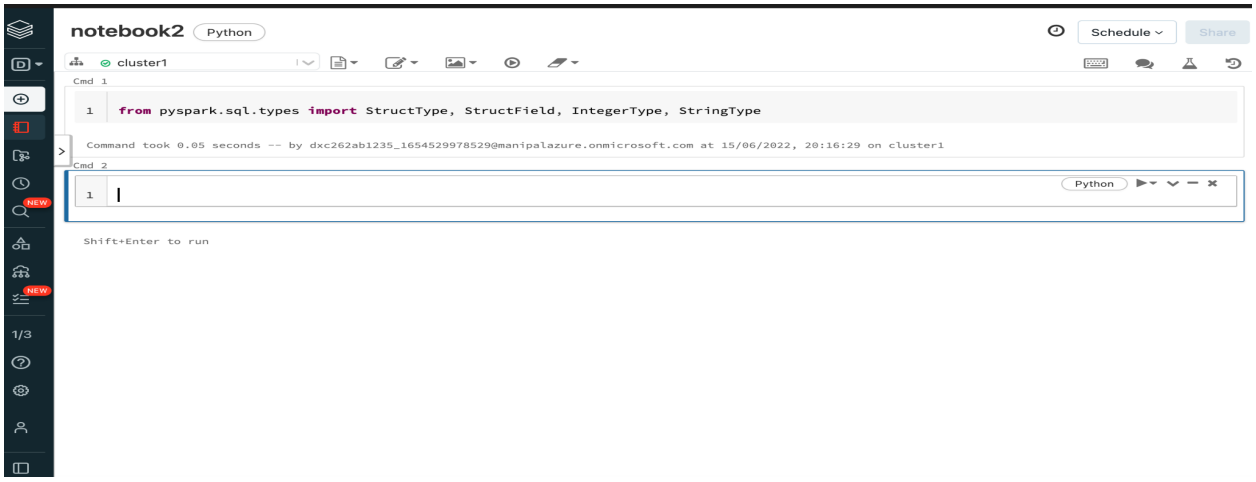
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table nces330.20.csv.



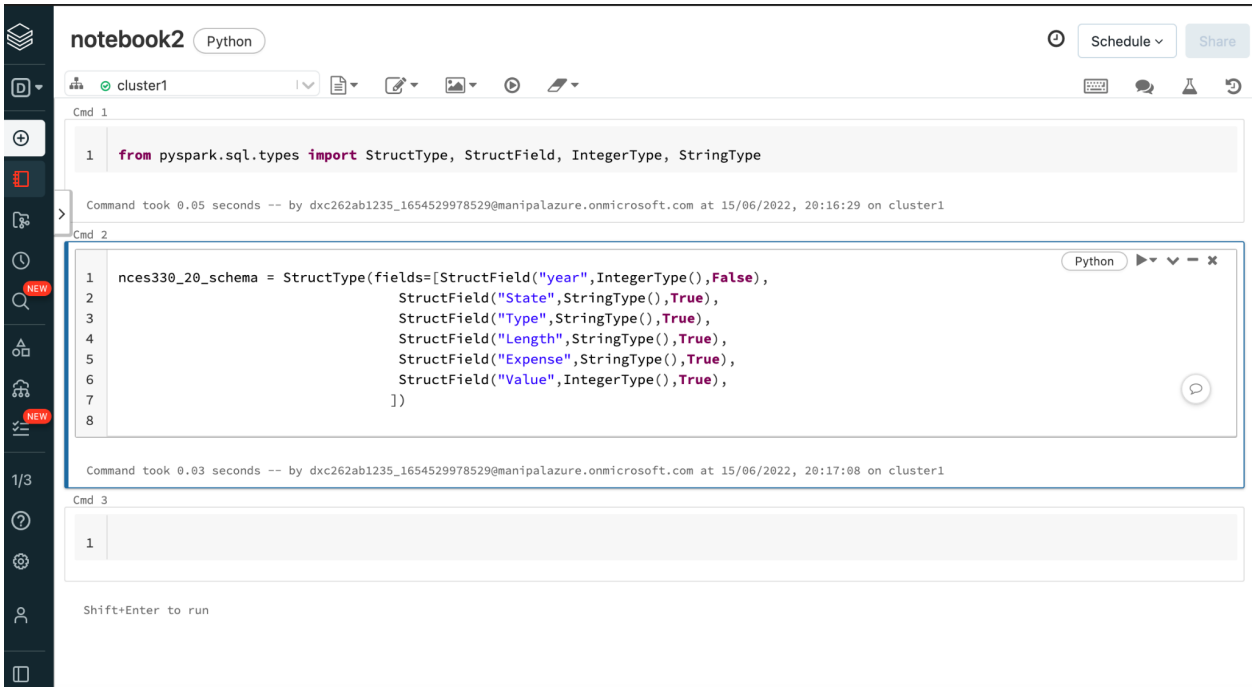
Step 1: Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType



Step 2:

```
nces330_20_schema = StructType(fields=[StructField("year",IntegerType(),False),
                                       StructField("State",StringType(),True),
                                       StructField("Type",StringType(),True),
                                       StructField("Length",StringType(),True),
                                       StructField("Expense",StringType(),True),
                                       StructField("Value",IntegerType(),True),
                                      ])
```



Step 3: Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

**nces330_20_df = spark.read **

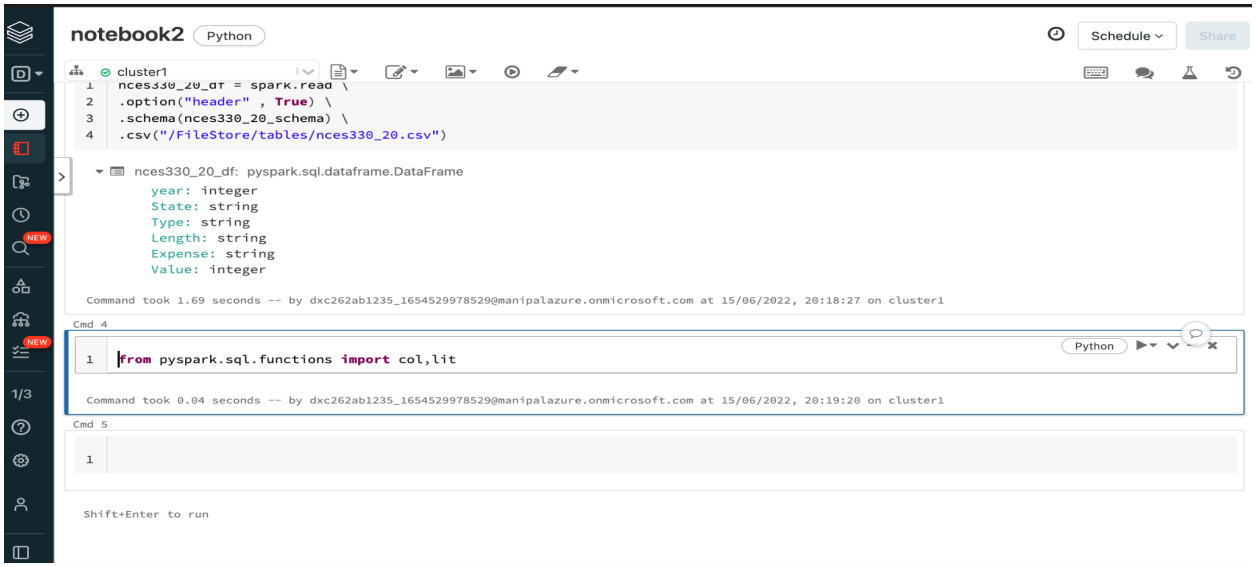
**.option("header" , True) **

**.schema(nces330_20_schema) **

.csv("/FileStore/tables/nces330_20.csv")

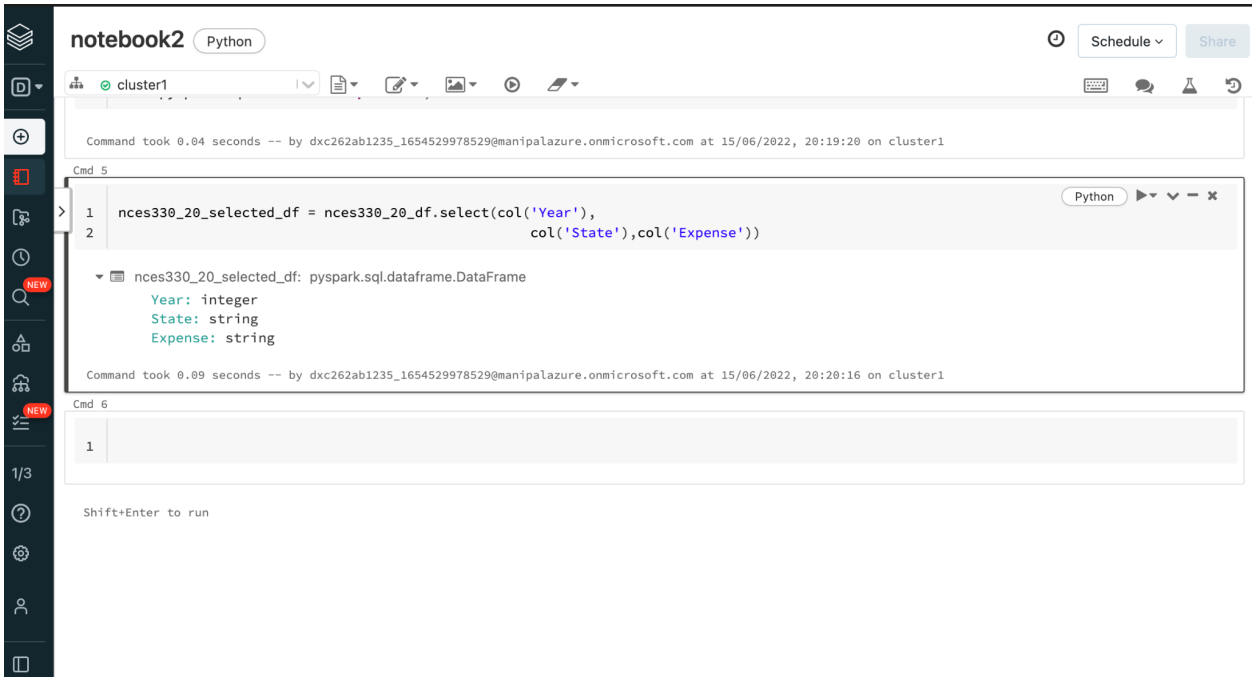
Step 4:

