ASSIGNMENT-10

(15th JUNE 2022)

NAME – MOHANA LIKHITHA THOTAKURA ROLLNO – DXC-262AB-1219

BATCH – DXC-262-ANALYTICS-B12-AZURE COMPANY – DXC TECHNOLOGY

EMPLOYEE DOMAIN – AZURE ANALYTICS TRAINER NAME – MR. AJAY KUMAR

DATE OF SUBMISSION – 15th JUNE 2022 NO. OF QUESTIONS: 6

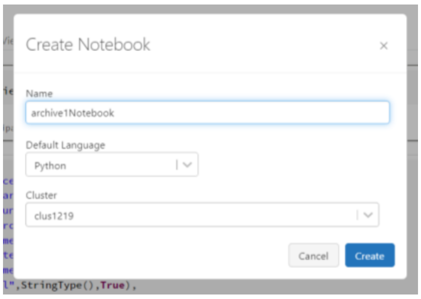
Case 1. Using archive1.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

File Being used: country_codes.csv

Step 1: First, login to your Azure Portal and create a Databricks workspace.

Step 2: Open the Databricks workspace and create cluster for your future use.

Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.



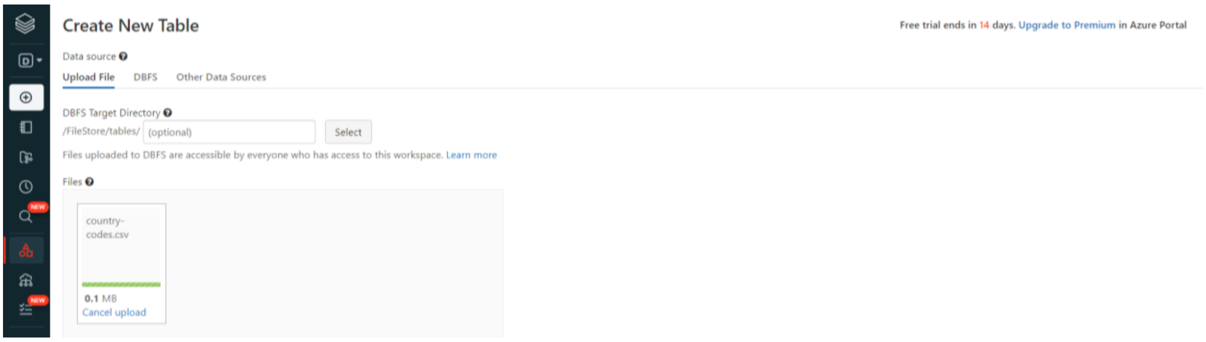
Create Notebook

Name: archive1Notebook

Default Language: Python

Cluster: clus1219

Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

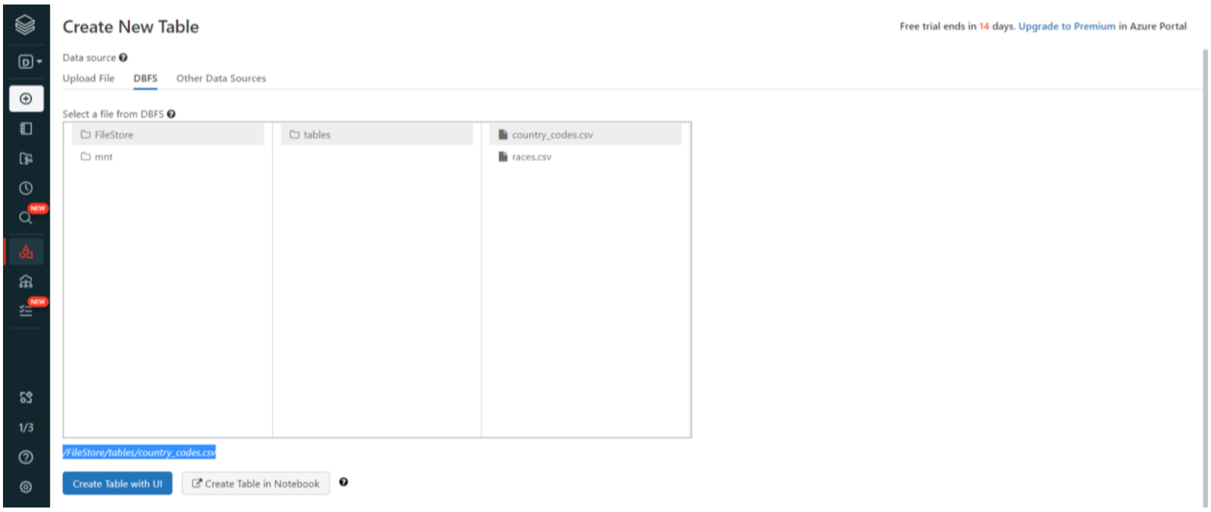
Data source: Upload File

DBFS Target Directory: /FileStore/tables/ (optional)

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

Files: country-codes.csv 0.1 MB

Later, click on DBFS and select the file that you have dropped. This will give you the file path and copy that.



Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

Data source

Upload File | DBFS | Other Data Sources

Select a file from DBFS

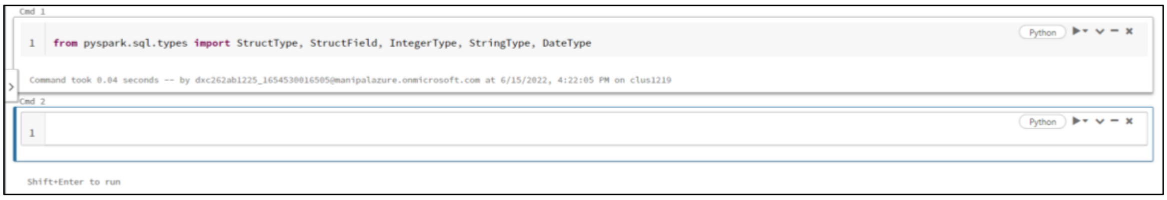
- FileStore
- mnt

- tables

- country_codes.csv
- races.csv

/FileStore/tables/country_codes.csv

Step-5: Import the required fields and features from pyspark.
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType



Cmd 1

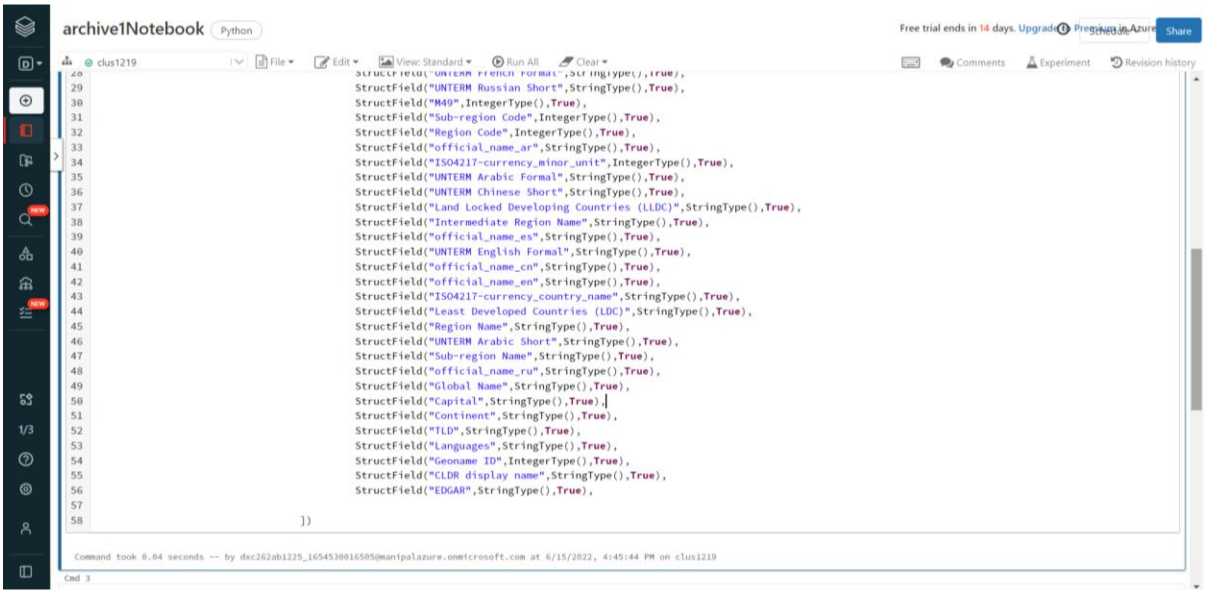
```
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType
```

Command took 0.04 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 4:22:05 PM on clus1219

Cmd 2

Shift+Enter to run

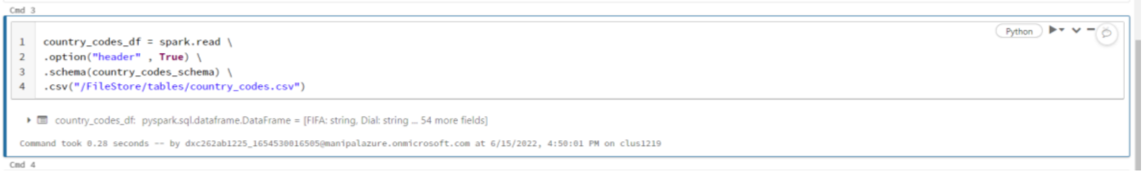
```
country_codes_schema = StructType(fields=[StructField("FIFA", StringType(),False),
                                  StructField("Dial", StringType(),True),
                                  StructField("ISO3166-1-Alpha-3",StringType(),True),
                                  StructField("MARC", 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),

                              ])
```



```
                    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),

                ])
```

```
Command took 0.04 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 4:45:44 PM on clus1219
```

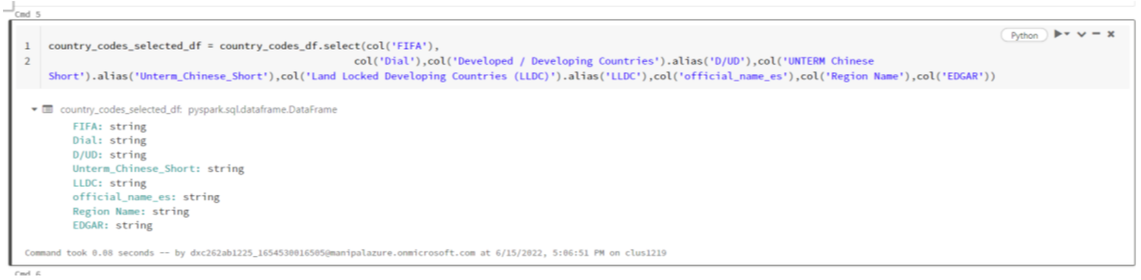
```
country_codes_df = spark.read \
.option("header" , True) \
.schema(country_codes_schema) \
.csv("/FileStore/tables/country_codes.csv")
```



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

```
country_codes_df:  pyspark.sql.dataframe.DataFrame = [FIFA: string, Dial: string ... 54 more fields]
Command took 0.28 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 4:50:01 PM on clus1219
```

