1. GroupBy and Aggregate Functions:

- **Definition**: GroupBy is a method in PySpark used to group rows together based on one or more columns and perform aggregate functions on the grouped data.

- **Aggregate Functions:**

- **sum():** Computes the sum of values in a group.
- **min():** Finds the minimum value in a group.
- **max():** Finds the maximum value in a group.
- **avg():** Computes the average value in a group.
- **mean():** Computes the mean value in a group.
- **count():** Counts the number of rows in each group.

2. Handling Missing Values:

- **Dropping Rows with Null Values:**

- **na.drop():** Drops rows containing null values.
- **Parameters:**
- **how:** Specifies whether to drop rows if any or all values are null.
- **thresh:** Specifies the minimum number of non-null values required for a row to be kept.
- **subset:** Specifies the columns to consider for null value checks.
- Filling Missing Values:

- **na.fill():** Replaces null values with specified values.
- **Single Value:** Replace nulls with a single specified value.

- **Mean/Median/Mode:** Impute nulls with mean, median, or mode of the column using Imputer.

3. Sorting and Ordering:

- **Sorting:**
- **sort():** Sorts DataFrame based on specified columns.
- **orderBy():** Sorts DataFrame in ascending order based on specified columns.

4. Joins:

- **Definition:** Joins in PySpark are used to combine two DataFrames based on a common column(s).

- **Types of Joins:**

- **Inner Join:** Returns rows with matching keys in both DataFrames.
- **Outer Join:** Returns all rows when there is a match in either DataFrame.
- **Left Join:** Returns all rows from the left DataFrame and matching rows from the right DataFrame.
- **Right Join:** Returns all rows from the right DataFrame and matching rows from the left DataFrame.
- **Left Semi Join:** Returns rows from the left DataFrame for which a match exists in the right DataFrame.
- **Left Anti Join:** Returns rows from the left DataFrame for which no match exists in the right DataFrame.

5. Union:

Definition: Union is used to combine two DataFrames with the same schema.

Distinct Union: Use distinct() to merge without duplicates.

6. User Defined Functions (UDF):

- **Definition:** UDFs extend the functionality of PySpark by allowing users to define custom functions for DataFrame operations.
- **Creating UDF:**
- Define a Python function.
- Register the function as a UDF.
- **Applying UDF:**
- Use the UDF with withColumn() to apply it to DataFrame columns.
- Each topic provides essential functionalities and operations in PySpark for data manipulation, analysis, and transformation. Understanding and mastering these concepts are crucial for effective data processing in PySpark.

GroupBy and Aggregate function

In [1]:
```
from pyspark.sql import SparkSession
spark =SparkSession.builder.appName("Practice").getOrCreate()
df_pyspark= spark.read.csv("file1.csv",header=True,inferSchema=True)
df_pyspark.show()
df_pyspark.groupBy("Departments").sum("salary").show()
```

```
+---------+------------+------+
|     Name| Departments|salary|
+---------+------------+------+
|    Krish|Data Science| 10000|
|    Krish|         IOT|  5000|
|   Mahesh|    Big Data|  4000|
|    Krish|    Big Data|  4000|
|   Mahesh|Data Science|  3000|
|Sudhanshu|Data Science| 20000|
|Sudhanshu|         IOT| 10000|
|Sudhanshu|    Big Data|  5000|
|    Sunny|Data Science| 10000|
|    Sunny|    Big Data|  2000|
+---------+------------+------+

+------------+-----------+
| Departments|sum(salary)|
+------------+-----------+
|         IOT|      15000|
|    Big Data|      15000|
|Data Science|      43000|
+------------+-----------+
```

In [2]:
```
df_pyspark.groupBy("Departments").sum("salary").show()
```

```
+------------+-----------+
| Departments|sum(salary)|
+------------+-----------+
|         IOT|      15000|
|    Big Data|      15000|
|Data Science|      43000|
+------------+-----------+
```

