

NAME: SHAIK ABDUL KHADAR JILANI

ROLLNO: DXC262AB12038

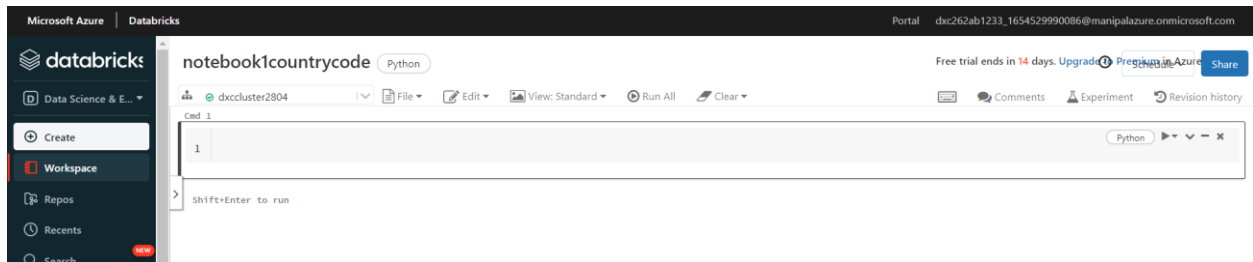
BATCH: DXC-262-Analytics-B12-Azure

SUBMISSION: 15-6-2022

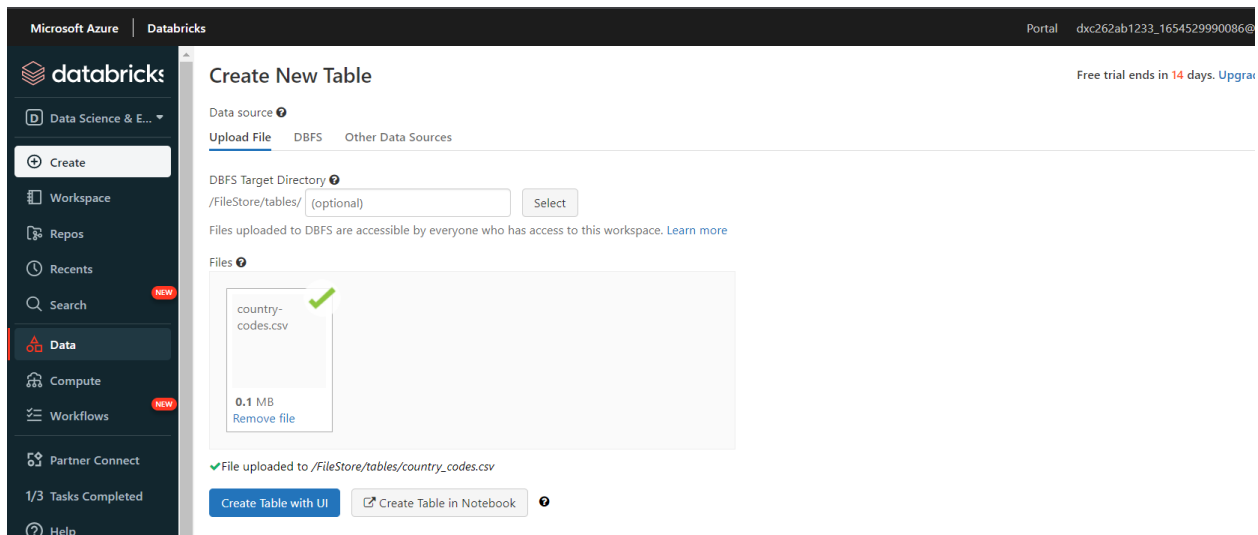
COMPANY: DXC TECHNOLOGY

1) Using archive1.zip file - please ingest data into data bricks DBFS path & query the data, display with notebooks accordingly

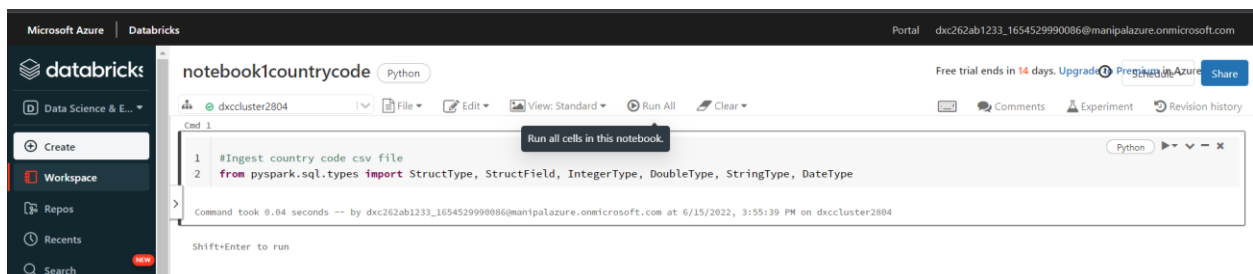
After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook and



Now 1st we need to upload the country code file so go to Data tab and click on create table