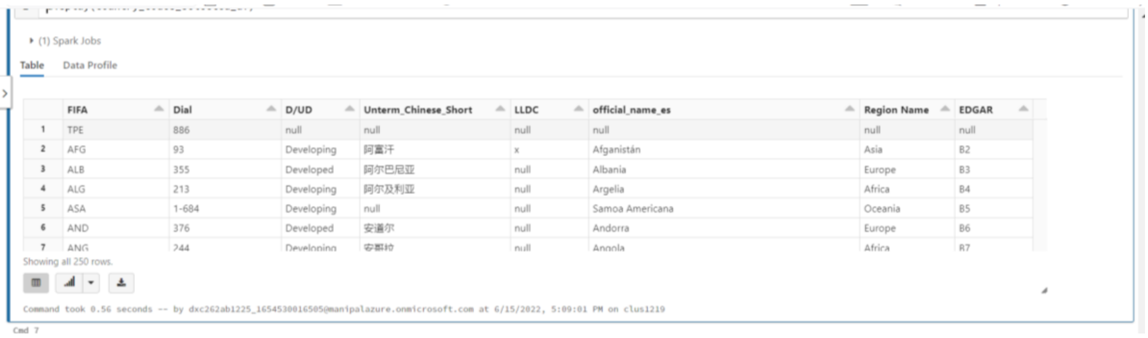
```
from pyspark.sql.functions import col,lit

country_codes_selected_df = country_codes_df.select(col('FIFA'),
                                                    col('Dial'),col('Developed / Developing 
Countries').alias('D/UD'),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'))
```



```
country_codes_selected_df = country_codes_df.select(col('FIFA'),
                                                    col('Dial'),col('Developed / Developing Countries').alias('D/UD'),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'))
```

```
country_codes_selected_df:  pyspark.sql.dataframe.DataFrame
    FIFA: string
    Dial: string
    D/UD: string
    Unterm_Chinese_Short: string
    LLDC: string
    official_name_es: string
    Region Name: string
    EDGAR: string

Command took 0.08 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:06:51 PM on clus1219
```

display(country_codes_selected_df)



	FIFA	Dial	D/UD	Unterm_Chinese_Short	LLDC	official_name_es	Region Name	EDGAR
1	TPE	886	null	null	null	null	null	null
2	AFG	93	Developing	阿富汗	x	Afganistán	Asia	B2
3	ALB	355	Developed	阿尔巴尼亚	null	Albania	Europe	B3
4	ALG	213	Developing	阿尔及利亚	null	Argelia	Africa	B4
5	ASA	1-684	Developing	null	null	Samoa Americana	Oceania	B5
6	AND	376	Developed	安道尔	null	Andorra	Europe	B6
7	ANG	244	Developing	安哥拉	null	Angola	Africa	B7

Showing all 250 rows.

```
Command took 0.56 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:09:01 PM on clus1219
Cmd 7
```

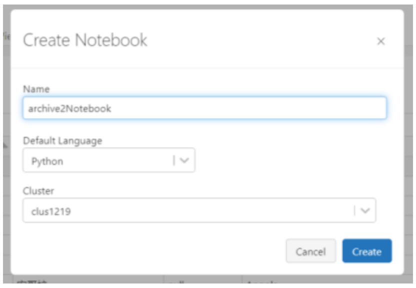
Case 2. Using archive2.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

File Being Used: nces330.20.csv

Step 1: First, login to your Azure Portal and create a Databricks workspace.

Step 2: Open the Databricks workspace and create cluster for your future use.

Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.

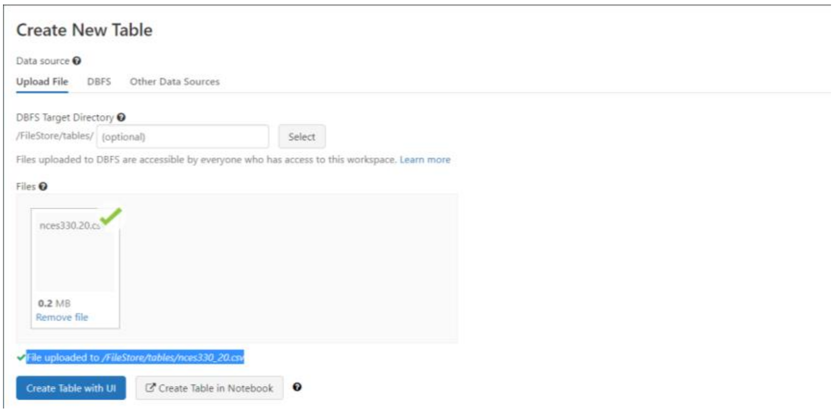


Name: archive2Notebook

Default Language: Python

Cluster: clus1219

Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



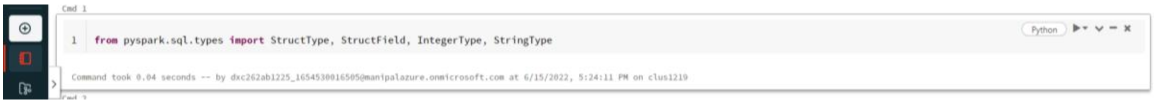
DBFS Target Directory: /FileStore/tables/ (optional)

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

nces330.20.csv
0.2 MB

File uploaded to /FileStore/tables/nces330_20.csv

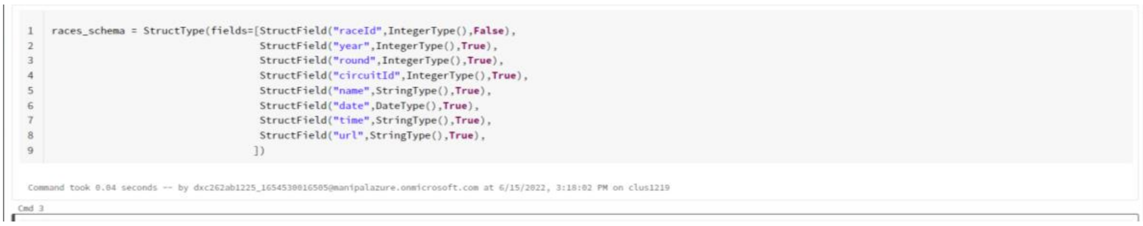
```
from pyspark.sql.types import StructType, StructField, IntegerType, StringType
```