from pyspark.sql.functions import col,lit



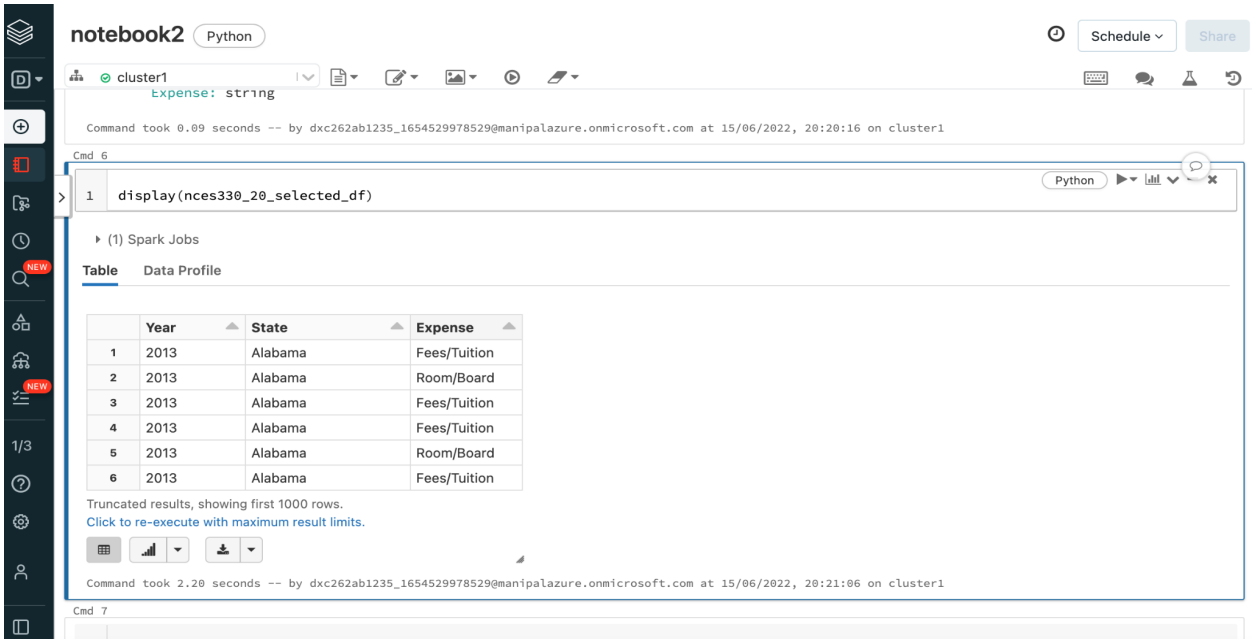
Step 5:

```
nces330_20_selected_df = nces330_20_df.select(col('Year'),
                                              col('State'),col('Expense'))
```



Step 6:

```
display(nces330_20_selected_df)
```

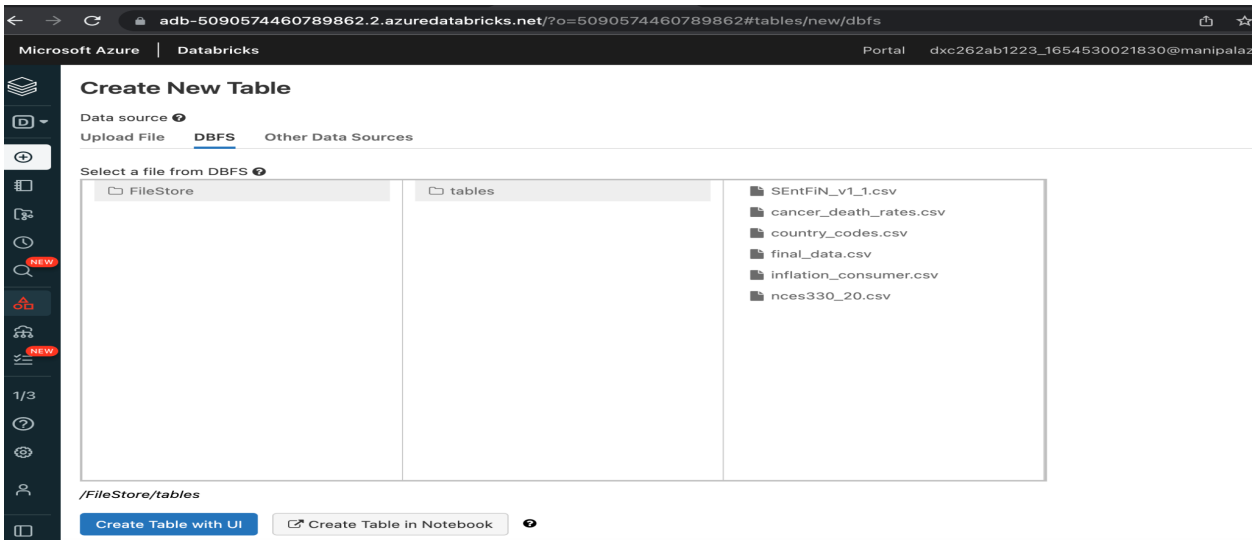


Case 3. Using archive3.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive3.zip contains a csv file named final_data.csv.

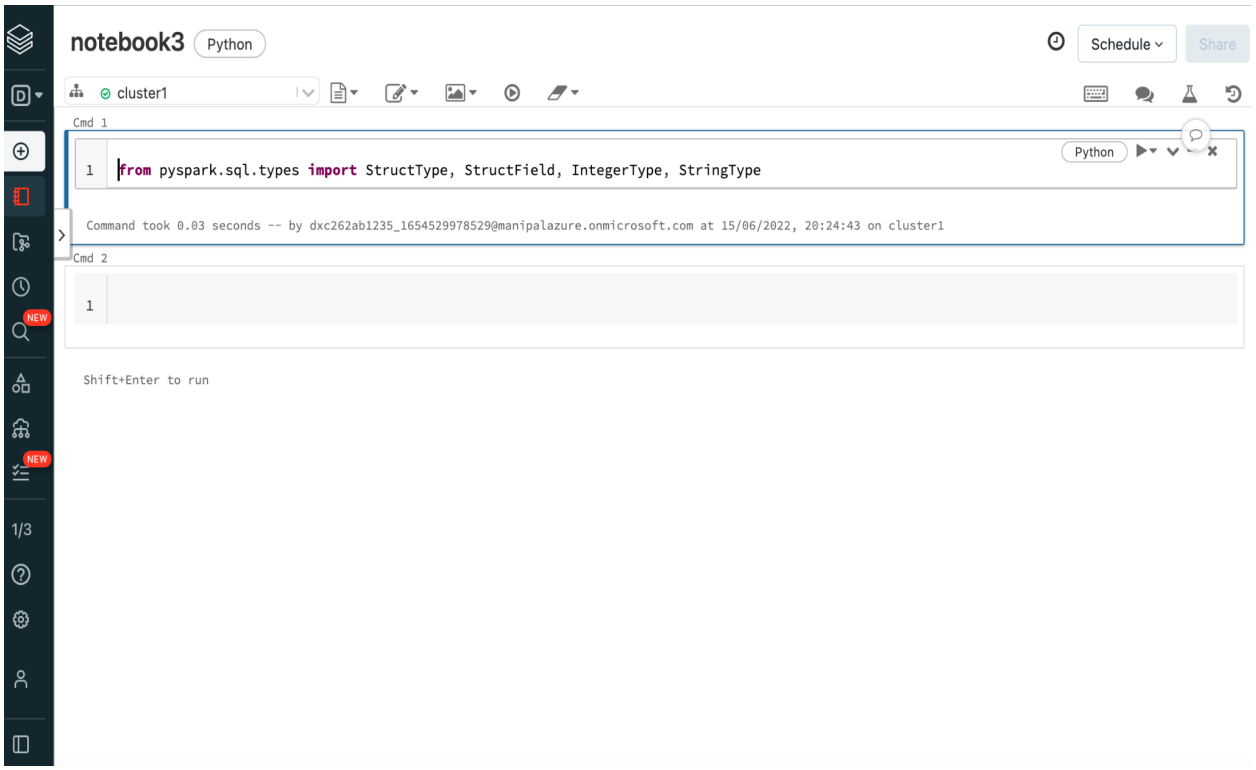
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table final_data.csv.