Now we need to ingest the county code csv file



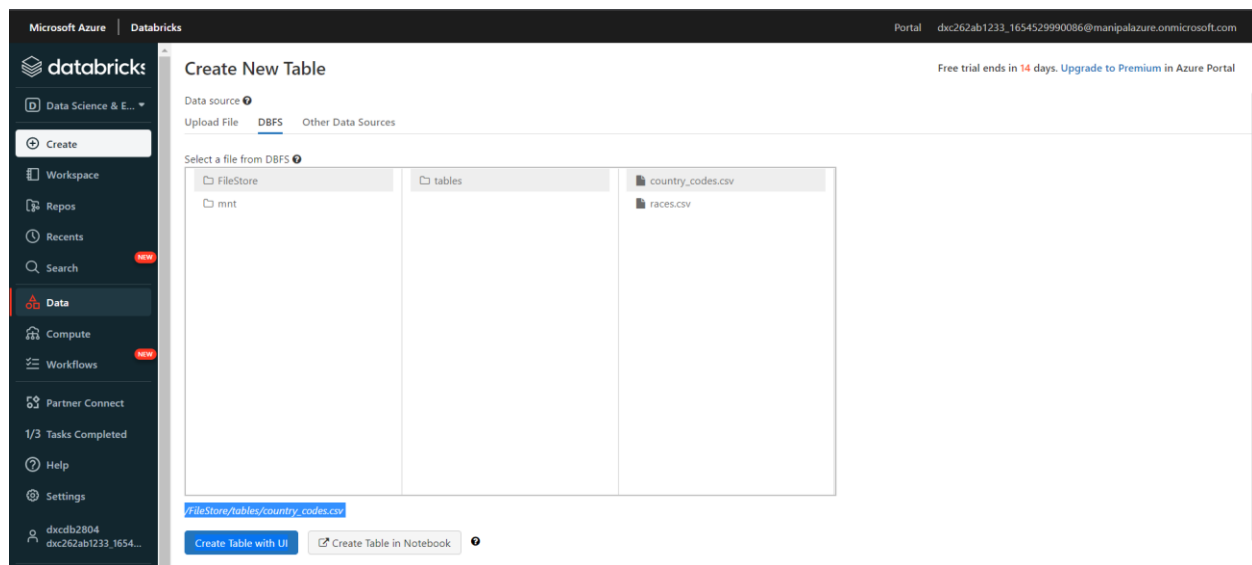
We have included schema

```
Cmd 2
1 #include the schema
2 country_schema = StructType(fields=[StructField("marc",StringType(),True),
3     StructField("Capital",StringType(),True),
4     StructField("M49",IntegerType(),True),
5     StructField("Regioncode",IntegerType(),True),
6
7 ])

Command took 0.03 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 4:04:19 PM on dxcccluster2804

Shift+Enter to run
```

We need to copy the path of the csv file before creating a data frame to do that go to Data click on create table and click on DBFS



We need to create a data frame

```
Cmd 3
1 #creating a data frame
2 country_df = spark.read \
3     .option("header",True) \
4     .schema(country_schema) \
5     .csv("/FileStore/tables/country_codes.csv")

country_df: pyspark.sql.dataframe.DataFrame
  marc: string
  Capital: string
  M49: integer
  Regioncode: integer

Command took 0.14 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 4:09:57 PM on dxcccluster2804

Shift+Enter to run
```

To add ingestion date

```
Cmd 5

1 #add ingestion date to the data frame
2 country_final_df = country_df.withColumn("ingestion_date",current_timestamp())
3

▼ country_final_df: pyspark.sql.dataframe.DataFrame
  marc: string
  Capital: string
  M49: integer
  Regioncode: integer
  ingestion_date: timestamp

Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 4:18:51 PM on dxcluster2804

Shift+Enter to run
```

To rename the columns we need

```
Cmd 6

1 country_renamed_final_df = country_final_df.withColumnRenamed("marc","MARC") \
2 .withColumnRenamed("Capital","CAPITAL") \
3 .withColumnRenamed("M49","m49") \
4 .withColumnRenamed("ingestion_date","INGESTION_DATE")

▼ country_renamed_final_df: pyspark.sql.dataframe.DataFrame
  MARC: string
  CAPITAL: string
  m49: integer
  Regioncode: integer
  INGESTION_DATE: timestamp

Command took 0.05 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 4:23:11 PM on dxcluster2804
```

write the output to processed container in parquet format

```
Cmd 7

1 country_renamed_final_df.write.mode('overwrite').partitionBy('MARC').parquet('/mnt/formulaidl/processed/country')

Cancel ** Running command...
▶ (1) Spark Jobs

Shift+Enter to run
```

To display the data

```
1 display(spark.read.parquet("/mnt/formulaidl/processed/country"))
```




▼ (5) Spark Jobs

- ▶ Job 2 [View](#) (Stages: 1/1)
- ▶ Job 3 [View](#) (Stages: 1/1)
- ▶ Job 4 [View](#) (Stages: 1/1)
- ▶ Job 5 [View](#) (Stages: 1/1)
- ▶ Job 6 [View](#) (Stages: 1/1)

Table **Data Profile**

	CAPITAL	m49	Regioncode	INGESTION_DATE	MARC
1	1	null	null	2022-06-15T10:58:27.936+0000	PUR
2	7	null	null	2022-06-15T10:58:27.936+0000	KAZ
3	7	null	null	2022-06-15T10:58:27.936+0000	RUS
4	1	null	null	2022-06-15T10:58:27.936+0000	CAN
5	1	null	null	2022-06-15T10:58:27.936+0000	USA
6	95	null	null	2022-06-15T10:58:27.936+0000	MYA
7	31	null	null	2022-06-15T10:58:27.936+0000	NFD

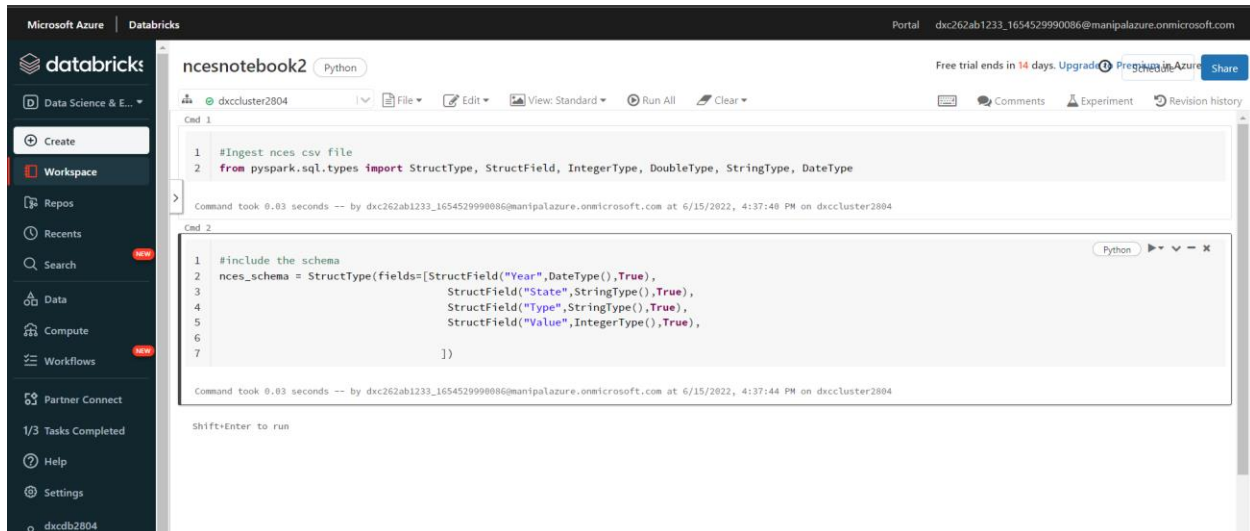
Showing all 250 rows.