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

```
Command took 0.04 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:24:11 PM on clus1219
```

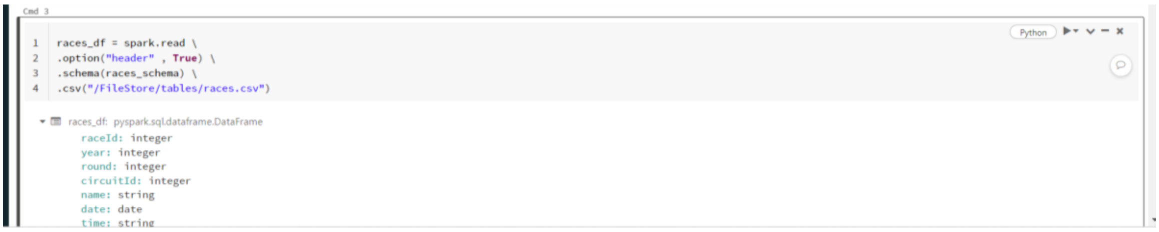
```
nces330_20_schema = StructType(fields=[StructField("year",IntegerType(),False),
                              StructField("State",StringType(),True),
                              StructField("Type",StringType(),True),
                              StructField("Length",StringType(),True),
      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



```
races_schema = StructType(fields=[StructField("raceId",IntegerType(),False),
                                  StructField("year",IntegerType(),True),
                                  StructField("round",IntegerType(),True),
                                  StructField("circuitId",IntegerType(),True),
                                  StructField("name",StringType(),True),
                                  StructField("date",DateType(),True),
                                  StructField("time",StringType(),True),
                                  StructField("url",StringType(),True),
                                 ])
```

```
Command took 0.04 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 3:18:02 PM on clus1219
```

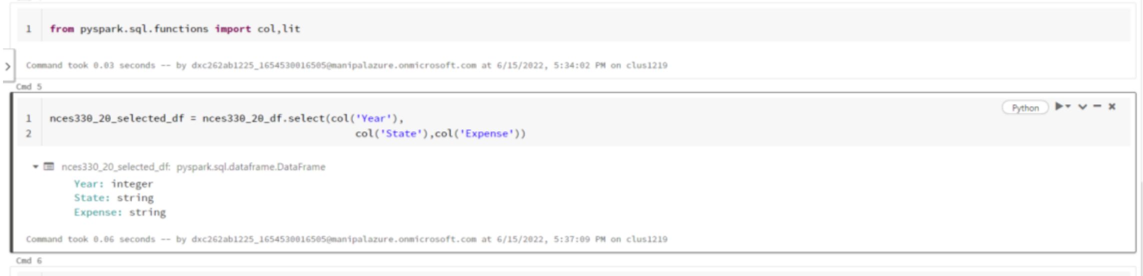
```
nces330_20_df = spark.read \
.option("header" , True) \
.schema(nces330_20_schema) \
.csv("/FileStore/tables/nces330_20.csv")
```



```
races_df = spark.read \
.option("header" , True) \
.schema(races_schema) \
.csv("/FileStore/tables/races.csv")
```

```
races_df:  pyspark.sql.dataframe.DataFrame
    raceId: integer
    year: integer
    round: integer
    circuitId: integer
    name: string
    date: date
    time: string
```

```
from pyspark.sql.functions import col,lit

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



```
from pyspark.sql.functions import col,lit
```

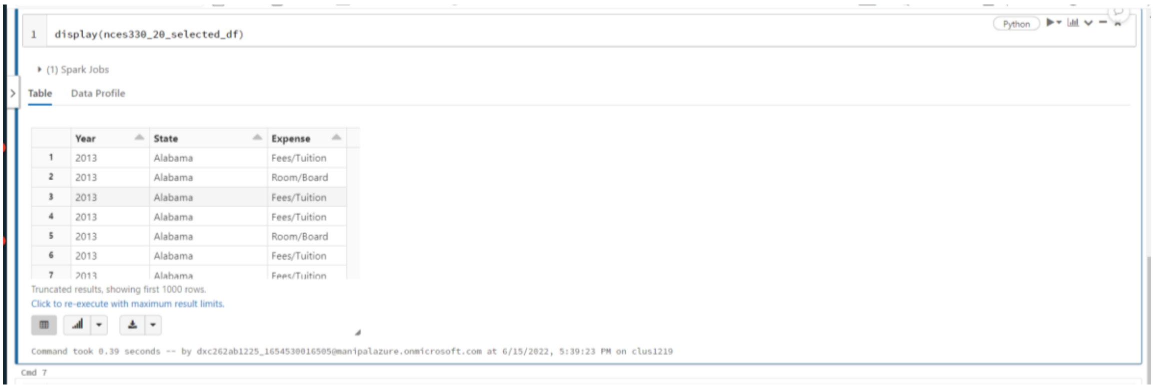
```
Command took 0.03 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:34:02 PM on clus1219
```

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

```
nces330_20_selected_df:  pyspark.sql.dataframe.DataFrame
    Year: integer
    State: string
    Expense: string
```

```
Command took 0.06 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:37:09 PM on clus1219
```

display(nces330_20_selected_df)



```
display(nces330_20_selected_df)
```

(1) Spark Jobs

	Year	State	Expense
1	2013	Alabama	Fees/Tuition
2	2013	Alabama	Room/Board
3	2013	Alabama	Fees/Tuition
4	2013	Alabama	Fees/Tuition
5	2013	Alabama	Room/Board
6	2013	Alabama	Fees/Tuition
7	2013	Alabama	Fees/Tuition

Truncated results, showing first 1000 rows.