In [3]:
```
df_pyspark.groupBy("Departments").min("salary").show()
```

```
+------------+-----------+
| Departments|min(salary)|
+------------+-----------+
|         IOT|       5000|
|    Big Data|       2000|
|Data Science|       3000|
+------------+-----------+
```

In [4]:
```
df_pyspark.groupBy("Departments").max("salary").show()
```

```
+------------+-----------+
| Departments|max(salary)|
+------------+-----------+
|         IOT|      10000|
|    Big Data|       5000|
|Data Science|      20000|
+------------+-----------+
```

In [5]:
```
df_pyspark.groupBy("Departments").avg("salary").show()
```

```
+------------+-----------+
| Departments|avg(salary)|
+------------+-----------+
|         IOT|     7500.0|
|    Big Data|     3750.0|
|Data Science|    10750.0|
+------------+-----------+
```

In [6]:
```
df_pyspark.groupBy("Departments").mean("salary").show()
```

```
+------------+-----------+
| Departments|avg(salary)|
+------------+-----------+
|         IOT|     7500.0|
|    Big Data|     3750.0|
|Data Science|    10750.0|
+------------+-----------+
```

In [7]:
```
df_pyspark.groupBy("Departments").count().show()
```

```
+------------+-----+
| Departments|count|
+------------+-----+
|         IOT|    2|
|    Big Data|    4|
|Data Science|    4|
+------------+-----+
```

In [8]:
```
df_pyspark.groupBy("Name","Departments").sum("salary").show()
```

```
+---------+-----------+-----------+
|     Name| Departments|sum(salary)|
+---------+-----------+-----------+
|    Sunny|    Big Data|       2000|
|   Mahesh|Data Science|       3000|
|Sudhanshu|Data Science|      20000|
|    Krish|Data Science|      10000|
|   Mahesh|    Big Data|       4000|
|    Sunny|Data Science|      10000|
|Sudhanshu|         IOT|      10000|
|    Krish|         IOT|       5000|
|Sudhanshu|    Big Data|       5000|
|    Krish|    Big Data|       4000|
+---------+-----------+-----------+
```

Agregate function using group by

In [9]:
```
df_pyspark.groupBy("Departments").agg(({"salary":"sum"})).show()
```

```
+------------+-----------+
| Departments|sum(salary)|
+------------+-----------+
|         IOT|      15000|
|    Big Data|      15000|
|Data Science|      43000|
+------------+-----------+
```

In [10]:
```
df_pyspark.agg(({"salary":"sum"})).show()
```

```
+-----------+
|sum(salary)|
+-----------+
|      73000|
+-----------+
```

In [11]:
```
df_pyspark.groupBy("Departments").pivot("Name").sum("salary").show()
```

```
+------------+-----+------+---------+-----+
| Departments|Krish|Mahesh|Sudhanshu|Sunny|
+------------+-----+------+---------+-----+
|         IOT| 5000|  NULL|    10000| NULL|
|    Big Data| 4000|  4000|     5000| 2000|
|Data Science|10000|  3000|    20000|10000|
+------------+-----+------+---------+-----+
```

Handling Missing Values

In [13]:
```
df_pyspark1=spark.read.csv("file 2.csv",header=True,inferSchema=True)
df_pyspark1.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

In [14]:
```
df_pyspark1.na.drop().show()
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
|    Sunny| 29|         4| 20000|
|     Paul| 24|         3| 20000|
|   Harsha| 21|         1| 15000|
|  Shubham| 23|         2| 18000|
+---------+---+----------+------+
```

how, thresh and subset

In [16]:
```
df_pyspark1.na.drop(how="all").show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

In [17]:
```
df_pyspark1.na.drop(how="any",thresh=2).show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
```

```
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
+---------+----+----------+------+
```

