

1.Implement Processing JSON and CSV data with PySpark

First we have to login into databricks and create a cluster then open new notebook and start the spark session using the following syntax

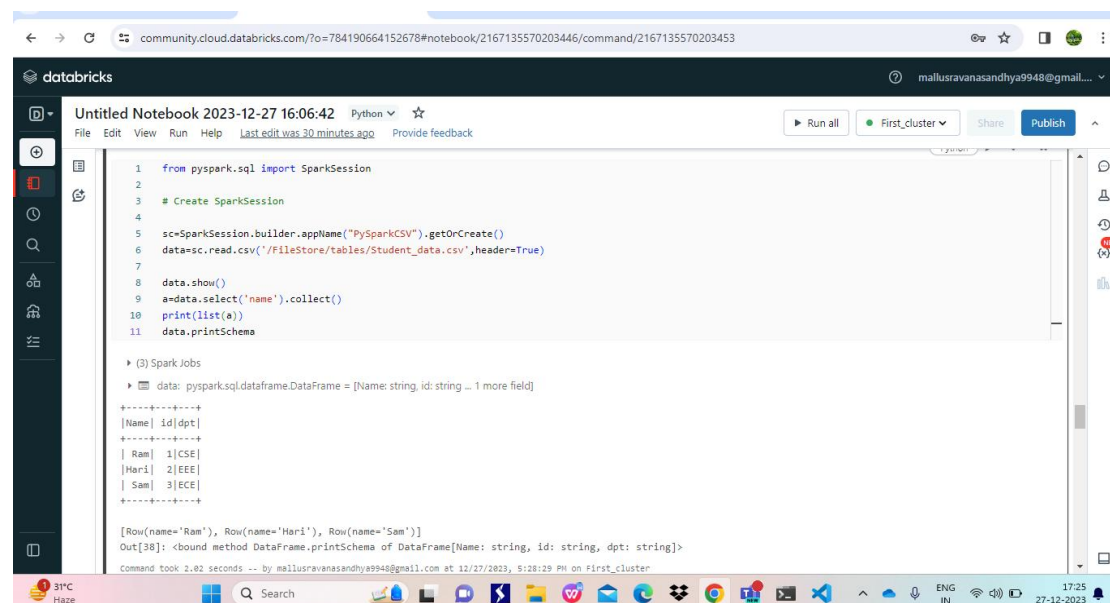
from pyspark.sql import SparkSession

sc=SparkSession.builder.appName("PySparkCSV").getOrCreate()

Then load the csv file from local system to databricks and copy the path and paste in the following command

Sc.read.csv("path of csv file")

Now we can perform any operations on csv data which is loaded from the file



The screenshot shows a Databricks notebook interface. The code in the notebook is as follows:

```
1 from pyspark.sql import SparkSession
2
3 # Create SparkSession
4
5 sc=SparkSession.builder.appName("PySparkCSV").getOrCreate()
6 data=sc.read.csv("/FileStore/tables/Student_data.csv",header=True)
7
8 data.show()
9 a=data.select('name').collect()
10 print(list(a))
11 data.printSchema
```

The output of the code is displayed below the code cell:

```
↳ (3) Spark Jobs
↳ data: pyspark.sql.dataframe.DataFrame = [Name: string, id: string ... 1 more field]

+-----+
|Name| id|dpt|
+-----+
| Ram|  1|CSE|
|Hari|  2|EEE|
| Sam|  3|ECE|
+-----+

[Row(name='Ram'), Row(name='Hari'), Row(name='Sam')]
Out[38]: <bound method DataFrame.printSchema of DataFrame[Name: string, id: string, dpt: string]>
Command took 2.02 seconds -- by mallusravanasandhya9948@gmail.com at 12/27/2023, 5:28:29 PM on First_cluster
```

Similarly we can add json files same as csv files

2. Explain ETL (Extract, Transform, Load) with pyspark

Extract : it extracts the input data from databases or local systems or APIs

Transform: It transforms the data as required by the user for analysis as per the requirement