Click to re-execute with maximum result limits.

```
Command took 0.39 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:39:23 PM on clus1219
```

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.

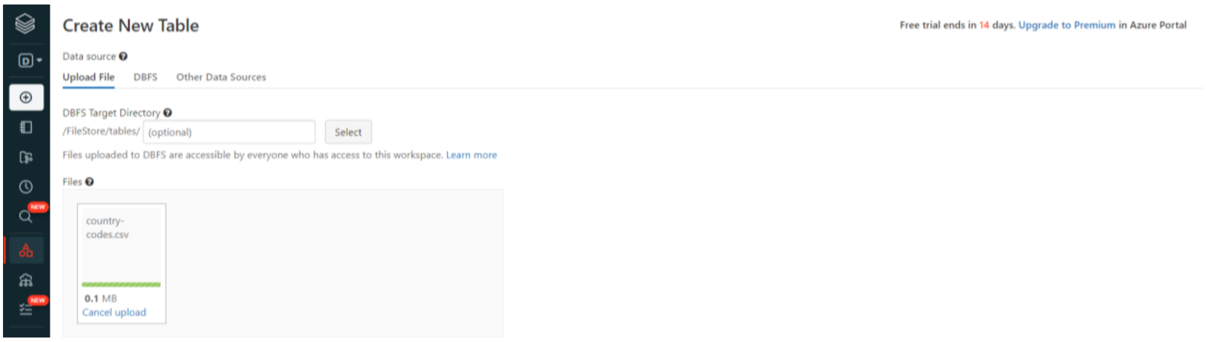
File Being used: final_data.csv

Step 1: First, login to your Azure Portal and create a Databricks workspace.

Step 2: Open the Databricks workspace and create cluster for your future use.

Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.

Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

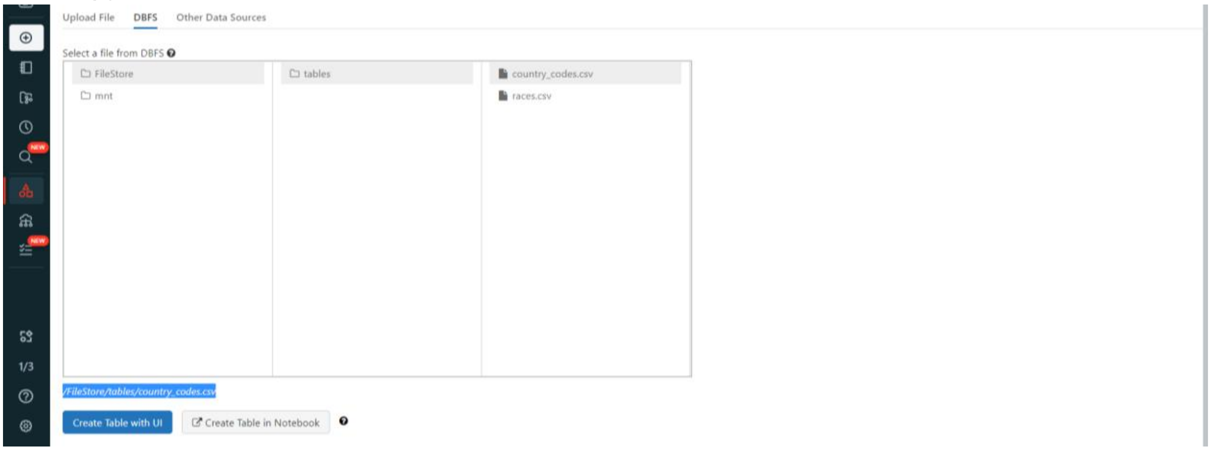
Data source

DBFS Target Directory: /FileStore/tables/ (optional)

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

Files: country-codes.csv 0.1 MB Cancel upload

Later, click on DBFS and select the file that you have dropped. This will give you the file path and copy that.

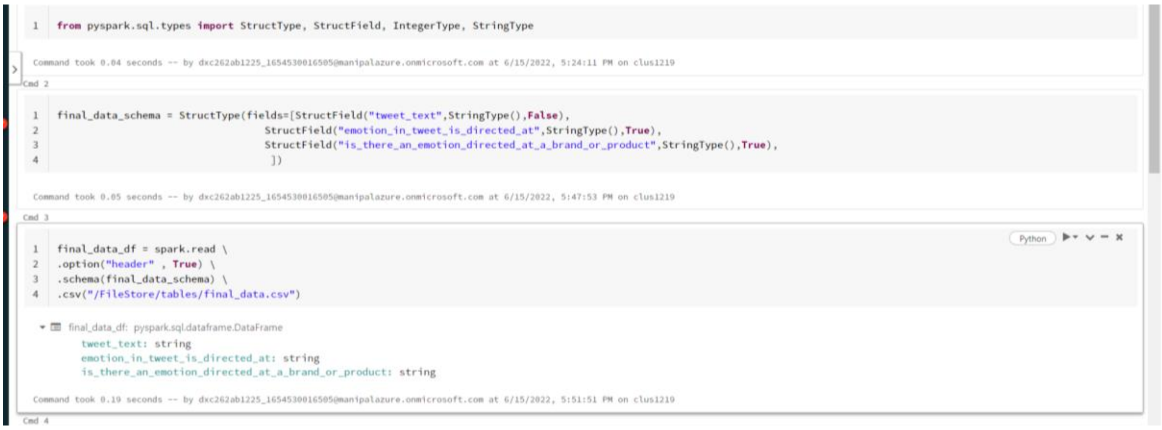


Select a file from DBFS

/FileStore/tables/country_codes.csv

Step-5: Import the required fields and features from pyspark.

