GroupBy with Aggregate Functions

Single Column:

Use groupBy("column name") and apply an aggregate function (e.g., sum(), avg(), count()).

```
df.groupBy("category").agg({"sales": "sum"}).show()
```

Multiple Columns:

Group by multiple columns by passing a list.

```
df.groupBy("category", "region").agg({"sales": "avg"}).show()
```

Pivot

Reshape data, turning unique values of a column into new column headers.

Used with aggregation.

```
df.groupBy("category").pivot("year").sum("sales").show()
```

Dropping Null Values

Use dropna() to remove rows containing null values.

Drop rows with nulls in any column:

```
df.dropna().show()
```

Drop rows with nulls in **specific columns**:

```
df.dropna(subset=["column1", "column2"]).show()
```

Drop rows based on a **threshold** of non-null values:

```
df.dropna(thresh=2).show() # At least 2 non-null values
```

Sorting in PySpark: sortBy and orderBy

sort() Sorts a DataFrame based on one or more columns.

Equivalent to orderBy in PySpark; they perform the same function.

```
df.sort("column1", "column2").show() # Default is ascending
df.sort(df["column1"].desc()).show() # Specify descending
```

orderBy() Another way to sort a DataFrame. Allows sorting by multiple columns.

Offers more flexibility by supporting column expressions for sorting.

```
df.orderBy("column1", "column2").show() # Default ascending
df.orderBy(df["column1"].desc(), df["column2"].asc()).show() # Mixed order
```

Types of Sorting

Single Column Sorting:

DATA ENGINEERING PYSPARK

df.orderBy("column1").show()

df.orderBy(df["column1"].desc()).show()

Multiple Column Sorting:

Ascending for all: df.orderBy("column1", "column2").show()

Mixed order: df.orderBy(df["column1"].asc(), df["column2"].desc()).show()

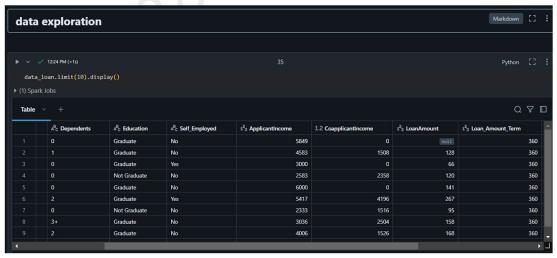
Using Expressions: Sorting can involve transformations:

from pyspark.sql.functions import col

df.orderBy(col("column1") + 1).show() # Sort by column1 incremented by 1

```
create data
  11:06 AM (1s)
  from pyspark.sql import SparkSession
  spark = SparkSession.builder \
  .appName("example") \
  .getOrCreate()
  simpleData = [("James", "Sales", "NY", 90000, 34, 10000),
  ("Michael", "Sales", "NY", 86000, 56, 20000),
  ("Robert", "Sales", "CA", 81000, 30, 23000),
  ("Maria", "Finance", "CA", 90000, 24, 23000), ("Raman", "Finance", "CA", 99000, 40, 24000),
  ("Scott", "Finance", "NY", 83000, 36, 19000),
  ("Jen", "Finance", "NY", 79000, 53, 15000),
  ("Jeff", "Marketing", "CA", 80000, 25, 18000),
  ("Kumar", "Marketing", "NY", 91000, 50, 21000)
  schema = ["employee_name","department","state","salary","age","bonus"]
  df = spark.createDataFrame(data=simpleData, schema = schema)
```



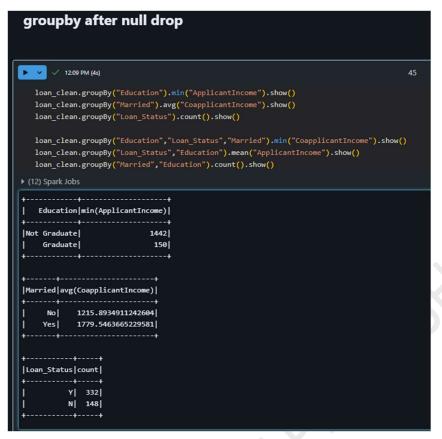


DATA ENGINEERING PYSPARK

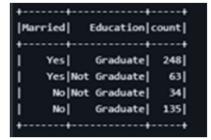
```
V 11:59 AM (1s)
   data_loan.printSchema()
   print((data_loan.count(), len(data_loan.columns)))
▶ (2) Spark Jobs
root
|-- Loan_ID: string (nullable = true)
|-- Gender: string (nullable = true)
|-- Married: string (nullable = true)
|-- Dependents: string (nullable = true)
|-- Education: string (nullable = true)
|-- Self_Employed: string (nullable = true)
|-- ApplicantIncome: integer (nullable = true)
|-- CoapplicantIncome: double (nullable = true)
|-- LoanAmount: integer (nullable = true)
|-- Loan_Amount_Term: integer (nullable = true)
|-- Credit_History: integer (nullable = true)
 |-- Property_Area: string (nullable = true)
 |-- Loan_Status: string (nullable = true)
(614, 13)
```