Load: It saves the data that is transformed inorder to do analysis further

To initialize the spark first we have to import some modules

From pyspark.sql import SparkSession

To perform some actions or transformations we have to import some more functions from `spark.sql.functions`

The command to initialize spark session is

```
Spark=SparkSession.builder.appName("name of session").getOrCreate()
```

To extract data we use below command

```
df = spark.read.csv(source_path, header=True,schema ='col_name1  
datatype,col_name2 datatype,...')
```

Schema is optional

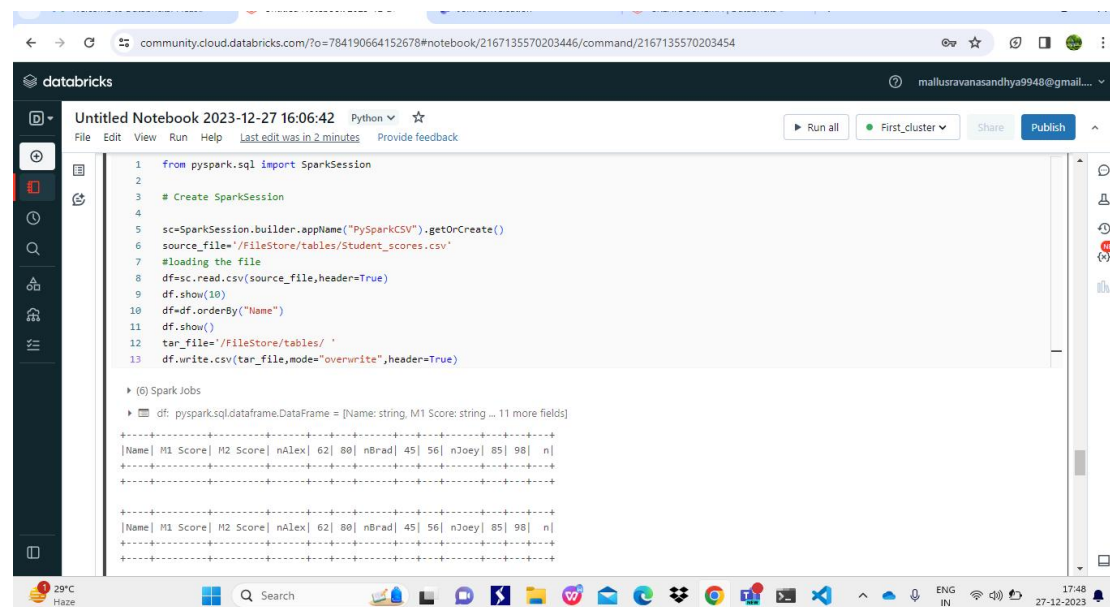
Here `source_path` is the path of the file from where we have to extract the data

Transform:

To transform data we have many functions to transform data some among them are `orderBy` function and got the results

Load :

the results are stored in a file path mentioned in `target_file`



The screenshot shows a Databricks notebook interface. The code in the notebook is as follows:

```
1 from pyspark.sql import SparkSession
2
3 # Create SparkSession
4
5 sc=SparkSession.builder.appName("PySparkCSV").getOrCreate()
6 source_file="/FileStore/tables/Student_scores.csv"
7 #loading the file
8 df=sc.read.csv(source_file,header=True)
9 df.show(10)
10 df=df.orderBy("Name")
11 df.show()
12 tar_file="/FileStore/tables/"
13 df.write.csv(tar_file,mode="overwrite",header=True)
```

The output of the code is displayed below the code cells. It shows the Spark Jobs and the resulting DataFrame. The DataFrame has 11 fields: Name, M1 Score, M2 Score, nAlex, nBrad, nJoey, and n. The output is displayed in a table format with 10 rows of data.