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

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),
                                 ])


final_data_df = spark.read \
.option("header" , True) \
.schema(nces330_20_schema) \
.csv("/FileStore/tables/final_data.csv")
```



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

Command took 0.04 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:24:11 PM on clus1219

Cmd 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),
                                 ])
```

Command took 0.05 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:47:53 PM on clus1219

Cmd 3

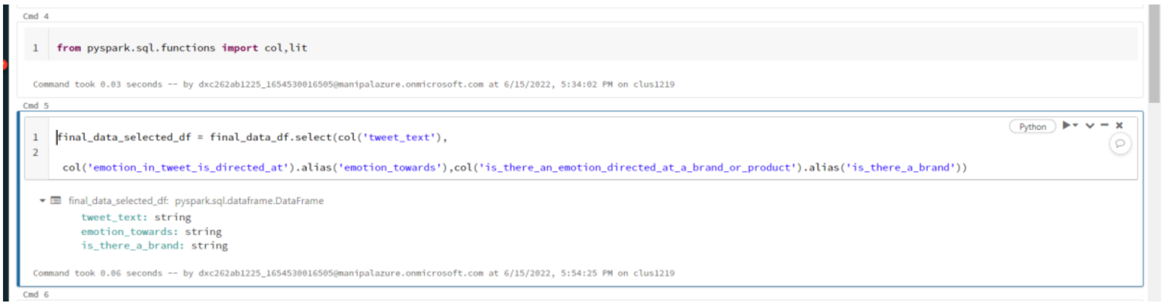
```
final_data_df = spark.read \
.option("header" , True) \
.schema(final_data_schema) \
.csv("/FileStore/tables/final_data.csv")
```

```
final_data_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 0.19 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:51:51 PM on clus1219

Cmd 4

```
from pyspark.sql.functions import col,lit

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'))
```



Cmd 4

```
from pyspark.sql.functions import col,lit
```

Command took 0.03 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:34:02 PM on clus1219

Cmd 5

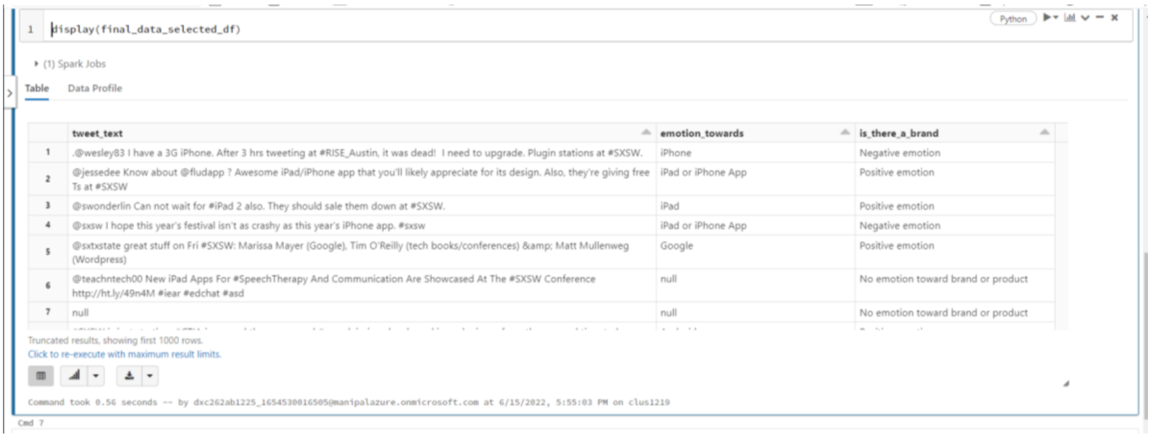
```
final_data_selected_df = final_data_df.select(col('tweet_text'),

 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'))
```

```
final_data_selected_df: pyspark.sql.dataframe.DataFrame
    tweet_text: string
    emotion_towards: string
    is_there_a_brand: string
```

Command took 0.06 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:54:25 PM on clus1219

Cmd 6

display(final_data_selected_df)



```
display(final_data_selected_df)
```

▸ (1) Spark Jobs

	tweet_text	emotion_towards	is_there_a_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	@jessedee 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	@sxtxstate great stuff on Fri #SXSW: Marissa Mayer (Google), Tim O'Reilly (tech books/conferences) &amp; Matt Mullenweg (Wordpress)	Google	Positive emotion
6	@teachntech00 New iPad Apps For #SpeechTherapy And Communication Are Showcased At The #SXSW Conference http://ht.ly/49n4M #iear #edchat #asd	null	No emotion toward brand or product
7	null	null	No emotion toward brand or product

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