In [18]:
```
df_pyspark1.na.drop(how="any",subset=["salary"]).show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
+---------+----+----------+------+
```

sorting

In [19]:
```
df_pyspark.sort("salary").show()
```

```
+---------+-----------+------+
|     Name| Departments|salary|
+---------+-----------+------+
|    Sunny|    Big Data|  2000|
|   Mahesh|Data Science|  3000|
|   Mahesh|    Big Data|  4000|
|    Krish|    Big Data|  4000|
|    Krish|         IOT|  5000|
|Sudhanshu|    Big Data|  5000|
|    Krish|Data Science| 10000|
|Sudhanshu|         IOT| 10000|
|    Sunny|Data Science| 10000|
|Sudhanshu|Data Science| 20000|
+---------+-----------+------+
```

In [20]:
```
df_pyspark.sort(df_pyspark["salary"].desc()).show()
```

```
+---------+-----------+------+
|     Name| Departments|salary|
+---------+-----------+------+
|Sudhanshu|Data Science| 20000|
|    Krish|Data Science| 10000|
|Sudhanshu|         IOT| 10000|
|    Sunny|Data Science| 10000|
|    Krish|         IOT|  5000|
|Sudhanshu|    Big Data|  5000|
|   Mahesh|    Big Data|  4000|
|    Krish|    Big Data|  4000|
|   Mahesh|Data Science|  3000|
|    Sunny|    Big Data|  2000|
+---------+-----------+------+
```

In [21]:
```
df_pyspark.sort("salary","Name").show()
```

```
+---------+-----------+------+
|     Name| Departments|salary|
+---------+-----------+------+
|    Sunny|    Big Data|  2000|
|   Mahesh|Data Science|  3000|
|    Krish|    Big Data|  4000|
|   Mahesh|    Big Data|  4000|
|    Krish|         IOT|  5000|
|Sudhanshu|    Big Data|  5000|
|    Krish|Data Science| 10000|
|Sudhanshu|         IOT| 10000|
|    Sunny|Data Science| 10000|
|Sudhanshu|Data Science| 20000|
+---------+-----------+------+
```

join() Pyspark

In [22]:
```

```

```
  Cell In[22], line 1
    emp = [(1,"Smith",-1,"2018","10","M",3000),(2, "Rose",1 , "2010", "20","M", 4000),(3,"Williams",1,"2010","10","M",1000),(4,
"Jones",2 ,"2005","10","F",2000),(5,"Brown",2,"2010","40","",-1),(6, "Brown", 2, "2010","50","",-1)]empColumns = ["emp_id","nam
e","superior_emp_id","year_joined", "emp_dept_id","gender","salary"]

    ^
SyntaxError: invalid syntax
```

In [ ]:
```

```

databricks

abhishek7621cse@gmail.com

Pyspark3 Python

File Edit View Run Help Last edit was 5 minutes ago New cell UI: OFF

Run all My Cluster Share Publish

Cmd 1

```
1
```

Cmd 2

Python

```
1
```

Cmd 3

Python

```
1  # Joins
2
```

Command took 1.47 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 4

```
1  from pyspark.sql import SparkSession
2  spark =SparkSession.builder.appName('pyspark - example join').getOrCreate()
3  emp = [(1,"Smith",-1,"2018","10","M",3000),(2, "Rose",1 , "2010", "20","M", 4000),(3,"Williams",1,"2010","10","M",1000),(4, "Jones",2 ,"2005","10","F",2000),(5,"Brown",2,"2010",
   "40","",-1),(6, "Brown", 2, "2010","50","",-1)]
4  empColumns = ["emp_id","name","superior_emp_id","year_joined", "emp_dept_id","gender","salary"]
5
6  empDF = spark.createDataFrame(data=emp, schema = empColumns)
7  empDF.printSchema()
8  empDF.show()
9
10 dept = [("Finance",10),("Marketing",20),("Sales",30),("IT",40)]
11 deptColumns = ["dept_name","dept_id"]
12 deptDF = spark.createDataFrame(data=dept, schema = deptColumns)
13 deptDF.printSchema()
14 deptDF.show()
```