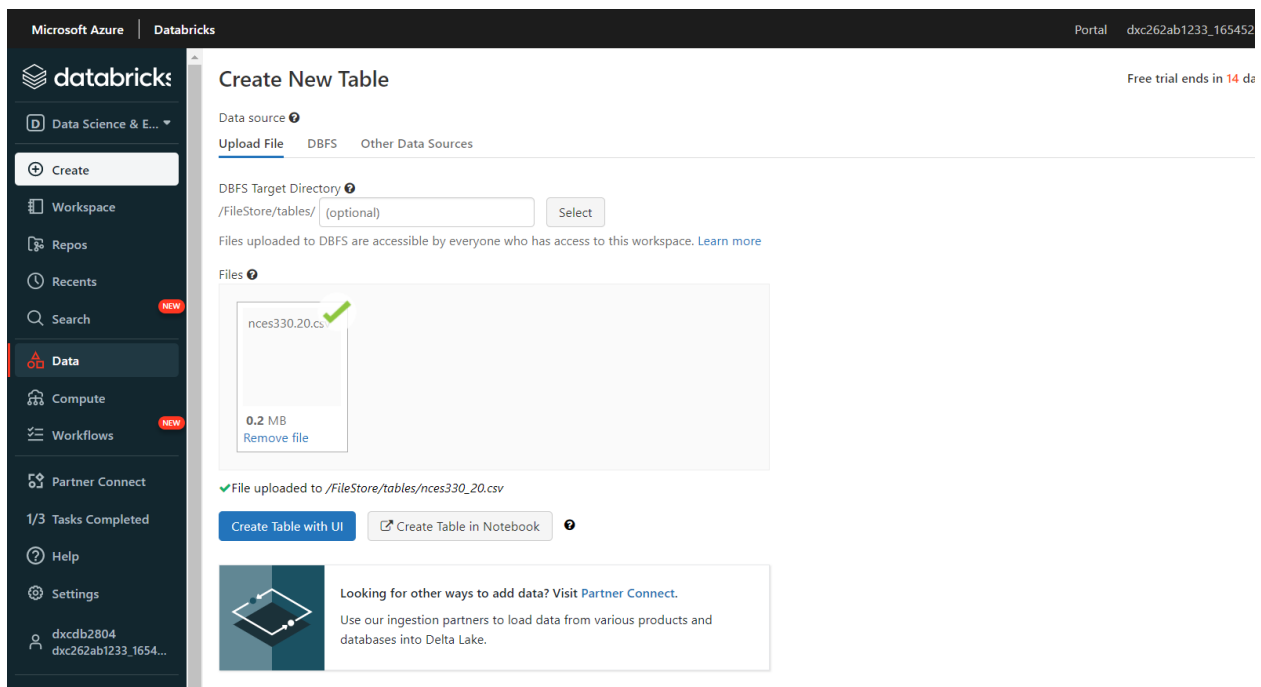
2) Using archive2.zip file - please ingest data into data bricks DBFS path & query the data display with notebooks accordingly

After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook and

Ingest the data and schema is created



To create a data frame we need to upload the file in data create table and upload by drag and drop



Now creating the data frame we need to copy the file path to include in the data frame

```
Cmd 3

1 #creating a data frame
2 nces_df = spark.read \
3   .option("header",True) \
4   .schema(nces_schema) \
5   .csv("/FileStore/tables/nces330_20.csv")

▼ nces_df: pyspark.sql.dataframe.DataFrame
  Year: date
  State: string
  Type: string
  Value: integer

Command took 0.15 seconds -- by dxc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 4:41:53 PM on dxcccluster2804
```

Now ingestion date

The screenshot shows a Databricks notebook interface. The left sidebar contains navigation options like 'Data Science & E...', 'Create', 'Workspace', 'Repos', 'Recents', 'Search', 'Data', 'Compute', 'Workflows', and 'Partner Connect'. The main area displays a Python notebook named 'ncesnotebook2'. The code in the notebook adds a 'current_timestamp' column to the 'nces_df' DataFrame. The output shows the updated DataFrame with an additional 'ingestion_date' column of type 'timestamp'.

```
1 from pyspark.sql.functions import current_timestamp
2
Command took 0.02 seconds -- by dxc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 4:42:49 PM on dxcccluster2804

Cmd 5
1 #add ingestion date to the data frame
2 nces_ingestiondate_df = nces_df.withColumn("ingestion_date",current_timestamp())
3
▼ nces_ingestiondate_df: pyspark.sql.dataframe.DataFrame
  Year: date
  State: string
  Type: string
  Value: integer
  ingestion_date: timestamp

Command took 0.04 seconds -- by dxc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 4:43:52 PM on dxcccluster2804
```

Renaming the selected columns

```
Cmd 6
1 nces_renamed_final_df = nces_ingestiondate_df.withColumnRenamed("Year","YEAR") \
2   .withColumnRenamed("State","STATE") \
3   .withColumnRenamed("Type","What_TYPE") \
4   .withColumnRenamed("ingestion_date","INGESTION_DATE")
▼ nces_renamed_final_df: pyspark.sql.dataframe.DataFrame
  YEAR: date
  STATE: string
  What_TYPE: string
  Value: integer
  INGESTION_DATE: timestamp

Command took 0.04 seconds -- by dxc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 4:46:07 PM on dxcccluster2804
```

To check the job

The screenshot shows the Databricks notebook interface with a job details panel on the right. The notebook code shows a write operation to a Parquet file. The job details panel shows that the job 'nces_renamed_final_df.write.mode('overwrite').p...' has succeeded. Below the job details, a table lists the stages of the job.