Name	M1 Score	M2 Score	nAlex	nBrad	nJoey	n			
nAlex	62	80	nBrad	45	56	nJoey	85	98	n
nBrad	45	56	nJoey	85	98	n			
nJoey	85	98	n						
n									
nAlex	62	80	nBrad	45	56	nJoey	85	98	n
nBrad	45	56	nJoey	85	98	n			
nJoey	85	98	n						
n									
nAlex	62	80	nBrad	45	56	nJoey	85	98	n
nBrad	45	56	nJoey	85	98	n			
nJoey	85	98	n						
n									

4.Using Spark SQL - Transformations such as Filter, Join, Simple Aggregations, GroupBy

Here we created a table `employee` and converted into dataframes `df` with `name,place,salary,age,bonus` as column

And other dataframe `edf` with `place` and `department` where `place` is common column in both dataframes

First printed both schema and data

Then by using filter command we have filtered employees whose salary is greater than 80000 and shown name and their salary

Then by using aggregate functions we have found the sum of bonus of all employees

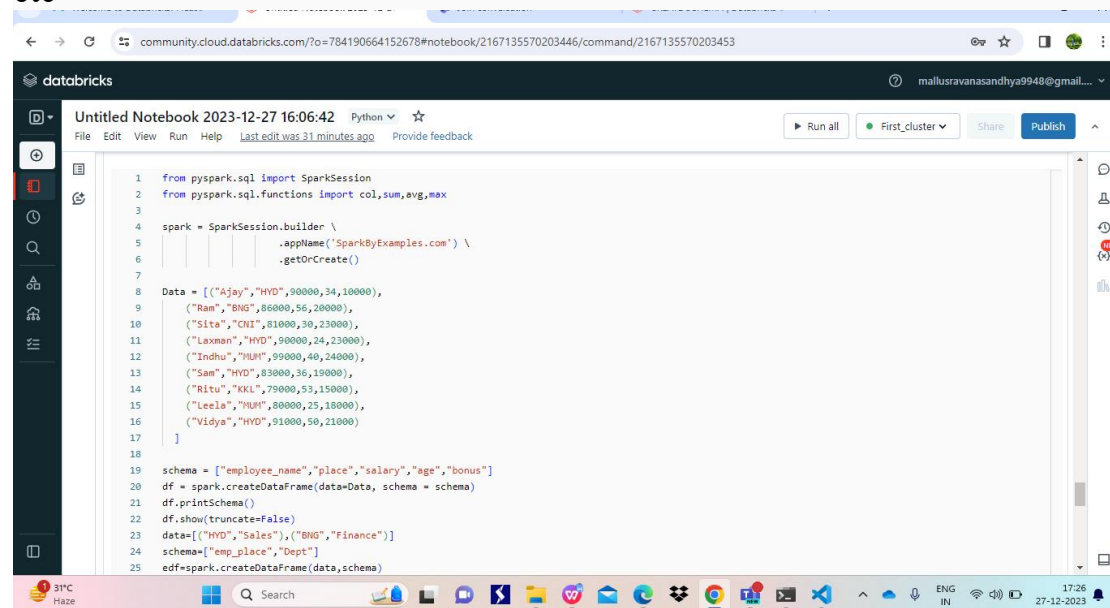
By using groupby, aggregate function we have collected the data of employees grouping by their salary and using aggregate function sum with the help of below statement and sorted by using sort method

```
dfSort=df.sort(df.place,df.salary).groupBy(df.place).agg(sum(df.salary))
```

Then we performed joins on two dataframes whose place is in common by using below command

```
df.join(edf,df.place == edf.emp_place,"inner").show(truncate=False)
```

We can perform different joins by simply replacing inner with other joins like outer,full,fullouter,left,leftouter, etc