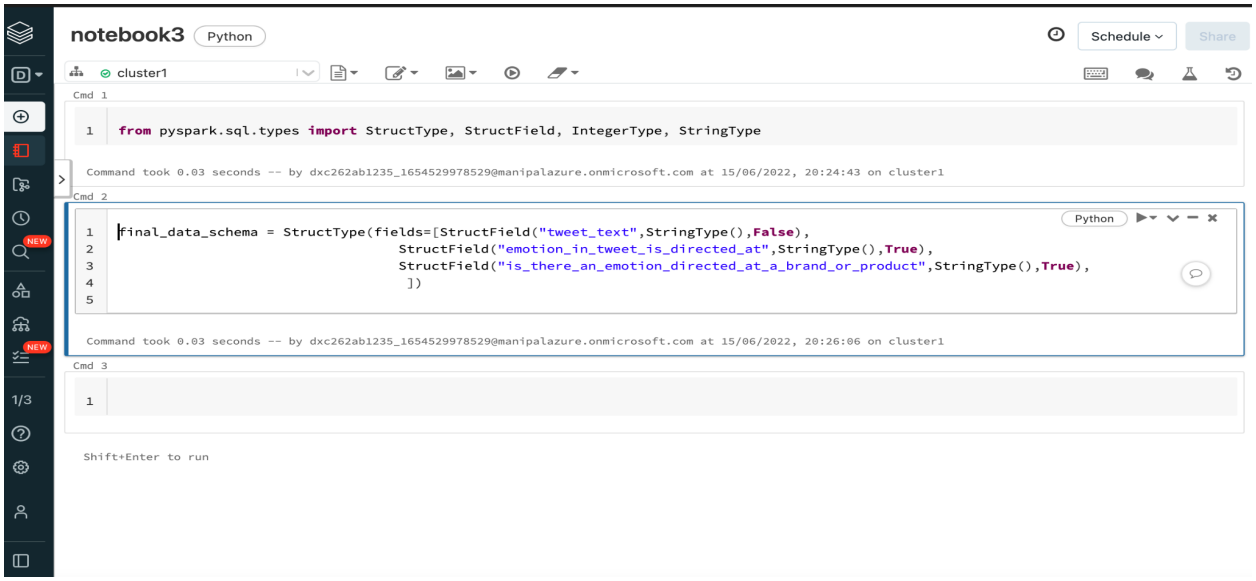
Step 1: Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType



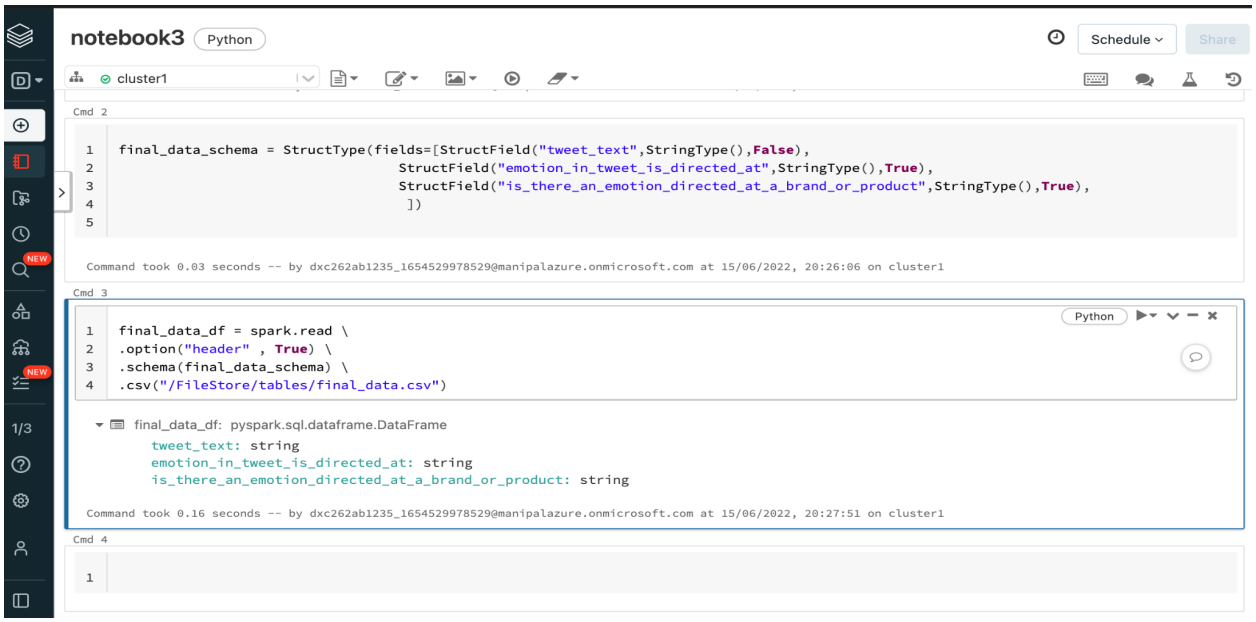
Step 2:

```
final_data_schema = StructType(fields=[StructField("tweet_text",StringType(),False),
                                       StructField("emotion_in_tweet_is_directed_at",StringType(),True),

StructField("is_there_an_emotion_directed_at_a_brand_or_product",StringType(),True),
                                       ])
```

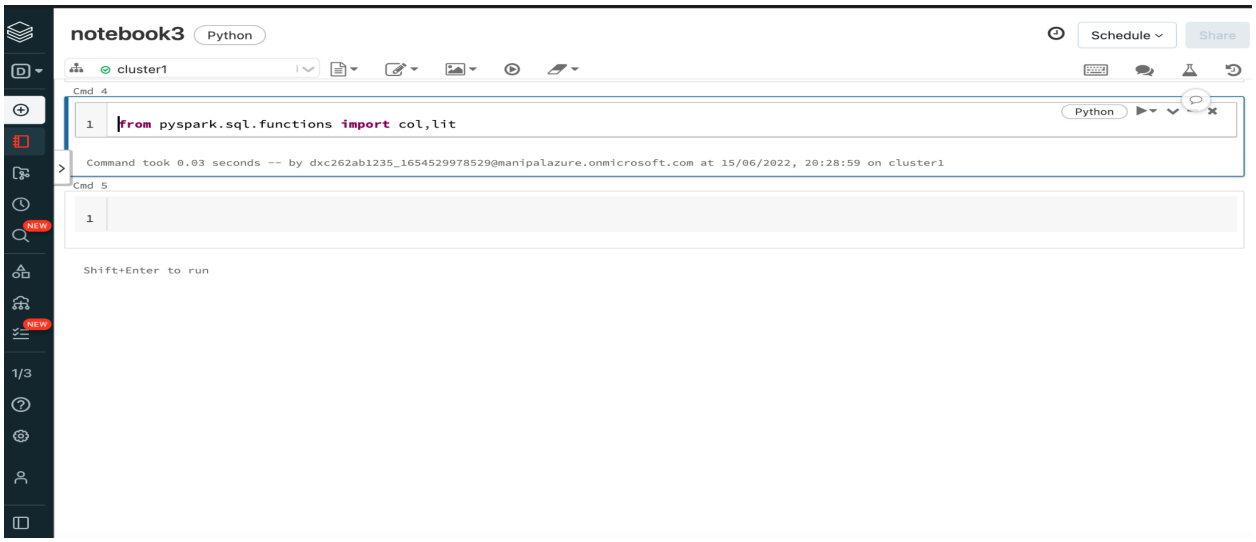


Step 3: Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

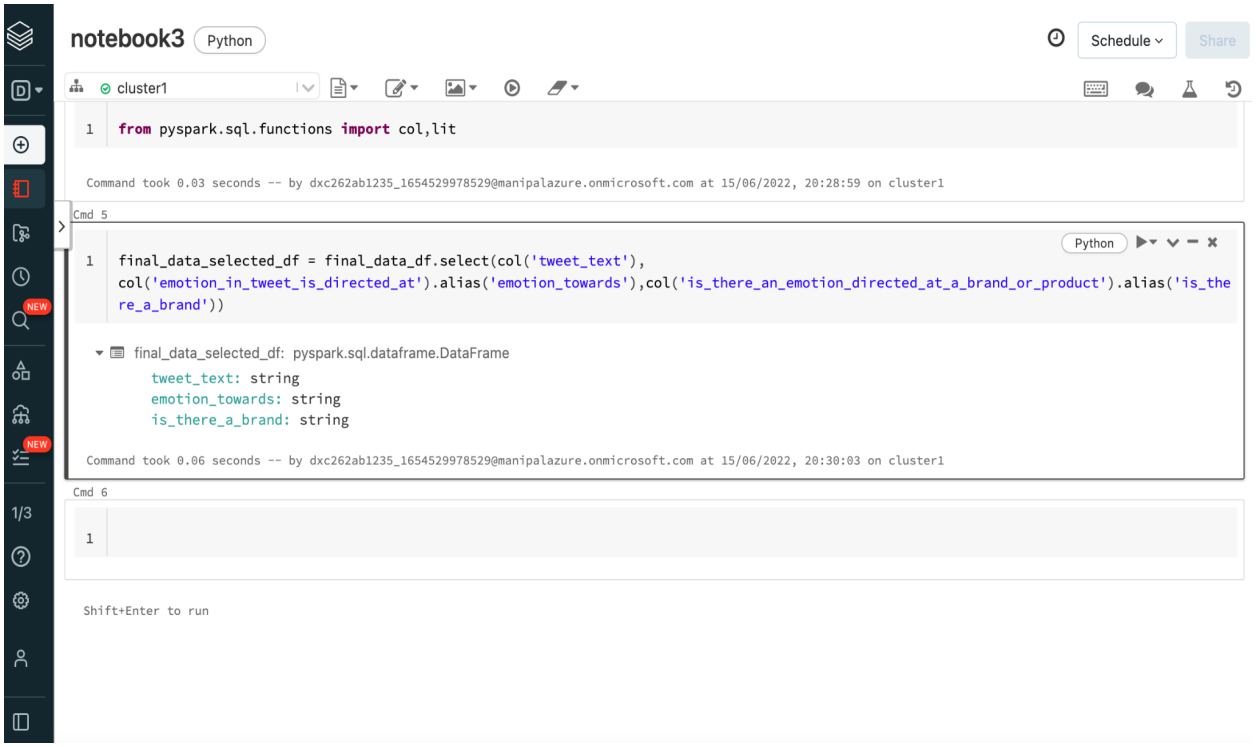
```
final_data_df = spark.read \
.option("header" , True) \
.schema(final_data_schema) \
.csv("/FileStore/tables/final_data.csv")
```