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle
7	1875153522092939229	nces_renamed_final_df.write.mode('overwrite').p... parquet at NativeMethodAccessorImpl.java:0 +details	2022/06/15 11:18:03	1 s	1/1	174.6 KIB	7.2 KIB	

To display the data

Cmd 8

```
1 display(spark.read.parquet("/mnt/formulaidl/processed/nces"))
```

(2) Spark Jobs

Table Data Profile

	YEAR	STATE	Value	INGESTION_DATE	What_TYPE
1	2013-01-01	Alabama	null	2022-06-15T11:18:03.704+0000	Public In-State
2	2013-01-01	Alabama	null	2022-06-15T11:18:03.704+0000	Public In-State
3	2013-01-01	Alabama	null	2022-06-15T11:18:03.704+0000	Public In-State
4	2013-01-01	Alaska	null	2022-06-15T11:18:03.704+0000	Public In-State
5	2013-01-01	Alaska	null	2022-06-15T11:18:03.704+0000	Public In-State
6	2013-01-01	Alaska	null	2022-06-15T11:18:03.704+0000	Public In-State
7	2013-01-01	Arizona	null	2022-06-15T11:18:03.704+0000	Public In-State

Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.

3) Using archive3.zip file - please ingest data into data bricks DBFS path & query the data display with notebooks accordingly

After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook

finaldatanotebook3 Python

Free trial ends in 14 days. Upgrade Pro on Azure Share

dxcccluster2804

File Edit View: Standard Run All Clear

Comments Experiment Revision history

Cmd 1

```
1 #Ingest final data csv file
2 from pyspark.sql.types import StructType, StructField, IntegerType, DoubleType, StringType, DateType
```

Command took 0.04 seconds -- by dxcc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 4:58:33 PM on dxcccluster2804

Cmd 2

```
1 #include the schema
2 tweeter_schema = StructType(fields=[StructField("tweet_text",StringType(),True),
3                                     StructField("emotion_in_tweet_is_directed_at",StringType(),True),
4                                     StructField("is_there_an_emotion_directed_at_a_brand_or_product",StringType(),True),
5                                     ])
6
```

Command took 0.03 seconds -- by dxcc262ab1233_165452999086@manipalazure.onmicrosoft.com at 6/15/2022, 5:08:59 PM on dxcccluster2804

Shift+Enter to run

Now to create data frame again go to Data create table and drag and drop the data in it

Microsoft Azure | Databricks

databricks

Data Science & E...

Create

Workspace

Repos

Recents

Search

Data

Compute

Workflows

Partner Connect

1/3 Tasks Completed

Help

Settings

dxccdb2804
dxcc262ab1233_1654...

Create New Table

Data source

Upload File DBFS Other Data Sources

DBFS Target Directory

/FileStore/tables/ (optional) Select

Files uploaded to DBFS are accessible by everyone who has access to this workspace. Learn more

Files

final_data.csv ✓

1.3 MB Remove file

File uploaded to /FileStore/tables/final_data.csv

Create Table with UI Create Table in Notebook

Looking for other ways to add data? Visit Partner Connect.

Use our ingestion partners to load data from various products and databases into Delta Lake.

Creating data frame

```
Cmd 3

1 #creating a data frame
2 tweeter_df = spark.read \
3 .option("header",True) \
4 .schema(tweeter_schema) \
5 .csv("/FileStore/tables/final_data.csv")

▼ tweeter_df: pyspark.sql.dataframe.DataFrame
  tweet_text: string
  emotion_in_tweet_is_directed_at: string
  is_there_an_emotion_directed_at_a_brand_or_product: string

Command took 0.13 seconds -- by dxc262ab1233_1654529990886@manipalazure.onmicrosoft.com at 6/15/2022, 5:05:45 PM on dxcccluster2804
```

Creating ingestion date

```
Cmd 4

1 from pyspark.sql.functions import current_timestamp

Command took 0.03 seconds -- by dxc262ab1233_1654529990886@manipalazure.onmicrosoft.com at 6/15/2022, 5:11:08 PM on dxcccluster2804

Cmd 5

1 #add ingestion date to the data frame
2 tweeter_ingd_df = tweeter_df.withColumn("ingestion_date",current_timestamp())
3

▼ tweeter_ingd_df: pyspark.sql.dataframe.DataFrame
  tweet_text: string
  emotion_in_tweet_is_directed_at: string
  is_there_an_emotion_directed_at_a_brand_or_product: string
  ingestion_date: timestamp
```

Renaming selected columns

```
Cmd 6

1 tweeter_renamed_df = tweeter_ingd_df.withColumnRenamed("tweet_text","tweeted") \
2 .withColumnRenamed("emotion_in_tweet_is_directed_at","Typeof_phone") \
3 .withColumnRenamed("is_there_an_emotion_directed_at_a_brand_or_product","BRAND") \
4 .withColumnRenamed("ingestion_date","INGESTION_DATE")

▼ tweeter_renamed_df: pyspark.sql.dataframe.DataFrame
  tweeted: string
  Typeof_phone: string
  BRAND: string
  INGESTION_DATE: timestamp

Command took 0.04 seconds -- by dxc262ab1233_1654529990886@manipalazure.onmicrosoft.com at 6/15/2022, 5:16:12 PM on dxcccluster2804
```

writing the output to processed container in parquet format and displaying it

Create

Workspace

Repos

Recents

Search

Data

Compute

Workflows

Partner Connect

1/3 Tasks Completed

Help

Settings

dxcdb2804
dxc262ab1233_1654...