```
Command took 0.56 seconds -- by dxc262ab1225_1654530016505@manipalazure.onmicrosoft.com at 6/15/2022, 5:55:03 PM on clus1219
```

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.

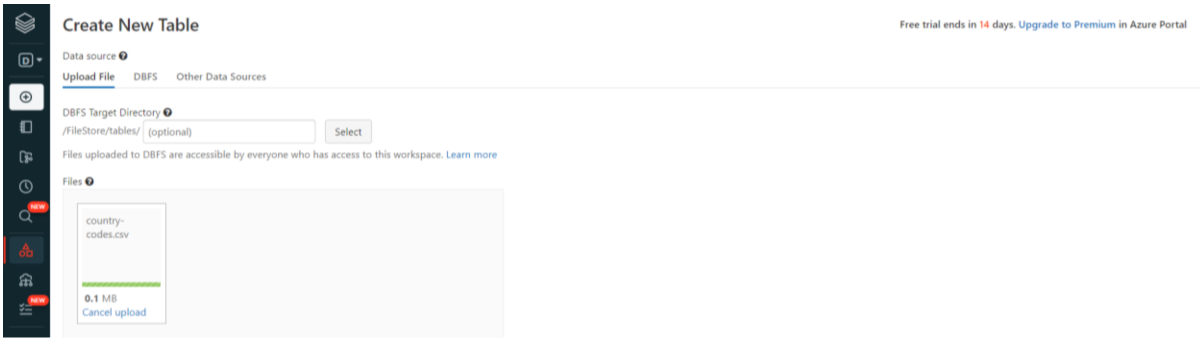
File Being used: SEntFiN-v1.1.csv

Step 1: First, login to your Azure Portal and create a Databricks workspace.

Step 2: Open the Databricks workspace and create cluster for your future use.

Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.

Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



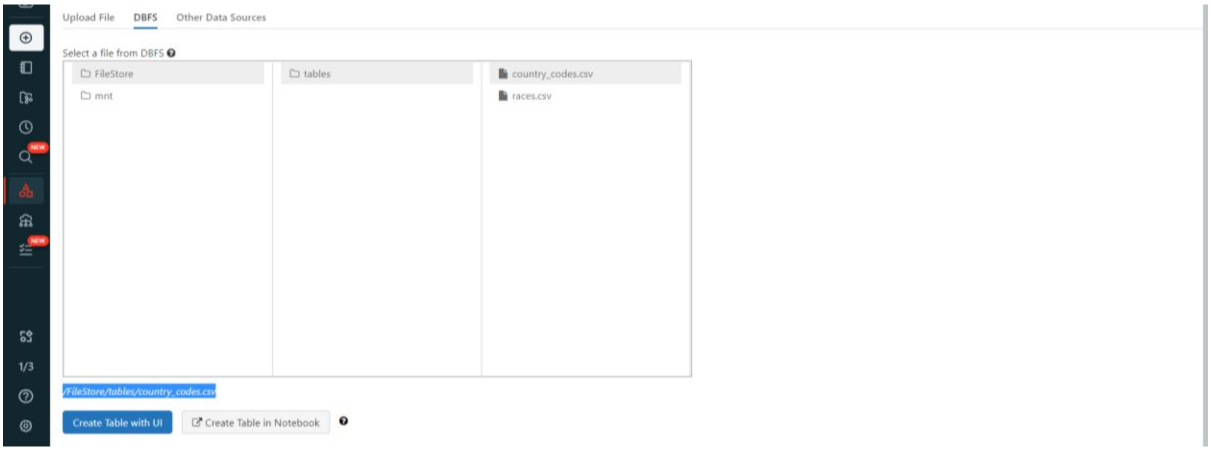
Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

DBFS Target Directory
/FileStore/tables/ (optional) Select

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

Later, click on DBFS and select the file that you have dropped. This will give you the file path and copy that.



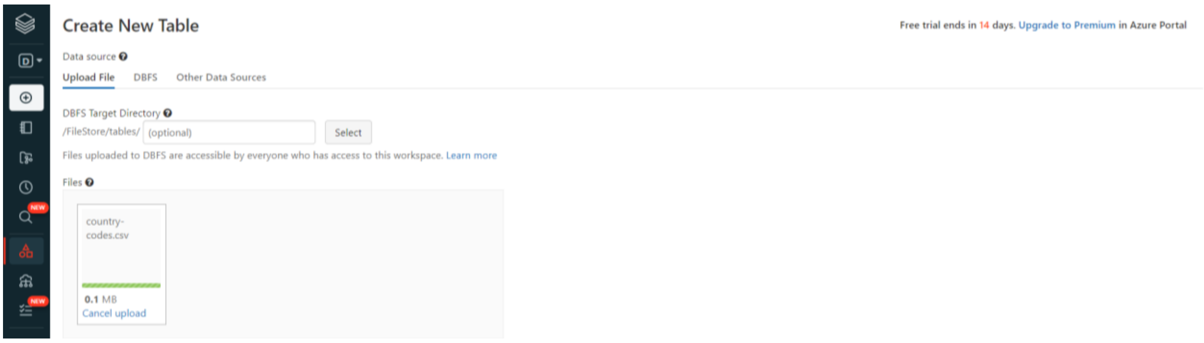
/FileStore/tables/country_codes.csv

Step-5: Import the required fields and features from pyspark.

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

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

SEntFiN-v1_1_schema = StructType(fields=[StructField("S No.",IntegerType(),False),
                                   StructField("Title",StringType(),True),
                                   StructField("Decisions",StringType(),True),
                                   StructField("Words",IntegerType(),True),

])


SEntFiN-v1_1_df = spark.read \
.option("header" , True) \
.schema(SEntFiN-v1_1_schema) \
.csv("/FileStore/tables/ SEntFiN-v1_1.csv")

from pyspark.sql.functions import col,lit

SEntFiN-v1_1_selected_df = SEntFiN-v1_1_df.select(col('S No'),
                                                   col('Title'),col('Words’))

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.

File Being used: cancer_death_rates.csv
Step 1: First, login to your Azure Portal and create a Databricks workspace.
Step 2: Open the Databricks workspace and create cluster for your future use.
Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.
Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