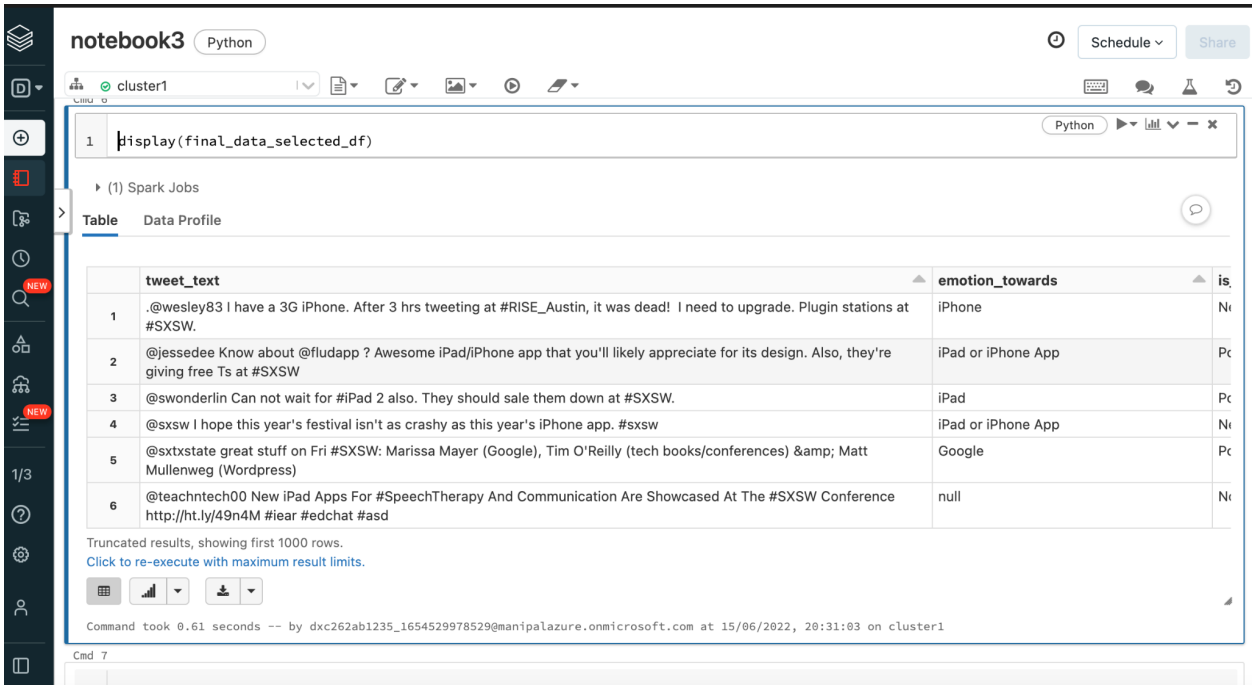
Step 4:
from pyspark.sql.functions import col,lit



Step 5:
final_data_selected_df = final_data_df.select(col('tweet_text'), col('emotion_in_tweet_is_directed_at').alias('emotion_towards'),col('is_there_an_emotion_directed_at_a_brand_or_product').alias('is_there_a_brand'))



Step 6:
display(final_data_selected_df)

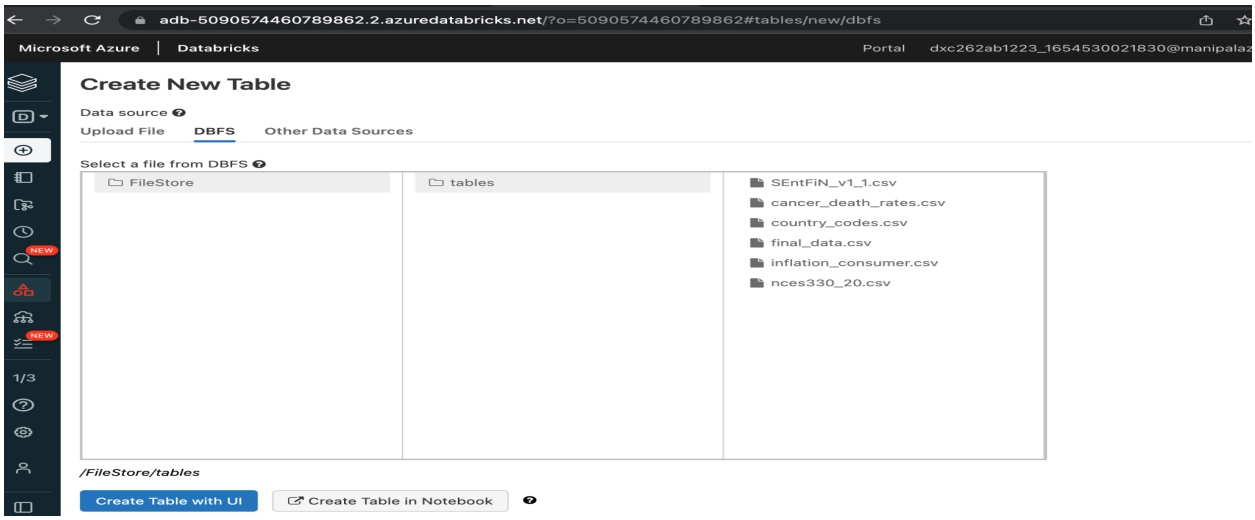


Case 4. Using archive4.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive4.zip contains a csv file named SEntFiN-v1.1.csv.

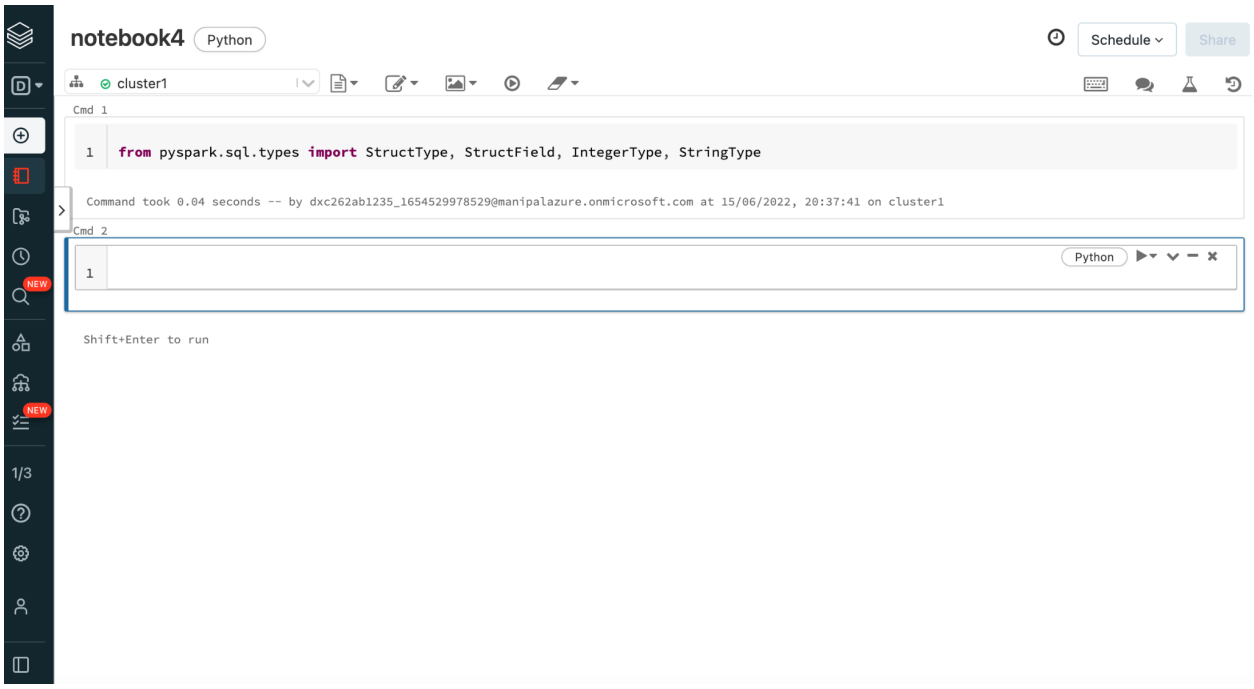
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table SEntFiN-v1.1.csv.



Step 1: Import the required fields and features from pyspark.