Menu options

```
Cmd 7

1 #write the output to processed container in parquet format
2 tweeter_renamed_df.write.mode('overwrite').parquet('/mnt/formulaidl/processed/tweeter')

▼ (1) Spark Jobs
  Job 10 View (Stages: 1/1)

Command took 1.22 seconds -- by dxc262ab1233_1654529990886@manipalazure.onmicrosoft.com at 6/15/2022, 5:18:16 PM on dxcccluster2804

Cmd 8

1 display(spark.read.parquet('/mnt/formulaidl/processed/tweeter'))

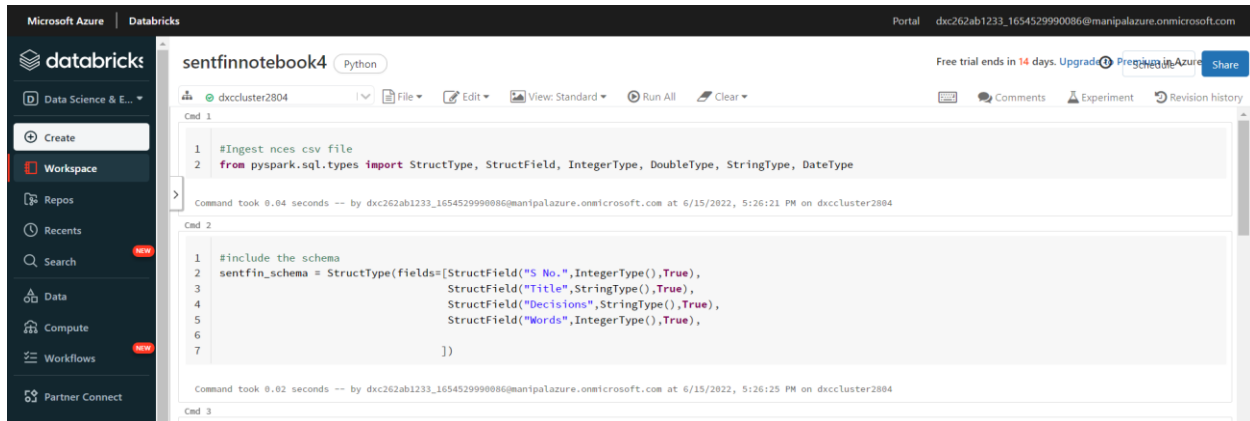
▼ (2) Spark Jobs
```

	tweeted	Typeof_phone	BRAND
1	..@wesley83 I have a 3G iPhone. After 3 hrs tweeting at #Rise_Austin, it was dead! I need to upgrade. Plugin stations at #SXSW.	iPhone	Negative emotion
2	@jessede Know about @fludapp? Awesome iPad/iPhone app that you'll likely appreciate for its design. Also, they're giving free Ts at #SXSW	iPad or iPhone App	Positive emotion
3	@swonderlin Can not wait for #iPad 2 also. They should sale them down at #SXSW.	iPad	Positive emotion
4	@sxsw I hope this year's festival isn't as crashy as this year's iPhone app. #sxsw	iPad or iPhone App	Negative emotion
5	@sxswstate great stuff on Fri #SXSW: Marissa Mayer (Google). Tim O'Reilly (tech books/conferences) & Matt Mullenweg (WordPress)	Google	Positive emotion
6	@teachntech00 New iPad Apps For #SpeechTherapy And Communication Are Showcased At The #SXSW Conference	null	No emotion toward brand or produ

4) Using archive4.zip file - please ingest data into data bricks DBFS path & query the data display with notebooks accordingly

After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook

Ingest the data and giving schema for the data



```
1 #Ingest nces csv file
2 from pyspark.sql.types import StructType, StructField, IntegerType, DoubleType, StringType, DateType

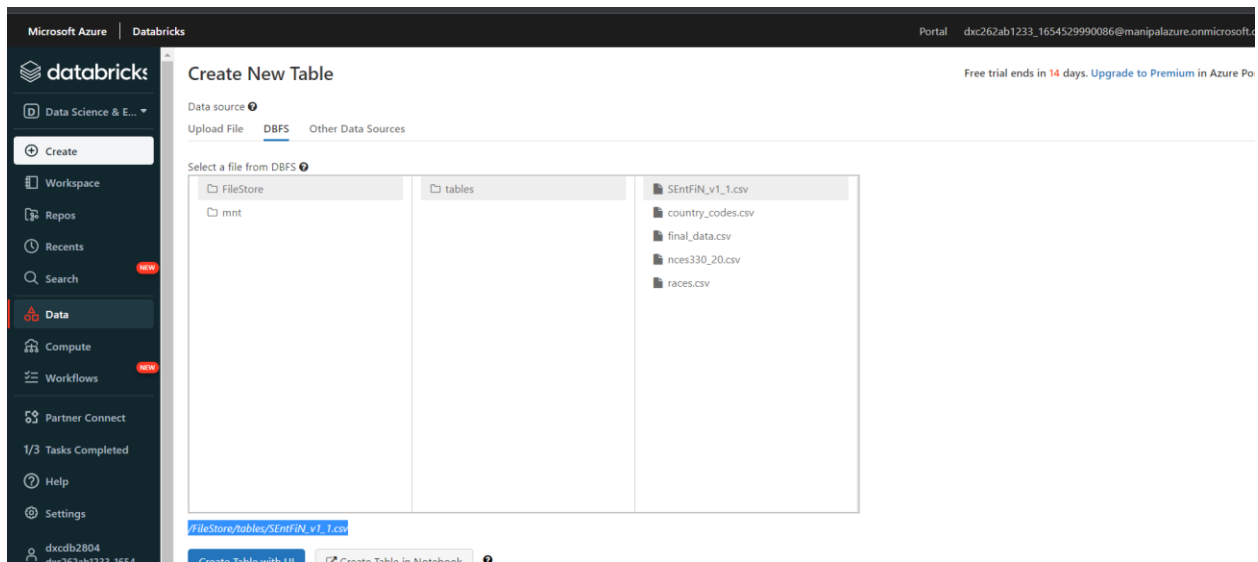
Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:26:21 PM on dxcluster2804

Cnd 2

1 #include the schema
2 sentFin_schema = StructType(fields=[StructField("No.", IntegerType(), True),
3                                     StructField("Title", StringType(), True),
4                                     StructField("Decisions", StringType(), True),
5                                     StructField("Words", IntegerType(), True),
6                                     ])
7

Command took 0.02 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:26:25 PM on dxcluster2804
```