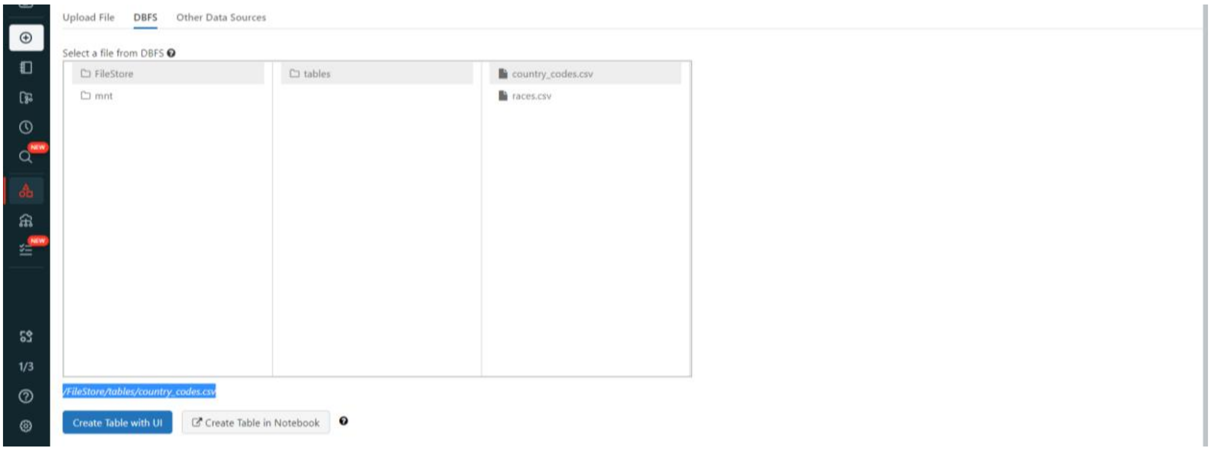
Data source

DBFS Target Directory: /FileStore/tables/ (optional)

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

Files: country-codes.csv 0.1 MB

Later, click on DBFS and select the file that you have dropped. This will give you the file path and copy that.



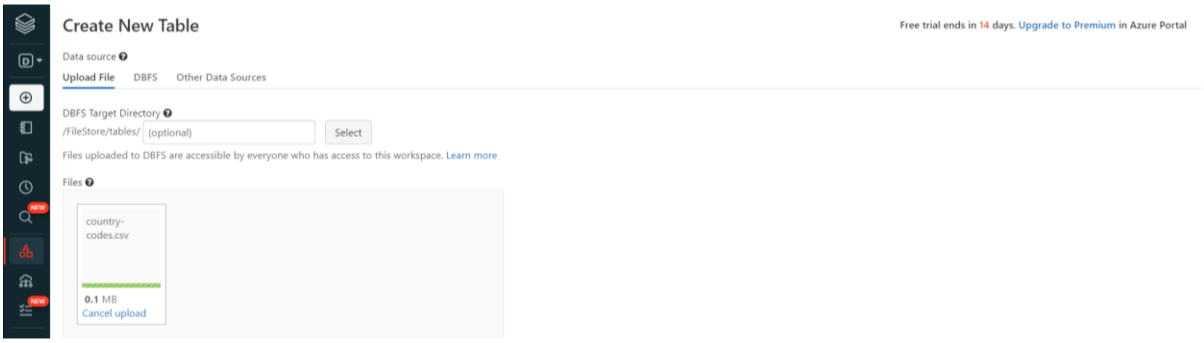
/FileStore/tables/country_codes.csv

Step-5: Import the required fields and features from pyspark.

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

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),

])


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

from pyspark.sql.functions import col,lit

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’))

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.

File Being used: inflation_gdp.csv
Step 1: First, login to your Azure Portal and create a Databricks workspace.
Step 2: Open the Databricks workspace and create cluster for your future use.
Step 3: Now, create a notebook by clicking on the create Notebook option from the side panel.
Step 4: After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the **drag & drop** region.



Create New Table

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

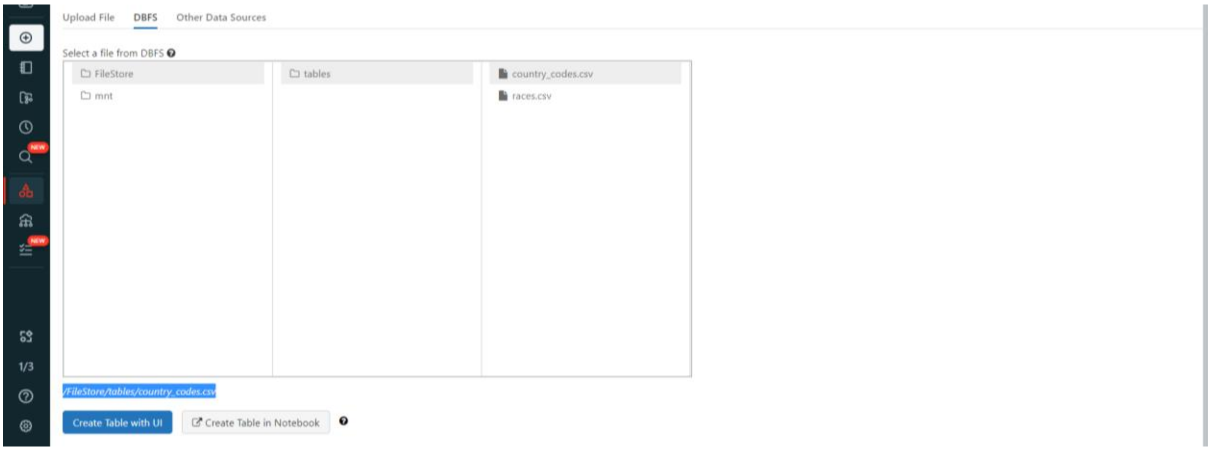
Data source

DBFS Target Directory: /FileStore/tables/ (optional)

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

Files: country-codes.csv, 0.1 MB

Later, click on DBFS and select the file that you have dropped. This will give you the file path and copy that.



Select a file from DBFS

/FileStore/tables/country_codes.csv

Step-5: Import the required fields and features from pyspark.

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

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


])



inflation_gdp _df = spark.read \
.option("header" , True) \
.schema(inflation_gdp _schema) \
.csv("/FileStore/tables/ inflation_gdp.csv")

from pyspark.sql.functions import col,lit

inflation_gdp _selected_df = inflation_gdp _df.select(col('S No'),
                                                      col('Title'),col('Words’))

display(inflation_gdp _df)
```