The screenshot shows a Databricks notebook interface. The notebook is titled "Untitled Notebook 2023-12-27 16:06:42" and is running Python code. The code defines a SparkSession, creates a DataFrame with employee data, and performs various operations including schema printing, filtering, and joining. The employee data is as follows:

employee_name	place	salary	age	bonus
Ajay	HYD	90000	34	10000
Ram	BNG	86000	56	20000
Sita	CHN	81000	30	23000
Laxman	HYD	90000	24	23000
Indhu	MUM	99000	40	24000
Sam	HYD	83000	36	19000
Ritu	KKL	79000	53	15000
Leela	MUM	80000	25	18000
Vidya	HYD	91000	50	21000

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import col,sum,avg,max
3
4 spark = SparkSession.builder \
5     .appName('SparkByExamples.com') \
6     .getOrCreate()
7
8 Data = [("Ajay","HYD",90000,34,10000),
9         ("Ram","BNG",86000,56,20000),
10        ("Sita","CHN",81000,30,23000),
11        ("Laxman","HYD",90000,24,23000),
12        ("Indhu","MUM",99000,40,24000),
13        ("Sam","HYD",83000,36,19000),
14        ("Ritu","KKL",79000,53,15000),
15        ("Leela","MUM",80000,25,18000),
16        ("Vidya","HYD",91000,50,21000)]
17
18
19 schema = ["employee_name","place","salary","age","bonus"]
20 df = spark.createDataFrame(data=Data, schema = schema)
21 df.printSchema()
22 df.show(truncate=False)
23 data=[("HYD","Sales"),("BNG","Finance")]
24 schema=["emp_place","Dept"]
25 edf=spark.createDataFrame(data,schema)
```

community.cloud.databricks.com/?o=784190664152678#notebook/2167135570203446/command/2167135570203455

databricks mallusravanasandhya9948@gmail...

Untitled Notebook 2023-12-27 16:06:42 Python

```
20 df = spark.createDataFrame(data=Data, schema = schema)
21 df.printSchema()
22 df.show(truncate=False)
23 data=[("HYD","Sales"),("BNG","Finance")]
24 schema=["emp_place","Dept"]
25 edf=spark.createDataFrame(data,schema)
26 edf.printSchema()
27 edf.show()
28 dffilter=df.filter(df.salary>80000)
29 dffilter.show()
30 agg=df.agg(sum(df.bonus))
31 agg.show()
32 dfSort=df.sort(df.place,df.salary).groupBy(df.place).agg(sum(df.salary))
33 dfSort.show()
34 df.join(edf,df.place == edf.emp_place,"inner") \
35     .show(truncate=False)
36
37 df.join(edf,df.place == edf.emp_place,"outer") \
38     .show(truncate=False)
```

(19) Spark Jobs

- df: pyspark.sql.dataframe.DataFrame = [employee_name: string, place: string ... 3 more fields]
- edf: pyspark.sql.dataframe.DataFrame = [emp_place: string, Dept: string]
- dffilter: pyspark.sql.dataframe.DataFrame = [employee_name: string, place: string ... 3 more fields]
- agg: pyspark.sql.dataframe.DataFrame = [sum(bonus): long]

31°C Haze

community.cloud.databricks.com/?o=784190664152678#notebook/2167135570203446/command/2167135570203455

databricks mallusravanasandhya9948@gmail...

Untitled Notebook 2023-12-27 16:06:42 Python

(19) Spark Jobs

- df: pyspark.sql.dataframe.DataFrame = [employee_name: string, place: string ... 3 more fields]
- edf: pyspark.sql.dataframe.DataFrame = [emp_place: string, Dept: string]
- dffilter: pyspark.sql.dataframe.DataFrame = [employee_name: string, place: string ... 3 more fields]
- agg: pyspark.sql.dataframe.DataFrame = [sum(bonus): long]
- dfSort: pyspark.sql.dataframe.DataFrame = [place: string, sum(salary): long]