Creating a data frame first upload the file in data create table and drag and drop the file and copy the location and paste in the data frame we create



Creating a data frame

The screenshot shows the Databricks interface for a notebook named 'sentfinnotebook4'. The left sidebar contains navigation options like 'Workspace', 'Repos', 'Recents', 'Search', 'Data', 'Compute', and 'Workflows'. The main area displays a code cell with the following Python code:

```
1 #creating a data frame
2 sentfin_df = spark.read \
3   .option("header",True) \
4   .schema(sentfin_schema) \
5   .csv("/FileStore/tables/SentFIN_v1_1.csv")
```

The output of the code is a DataFrame with the following schema:

```
sentfin_df: pyspark.sql.dataframe.DataFrame
  S No.: integer
  Title: string
  Decisions: string
  Words: integer
```

Command took 0.14 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:28:40 PM on dxcluster2804

Creating ingestion date

The screenshot shows the Databricks interface for the same notebook. The code cell contains the following Python code:

```
1 from pyspark.sql.functions import current_timestamp
2
3 #add ingestion date to the data frame
4 sentfin_ingestiondate_df = sentfin_df.withColumn("ingestion_date",current_timestamp())
```

The output is a DataFrame with the following schema:

```
sentfin_ingestiondate_df: pyspark.sql.dataframe.DataFrame
  S No.: integer
  Title: string
  Decisions: string
  Words: integer
  ingestion_date: timestamp
```

Command took 0.02 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:29:36 PM on dxcluster2804

Renaming the selected columns

The screenshot shows the Databricks interface for the same notebook. The code cell contains the following Python code:

```
1 sentfin_renamed_final_df = sentfin_ingestiondate_df.withColumnRenamed("S No.", "Roll_No") \
2   .withColumnRenamed("Title", "TITLE") \
3   .withColumnRenamed("Words", "WORDS") \
4   .withColumnRenamed("ingestion_date", "INGESTION_DATE")
```

The output is a DataFrame with the following schema:

```
sentfin_renamed_final_df: pyspark.sql.dataframe.DataFrame
  Roll_No: integer
  TITLE: string
  Decisions: string
  WORDS: integer
  INGESTION_DATE: timestamp
```

Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:31:08 PM on dxcluster2804

writing the output to processed container in parquet format and displaying it

The screenshot shows the Databricks interface for the same notebook. The code cell contains the following Python code:

```
1 sentfin_renamed_final_df.write.mode("overwrite").parquet("/mnt/formulaid1/processed/sentfin")
```

Command took 1.23 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:31:52 PM on dxcluster2804

The next code cell contains the following Python code:

```
1 display(spark.read.parquet("/mnt/formulaid1/processed/sentfin"))
```

The output is a table with the following data:

	Roll_No	TITLE	Decisions	WORDS
1	1	SpiceJet to issue 6.4 crore warrants to promoters	["SpiceJet"; "neutral"]	8
2	2	MMTC Q2 net loss at Rs 10.4 crore	["MMTC"; "neutral"]	8
3	3	Mid-cap funds can deliver more, stay put: Experts	["Mid-cap funds"; "positive"]	8
4	4	Mid caps now turn into market darlings	["Mid caps"; "positive"]	7
5	5	Market seeing patience, if not conviction: Prakash Diwan	["Market"; "neutral"]	8
6	6	Infosys: Will the strong volume growth sustain?	["Infosys"; "neutral"]	7
7	7	Hudco raises Rs 279 cr via tax-free bonds	["Hudco"; "positive"]	8

Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.

5) Using archive5.zip file - please ingest data into data bricks DBFS path & query the data display with notebooks accordingly