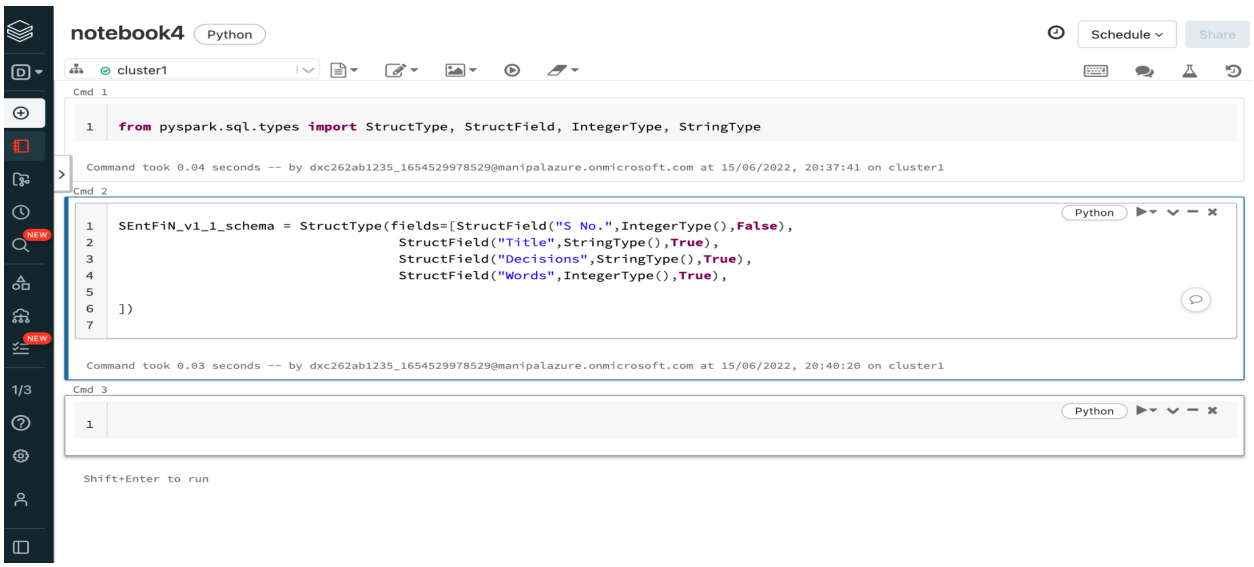
from pyspark.sql.types import StructType, StructField, IntegerType, StringType



Step 2:

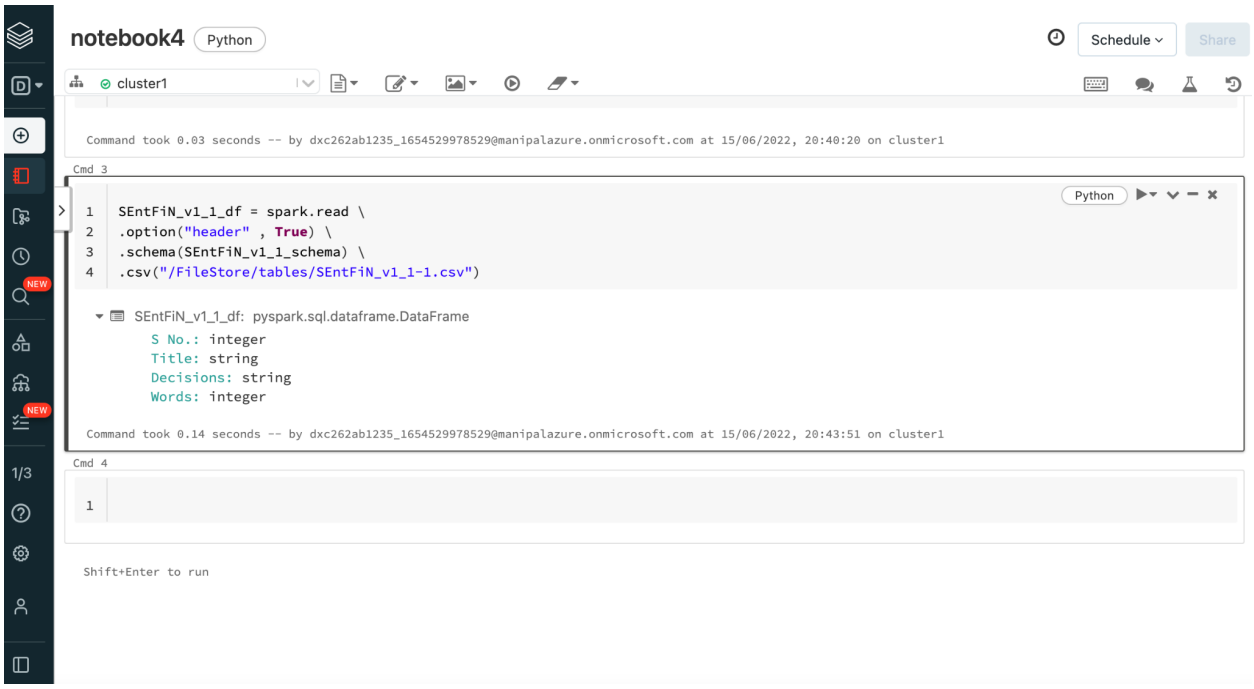
SEntFiN_v1_1_schema = StructType(fields=[StructField("S No.",IntegerType(),False),
StructField("Title",StringType(),True),
StructField("Decisions",StringType(),True),
StructField("Words",IntegerType(),True),

])



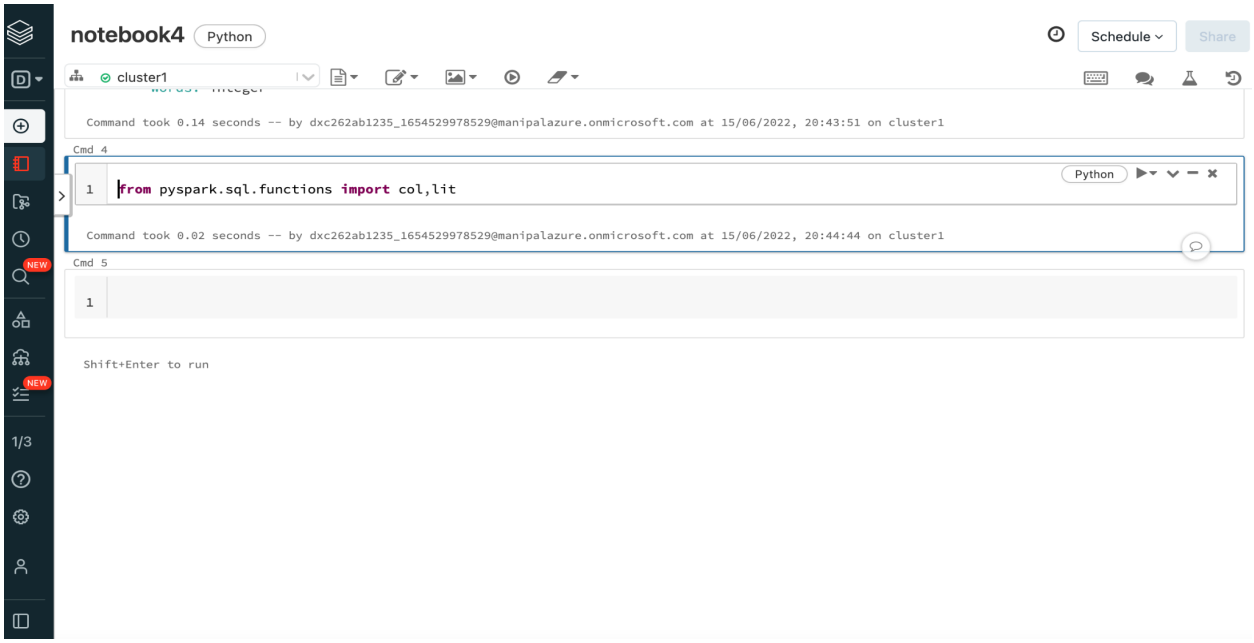
Step 3: Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

**SEntFiN_v1_1_df = spark.read **
**.option("header" , True) **
**.schema(SEntFiN_v1_1_schema) **
.csv("/FileStore/tables/SEntFiN_v1_1-1.csv")

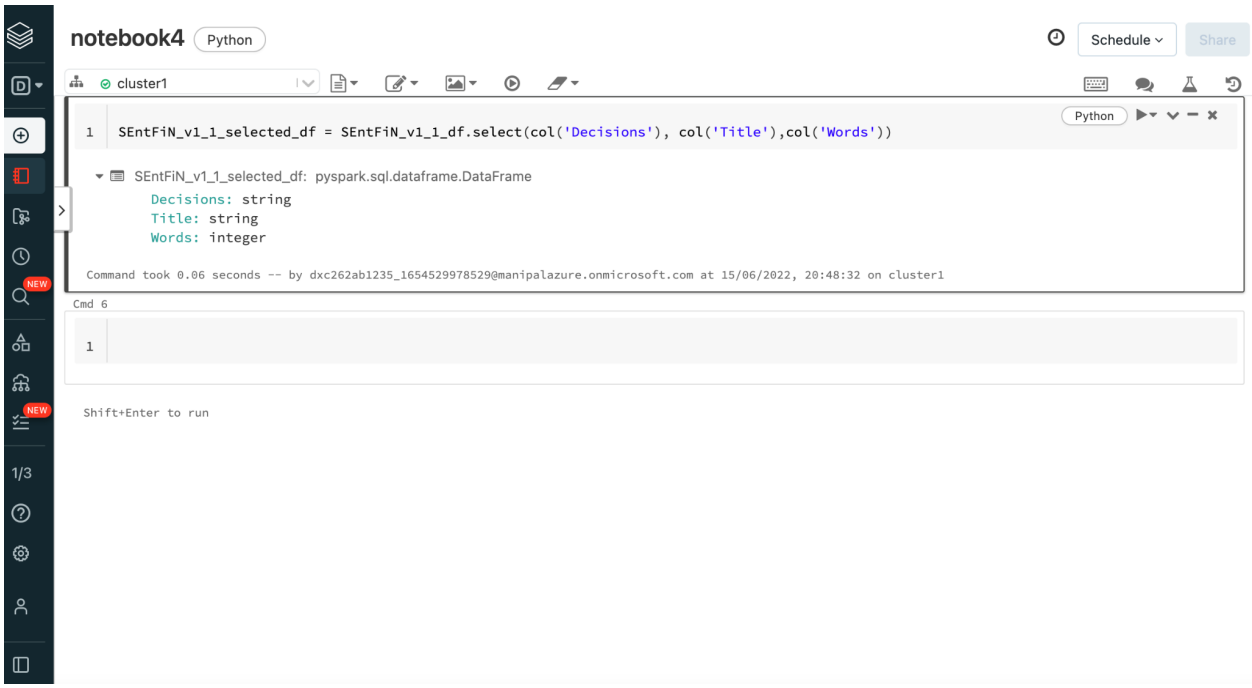


Step 4:

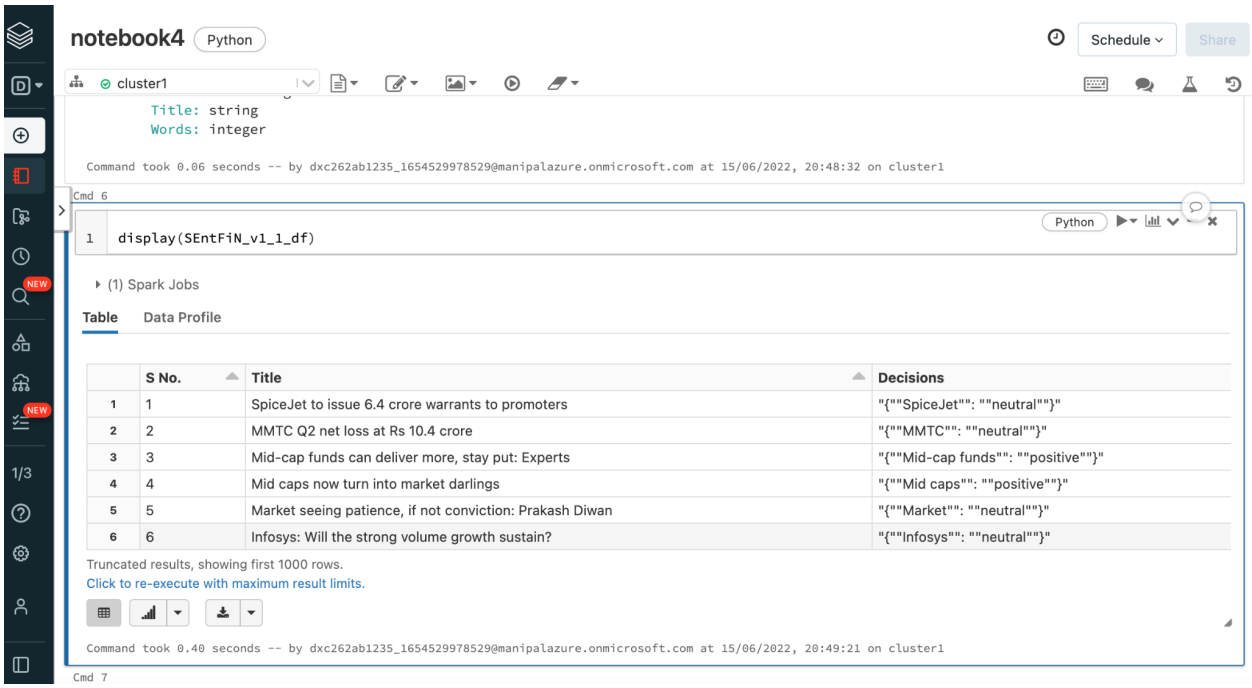
from pyspark.sql.functions import col,lit



Step 5:
SEntFiN_v1_1_selected_df = SEntFiN_v1_1_df.select(col('Decisions'), col('Title'),col('Words'))



Step 6:
display(SEntFiN_v1_1_df)

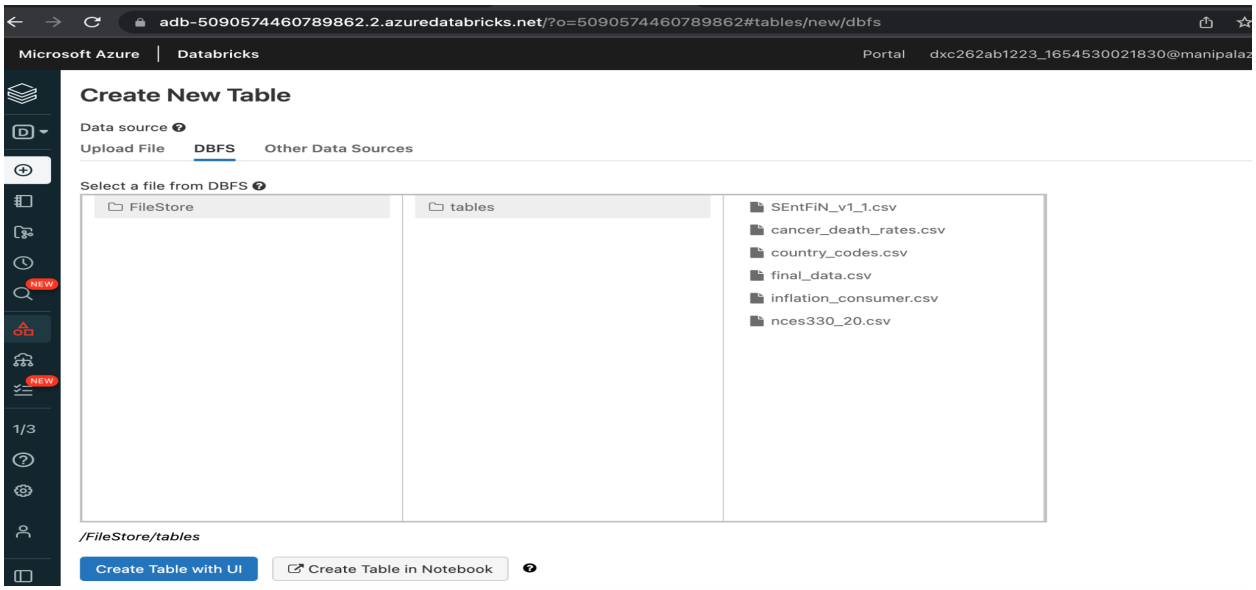


Case 5. Using archive5.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive5.zip contains a csv file named cancer-death-rates.csv.

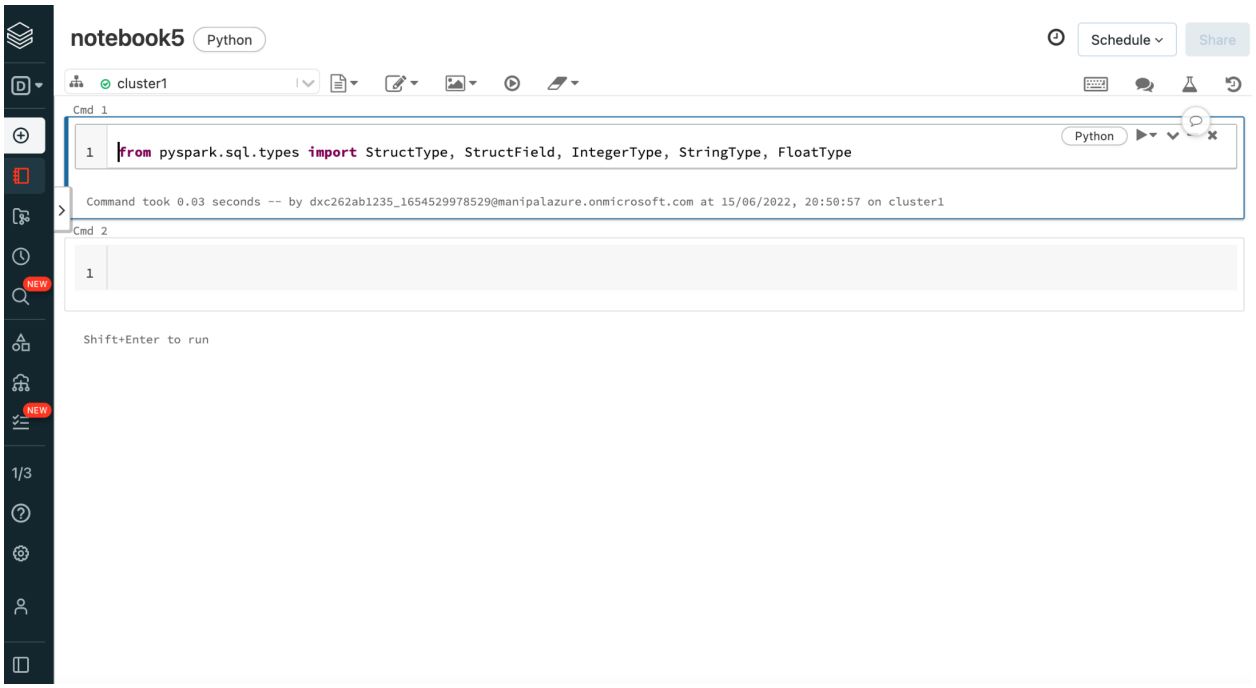
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table cancer-death-rates.csv.