```
root
|-- employee_name: string (nullable = true)
|-- place: string (nullable = true)
|-- salary: long (nullable = true)
|-- age: long (nullable = true)
|-- bonus: long (nullable = true)
```

employee_name	place	salary	age	bonus
Ajay	HYD	90000	34	10000
Ram	BNG	86000	56	20000
Sita	CNI	81000	30	23000
Laxman	HYD	90000	24	23000
Indhu	MUM	99000	40	24000
Sam	HYD	83000	36	19000
Ritu	KKL	79000	53	15000
Leela	MUM	80000	25	18000

31°C Haze

community.cloud.databricks.com/?o=784190664152678#notebook/2167135570203446/command/2167135570203455

databricks mallusravanasandhya9948@gmail...

Untitled Notebook 2023-12-27 16:06:42 Python

```
root
|-- emp_place: string (nullable = true)
|-- Dept: string (nullable = true)
```

emp_place	Dept
HYD	Sales
BNG	Finance

```
root
|-- employee_name: string (nullable = true)
|-- place: string (nullable = true)
|-- salary: long (nullable = true)
|-- age: long (nullable = true)
|-- bonus: long (nullable = true)
```

employee_name	place	salary	age	bonus
Ajay	HYD	90000	34	10000
Ram	BNG	86000	56	20000
Sita	CNI	81000	30	23000
Laxman	HYD	90000	24	23000
Indhu	MUM	99000	40	24000
Sam	HYD	83000	36	19000

Command took 6.07 seconds -- by mallusravanasandhya9948@gmail.com at 12/27/2023, 5:12:44 PM on First_cluster

Shift+Enter to run
Shift+Ctrl+Enter to run selected text

31°C Haze

community.cloud.databricks.com/?o=784190664152678#notebook/2167135570203446/command/2167135570203455

databricks mallusravanasandhya9948@gmail...

Untitled Notebook 2023-12-27 16:06:42 Python Last edit was 32 minutes ago Provide feedback Run all First_cluster Share Publish

```
| Vidya| HYD| 91000| 50| 21000|
+-----+
sum(bonus)
+-----+
| 173000|
+-----+

+-----+
place|sum(salary)|
+-----+
| HYD| 354000|
| BNG| 86000|
| CNI| 81000|
| MUM| 179000|
| KKL| 79000|
+-----+

Command took 6.07 seconds -- by mallusravanasandhya9948@gmail.com at 12/27/2023, 5:22:44 PM on First_cluster
```

Shift+Enter to run
Shift+ctrl+Enter to run selected text

31°C Haze Search 17:27 27-12-2023

community.cloud.databricks.com/?o=784190664152678#notebook/2167135570203446/command/2167135570203455

databricks mallusravanasandhya9948@gmail...

Untitled Notebook 2023-12-27 16:06:42 Python Last edit was 33 minutes ago Provide feedback Run all First_cluster Share Publish

```
|employee_name|place|salary|age|bonus|emp_place|Dept|
+-----+
|Ram| BNG| 86000| 56| 20000| BNG| Finance|
|Ajay| HYD| 90000| 34| 10000| HYD| Sales|
|Laxman| HYD| 90000| 24| 23000| HYD| Sales|
|Sam| HYD| 83000| 36| 19000| HYD| Sales|
|Vidya| HYD| 91000| 50| 21000| HYD| Sales|
+-----+

+-----+
employee_name|place|salary|age|bonus|emp_place|Dept|
+-----+
|Ram| BNG| 86000| 56| 20000| BNG| Finance|
|Sita| CNI| 81000| 30| 23000| null| null|
|Ajay| HYD| 90000| 34| 10000| HYD| Sales|
|Laxman| HYD| 90000| 24| 23000| HYD| Sales|
|Sam| HYD| 83000| 36| 19000| HYD| Sales|
|Vidya| HYD| 91000| 50| 21000| HYD| Sales|
|Ritu| KKL| 79000| 53| 15000| null| null|
+-----+

Command took 6.07 seconds -- by mallusravanasandhya9948@gmail.com at 12/27/2023, 5:22:44 PM on First_cluster
```

Shift+Enter to run
Shift+ctrl+Enter to run selected text

31°C Haze Search 17:28 27-12-2023