After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook

Ingest and including schema

```
1 #Ingest nces csv file
2 from pyspark.sql.types import StructType, StructField, IntegerType, DoubleType, StringType, DateType

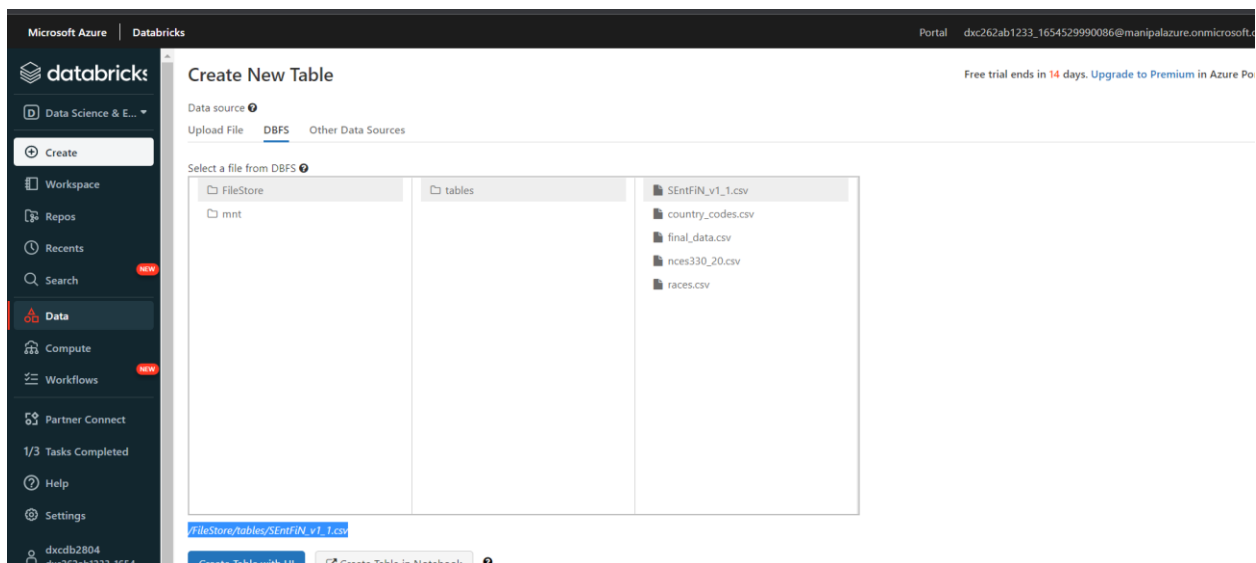
Command took 0.03 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:43:31 PM on dxcluster2804

Cmd 2

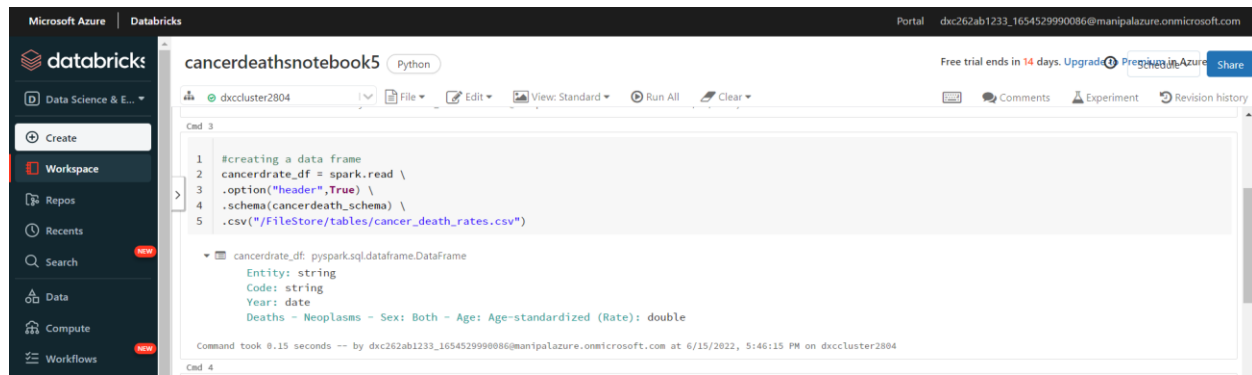
1 #include the schema
2 cancerdeath_schema = StructType(fields=[StructField("Entity",StringType(),True),
3                                           StructField("Code",StringType(),True),
4                                           StructField("Year",DateType(),True),
5                                           StructField("Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate)",DoubleType(),True),
6                                           ])
7
```

Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:46:05 PM on dxcluster2804

Creating a data frame first upload the file in data create table and drag and drop the file and copy the location and paste in the data frame we create



Creating data frame where we need to paste the location of the file



The screenshot shows a Databricks notebook titled 'cancerdeathsnotebook5'. The code in the cell reads a CSV file from the FileStore. The output shows a DataFrame with columns: Entity (string), Code (string), Year (date), and Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate) (double).

```
1 #creating a data frame
2 cancerdate_df = spark.read \
3   .option("header",True) \
4   .schema(cancerdeath_schema) \
5   .csv("/FileStore/tables/cancer_death_rates.csv")
```

▼ cancerdate_df: pyspark.sql.dataframe.DataFrame

- Entity: string
- Code: string
- Year: date
- Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate): double

Command took 0.15 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:46:15 PM on dxcluster2804

Creating ingestion date



The screenshot shows the next step in the notebook. The code imports the current_timestamp function and adds it as a new column to the DataFrame.

```
1 from pyspark.sql.functions import current_timestamp
```

Command took 0.03 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:46:39 PM on dxcluster2804

```
2 #add ingestion date to the data frame
3 cancerdr_ingestiondate_df = cancerdate_df.withColumn("ingestion_date",current_timestamp())
```

▼ cancerdr_ingestiondate_df: pyspark.sql.dataframe.DataFrame

- Entity: string
- Code: string
- Year: date
- Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate): double
- ingestion_date: timestamp

Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:47:29 PM on dxcluster2804

Renaming the selected columns



The screenshot shows the third step in the notebook. The code renames the columns of the DataFrame to match the final schema.

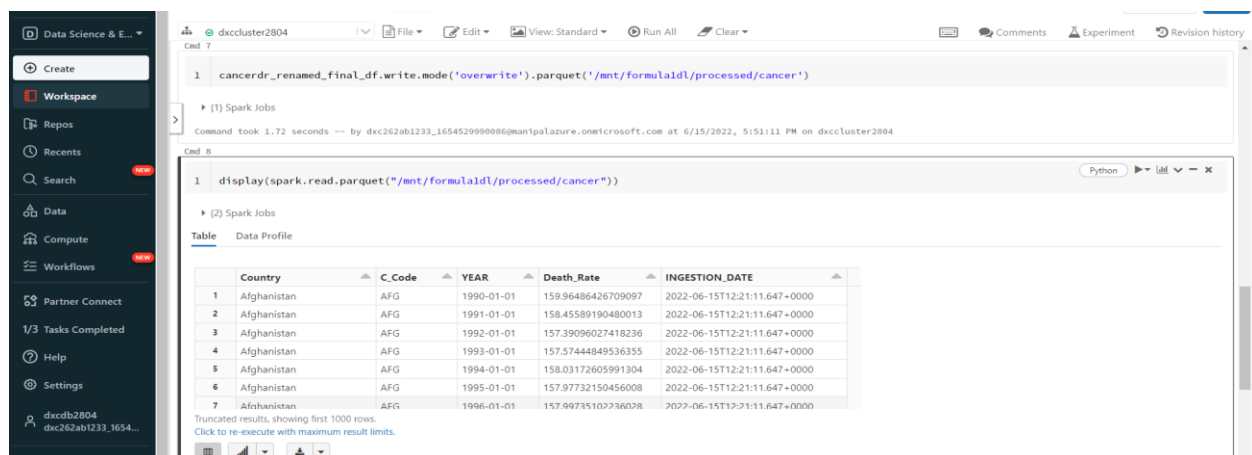
```
1 cancerdr_renamed_final_df = cancerdr_ingestiondate_df.withColumnRenamed("Entity","Country") \
2   .withColumnRenamed("Code","C_Code") \
3   .withColumnRenamed("Year","YEAR") \
4   .withColumnRenamed("ingestion_date","INGESTION_DATE") \
5   .withColumnRenamed("Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate)","Death_Rate")
```