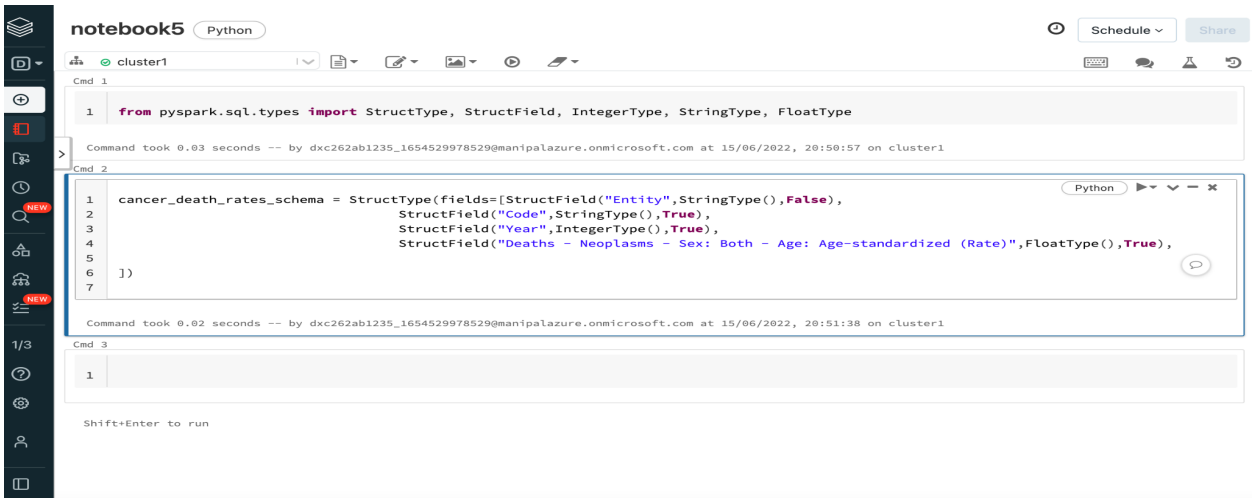
Step 1: Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, FloatType



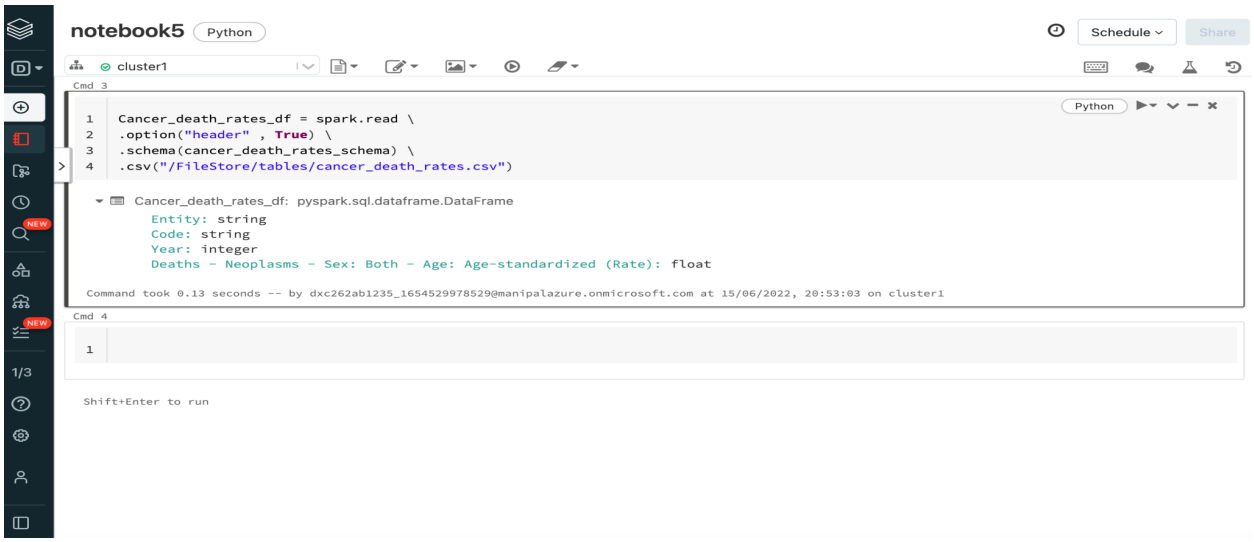
Step 2:

```
cancer_death_rates_schema =
StructType(fields=[StructField("Entity",StringType(),False),
                   StructField("Code",StringType(),True),
                   StructField("Year",IntegerType(),True),
                   StructField("Deaths - Neoplasms - Sex: Both - Age: Age-standardized
(Rate)",FloatType(),True),

])
```



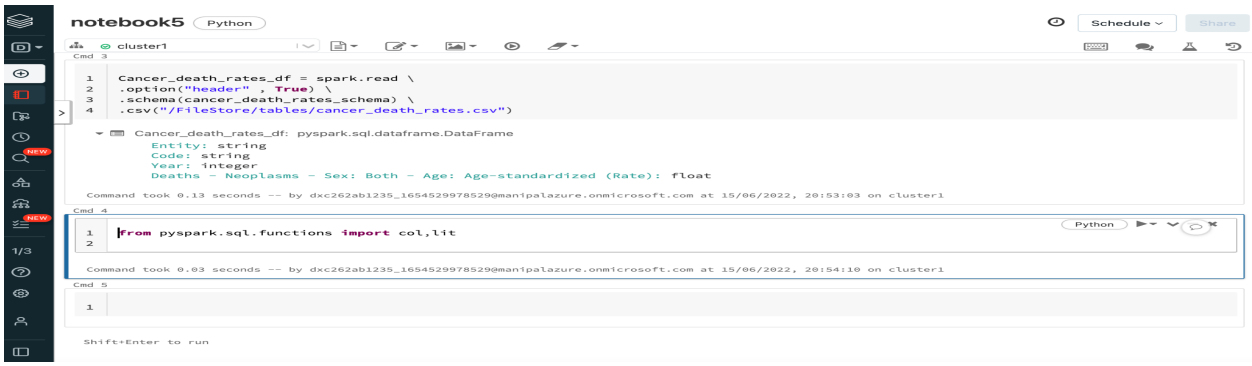
Step 3: Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

```
Cancer_death_rates_df = spark.read \
.option("header" , True) \
.schema(cancer_death_rates_schema) \
.csv("/FileStore/tables/cancer_death_rates.csv")
```



Step 4:

from pyspark.sql.functions import col,lit

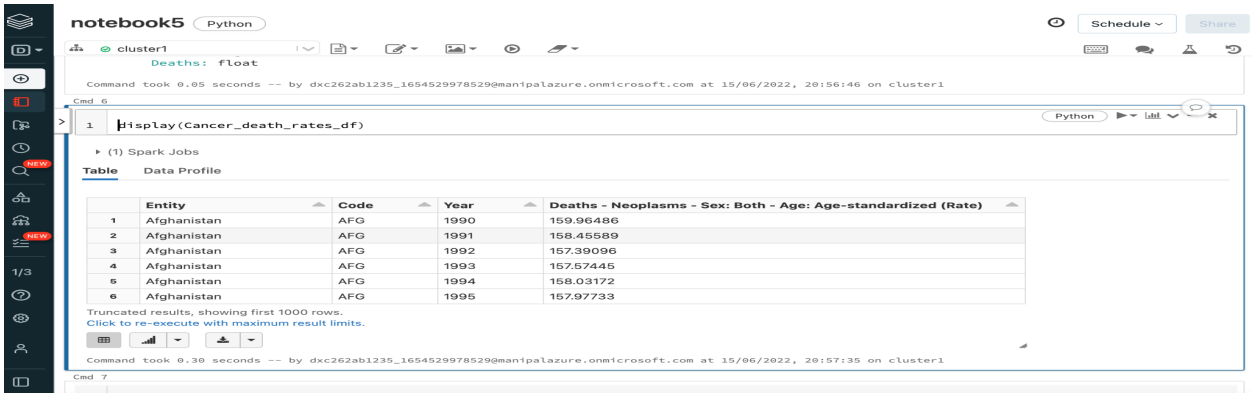


Step 5:

Cancer_death_rates_selected_df = Cancer_death_rates_df.select(col('Entity'), col('Year'),col('Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate)').alias('Deaths'))

Step 6:

display(Cancer_death_rates_df)

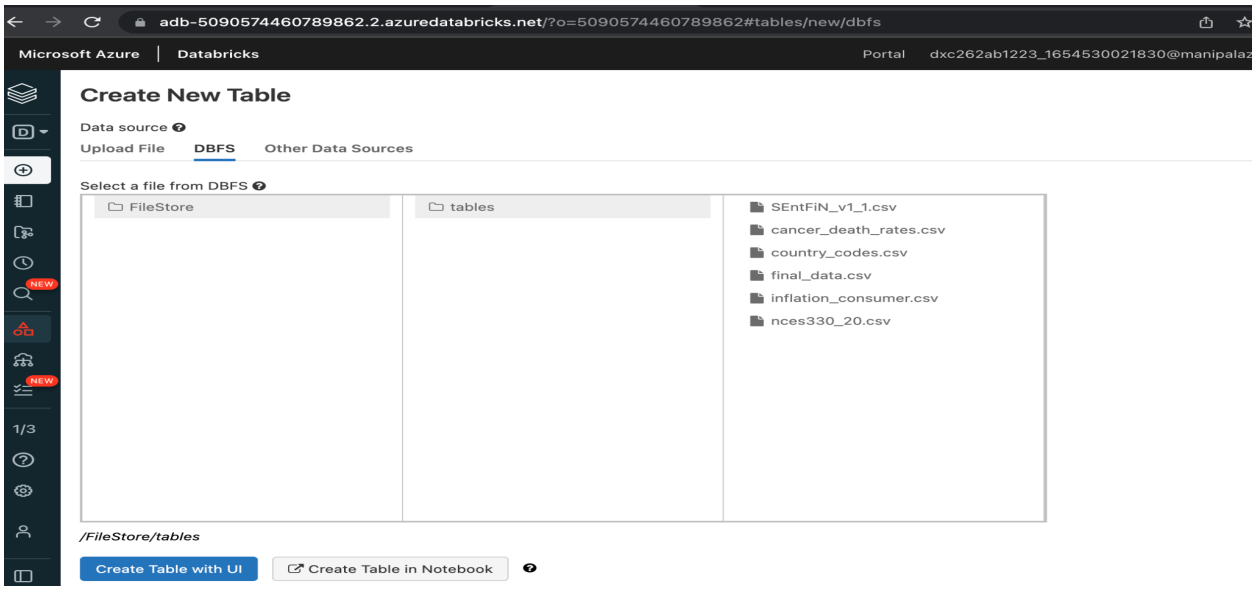


Case 6. Using archive6.zip file - please ingest data into databricks DBFS path & query the data, redesign columns accordingly using dataframe commands - display with notebooks accordingly

Ans: The file archive6.zip contains a csv file named inflation-consumer.csv.