(6) Spark Jobs

empDF: pyspark.sql.dataframe.DataFrame = [emp_id: long, name: string ... 5 more fields]

deptDF: pyspark.sql.dataframe.DataFrame = [dept_name: string, dept_id: long]

```
|     1|   Smith|            -1|       2018|         10|     M|  3000|
|     2|    Rose|             1|       2010|         20|     M|  4000|
|     3|Williams|             1|       2010|         10|     M|  1000|
|     4|   Jones|             2|       2005|         10|     F|  2000|
|     5|   Brown|             2|       2010|         40|      |    -1|
|     6|   Brown|             2|       2010|         50|      |    -1|
+------+--------+--------------+-----------+-----------+------+------+

root
 |-- dept_name: string (nullable = true)
 |-- dept_id: long (nullable = true)

+---------+-------+
|dept_name|dept_id|
+---------+-------+
|  Finance|     10|
|Marketing|     20|
|    Sales|     30|
|       IT|     40|
+---------+-------+
```

Command took 15.36 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 5

```
1  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"inner") .show()
```

(3) Spark Jobs

```
+------+--------+--------------+-----------+-----------+------+------+---------+-------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+--------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|            -1|       2018|         10|     M|  3000|  Finance|     10|
|     3|Williams|             1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|             2|       2005|         10|     F|  2000|  Finance|     10|
|     2|    Rose|             1|       2010|         20|     M|  4000|Marketing|     20|
|     5|   Brown|             2|       2010|         40|      |    -1|       IT|     40|
+------+--------+--------------+-----------+-----------+------+------+---------+-------+
```

Command took 6.38 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 6

```
1  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"outer").show()
2  #Or
3  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"full").show()
4  #Or
5  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"fullouter").show()
```

(9) Spark Jobs

```
|     1|   Smith|            -1|       2018|         10|     M|  3000|  Finance|     10|
|     3|Williams|             1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|             2|       2005|         10|     F|  2000|  Finance|     10|
|     2|    Rose|             1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|          null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|             2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|             2|       2010|         50|      |    -1|     null|   null|
+------+--------+--------------+-----------+-----------+------+------+---------+-------+

+------+--------+--------------+-----------+-----------+------+------+---------+-------+
```

```
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|           null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|              2|       2010|         50|      |    -1|     null|   null|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Command took 5.58 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 7

```
1  # Left Join
2  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"left").show()
3  #Or
4  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"leftouter").show()
```

▸ (12) Spark Jobs

```
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|              2|       2010|         50|      |    -1|     null|   null|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+


|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|              2|       2010|         50|      |    -1|     null|   null|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Command took 5.46 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 8

```
1  # Right Join
2  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"right").show()
3  #Or
4  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"rightouter").show()
5
```

▸ (12) Spark Jobs

```
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|           null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+


|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|           null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Command took 4.91 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 9

```
1  # Left semi Join
2  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"leftsemi").show()
3
```

▸ (3) Spark Jobs

```
+------+--------+---------------+-----------+-----------+------+------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|
+------+--------+---------------+-----------+-----------+------+------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|
|     3|Williams|              1|       2010|         10|     M|  1000|
|     4|   Jones|              2|       2005|         10|     F|  2000|
|     2|    Rose|              1|       2010|         20|     M|  4000|
|     5|   Brown|              2|       2010|         40|      |    -1|
+------+--------+---------------+-----------+-----------+------+------+
```

Command took 1.90 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 10

```
1  # Leftanti Join
2  empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"leftanti").show()
3
```

▸ (6) Spark Jobs

```
+------+-----+---------------+-----------+-----------+------+------+
|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary|
+------+-----+---------------+-----------+-----------+------+------+
|     6|Brown|              2|       2010|         50|      |    -1|
+------+-----+---------------+-----------+-----------+------+------+
```

Command took 1.70 seconds -- by abhishek7621cse@gmail.com at 2/12/2024, 11:38:21 AM on My Cluster

Cmd 11

```
1
```

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