▼ cancerdr_renamed_final_df: pyspark.sql.dataframe.DataFrame

- Country: string
- C_Code: string
- YEAR: date
- Death_Rate: double
- INGESTION_DATE: timestamp

Command took 0.04 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:50:17 PM on dxcluster2804

writing the output to processed container in parquet format and displaying it



The screenshot shows the final steps in the notebook. The code writes the DataFrame to a Parquet file and then displays it.

```
1 cancerdr_renamed_final_df.write.mode("overwrite").parquet("/mnt/formulaidl/processed/cancer")
```

► (1) Spark Jobs

Command took 1.72 seconds -- by dxc262ab1233_1654529990086@manipalazure.onmicrosoft.com at 6/15/2022, 5:51:11 PM on dxcluster2804

```
2 display(spark.read.parquet("/mnt/formulaidl/processed/cancer"))
```

► (2) Spark Jobs

Table Data Profile

	Country	C_Code	YEAR	Death_Rate	INGESTION_DATE
1	Afghanistan	AFG	1990-01-01	159.96486426709097	2022-06-15T12:21:11.647+0000
2	Afghanistan	AFG	1991-01-01	158.45589190480013	2022-06-15T12:21:11.647+0000
3	Afghanistan	AFG	1992-01-01	157.39096027418236	2022-06-15T12:21:11.647+0000
4	Afghanistan	AFG	1993-01-01	157.57444849536355	2022-06-15T12:21:11.647+0000
5	Afghanistan	AFG	1994-01-01	158.03172605991304	2022-06-15T12:21:11.647+0000
6	Afghanistan	AFG	1995-01-01	157.97732150456008	2022-06-15T12:21:11.647+0000
7	Afghanistan	AFG	1996-01-01	157.98735102736028	2022-06-15T12:21:11.647+0000

Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.

6) Using archive6.zip file - please ingest data into data bricks DBFS path & query the data display with notebooks accordingly

After creating the data bricks come to launch workspace and in that create a cluster and go to workspace and create a dataanalytics_project in that create a notebook

Ingest and including schema

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

```
inflation_schema = StructType(fields=[StructField("Country", StringType(), True),
```

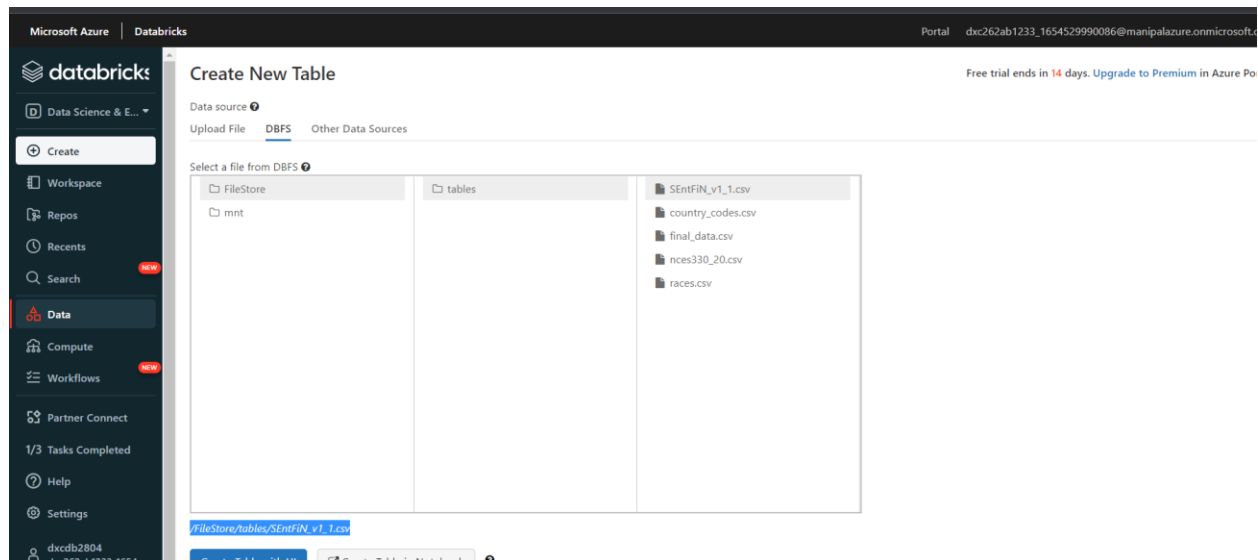
```
    StructField("Country Code",StringType(),True),
```

```
    StructField("Year",IntegerType(),True),
```

```
    StructField("Inflation",DoubleType(),True),
```

```
])
```

For Creating a data frame first upload the file in data create table and drag and drop the file and copy the location and paste in the data frame we create



Creating data frame

```
inflation_df = spark.read \
```

```
.option("header", True) \
```

```
.schema(inflation_schema) \
```

```
.csv("/FileStore/tables/inflation-gdp.csv")
```

```
from pyspark.sql.functions import current_timestamp
```

#add ingestion date to the data frame

```
inflation_final_df = inflation_df.withColumn("ingestion_date",current_timestamp())
```

Renaming the selected columns

```
inflationgdp_renamed_final_df =  
inflation_final_df.withColumnRenamed("Country","COUNTRY") \  
.  
withColumnRenamed("Country Code","COUNTRY_CODE") \  
.  
withColumnRenamed("Year","IN_THEYEAR") \  
.  
withColumnRenamed("ingestion_date","INGESTION_DATE")  
.  
withColumnRenamed("Inflation","Inflation_gdb_ratio")
```

writing the output to processed container in parquet format and displaying it

```
inflationgdp_renamed_final_df.write.mode('overwrite').partitionBy('IN_THEYEAR').parquet('/  
mnt/formula1dl/processed/inflation')
```

displaying it

```
display(spark.read.parquet("/mnt/formula1dl/processed/inflation"))
```