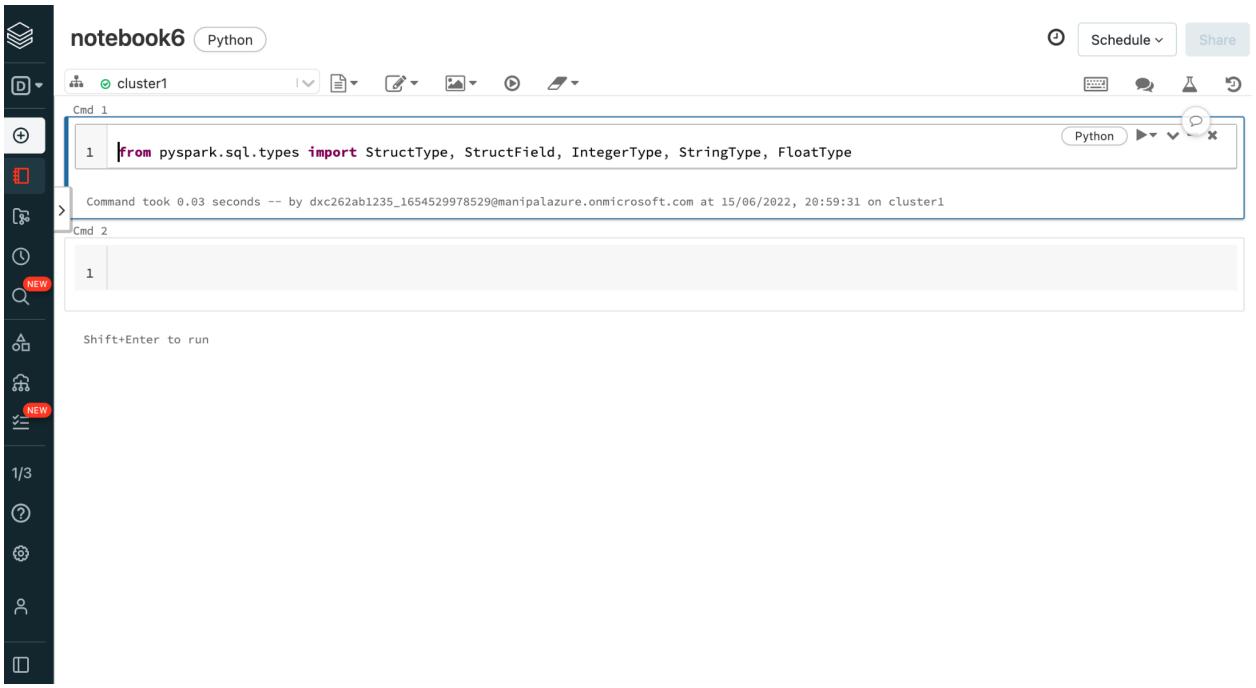
Ingest data into databricks DBFS path.

Click on data then on DBFS and click Filestore and tables and there you can view the table inflation-consumer.csv.



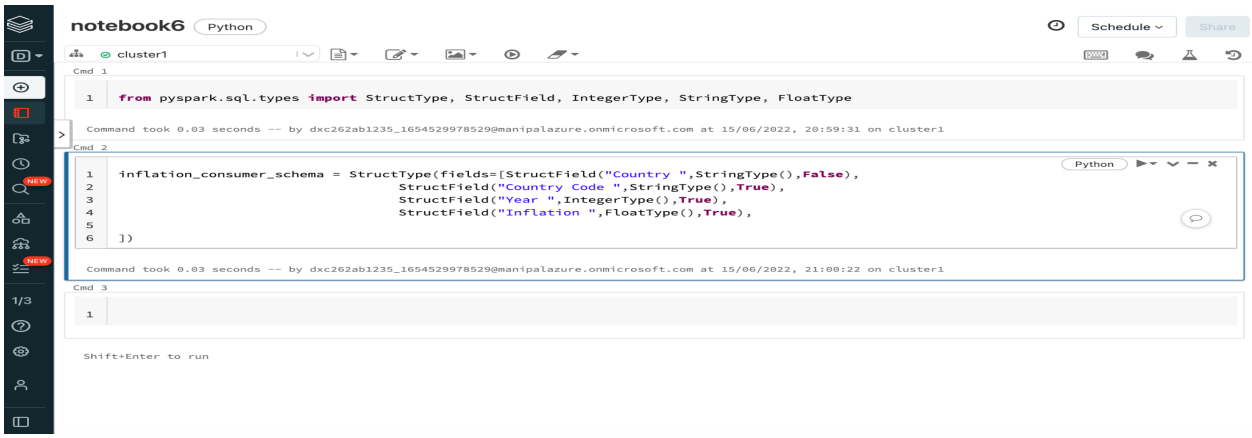
Step 1: Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, FloatType



Step 2:

```
inflation_consumer_schema = StructType(fields=[StructField("Country
",StringType(),False),
                                    StructField("Country Code ",StringType(),True),
                                    StructField("Year ",IntegerType(),True),
                                    StructField("Inflation ",FloatType(),True),

])
```



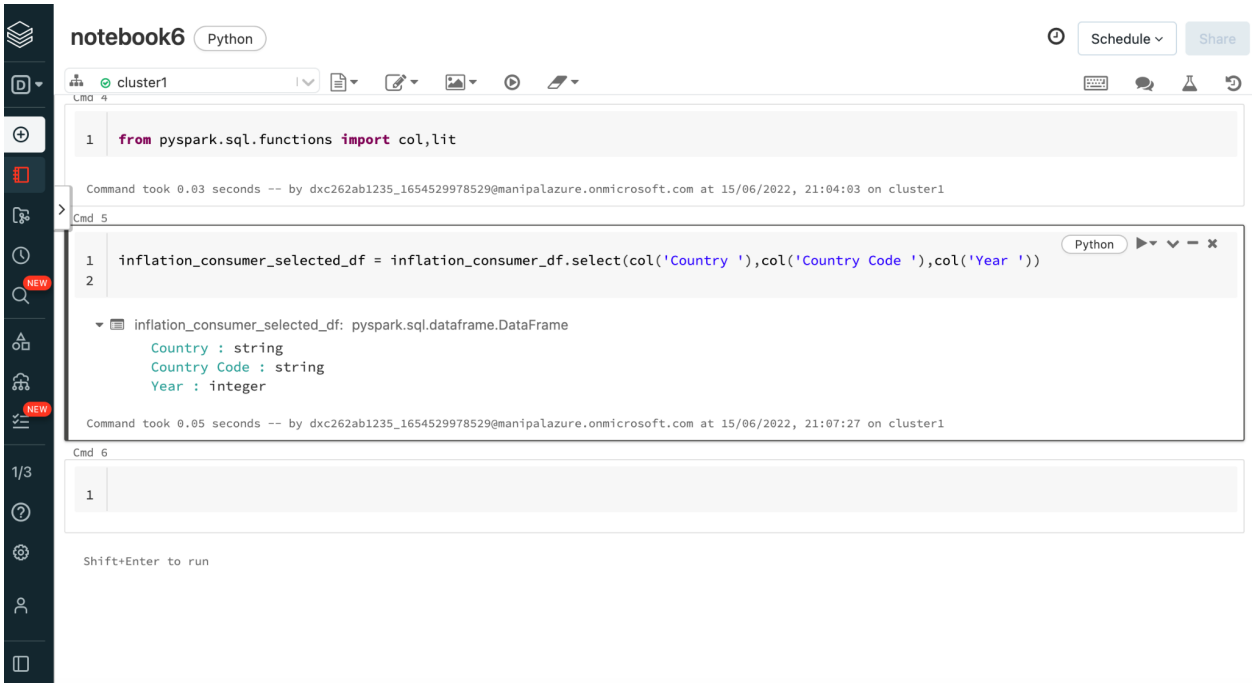
Step 3: Click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

```
inflation_consumer_df = spark.read \
.option("header" , True) \
.schema(inflation_consumer_schema) \
.csv("/FileStore/tables/inflation_consumer.csv")
```

Step 4:

from pyspark.sql.functions import col,lit

inflation_consumer_selected_df = inflation_consumer_df.select(col('Country '),col('Country Code '),col('Year '))



Step 5:

display(inflation_consumer_df)