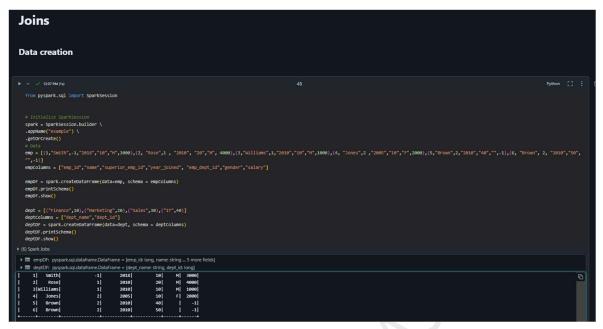
groupby ▶ ∨ ✓ 12:05 PM (2s) data_loan.groupBy("Education").min("ApplicantIncome").show() data_loan.groupBy("Married").avg("CoapplicantIncome").show() data_loan.groupBy("Loan_Status").count().show() ▶ (6) Spark Jobs Education|min(ApplicantIncome)| 210| 150| |Not Graduate Graduate +-----+ |Married|avg(CoapplicantIncome)| +-----+ null| 251.33333333333334| No| 1316.5586854460093| Yes | 1794.632964795578| |Loan_Status|count| Y| 422| N| 192|

```
drop null()
       12:07 PM (1s)
   loan_copy = data_loan
   loan_clean = loan_copy.na.drop()
▶ (4) Spark Jobs
 ▶ ■ loan_copy: pyspark.sql.dataframe.DataFrame = [Loan_ID: string, Gender: string ... 11 more fields]
 ▶ ■ loan_clean: pyspark.sql.dataframe.DataFrame = [Loan_ID: string, Gender: string ... 11 more fields]
(614, 13)
(480, 13)
data after dropping null
                                                                             + Code
                                                                                      + Text
       12:08 PM (1s)
   print((data_loan.count(), len(data_loan.columns)))
   print((loan_clean.count(), len(loan_clean.columns)))
▶ (4) Spark Jobs
(614, 13)
(480, 13)
```



```
| Education|Loan_Status|Married|min(CoapplicantIncome)|
0.0
                                      0.0
                                       0.0
                                       0.0
                                       0.0
                                       0.0
 Graduate|
                Y No
                                       0.0
|Loan_Status| Education|avg(ApplicantIncome)|
      Y| Graduate| 5465.49446494465|
      N|Not Graduate| 3654.3333333333335|
       N| Graduate| 6397.428571428572|
Y|Not Graduate| 4026.4590163934427|
    -----+-------
```





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

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

```
Inner, Outer, Full
    12:39 PM (1s)
  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 2018 10 M 3000 Finance 1 2010 10 M 1000 Finance 2 2005 10 F 2000 Finance 1 2010 20 M 4000 Marketing 2 2010 40 -1 IT
    1 Smith
                                                                            10
    3|Williams|
                                                                            10
    4 Jones
                                                                            10
    2
         Rose
                                                                            20
     5 Brown
                                                                           40
```

DATA ENGINEERING PYSPARK

```
empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "full").show()
|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
                           -1 2018 10 M 3000 Finance 1 2010 10 M 1000 Finance 2 2005 10 F 2000 Finance 1 2010 20 M 4000 Marketing 1 null null null null Sales 2 2010 40 -1 IT 2 2010 50 -1 null
      1 Smith
      3|Williams|
                                                                                                         10
      4 Jones
                                                                                                         10
                                                                                                         20
             Rose
   null|
                                                                                                         30
      5
                                                                                                         40
      6
                                                                                           null null
```

```
left, left outer
      12:40 PM (1s)
   empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "left").show()
▶ (6) Spark Jobs
|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
                                                           10| M| 3000| Finance|
20| M| 4000|Marketing|
10| M| 1000| Finance|
10| F| 2000| Finance|
                             -1 2018 | 1 2010 | 1 2010 | 2 2010 | 2 2010 | 2 2010 |
      1 Smith
                                                                   20| M| 4000|Marketing|
10| M| 1000| Finance|
10| F| 2000| Finance|
40| | -1| IT|
50| | -1| null|
      2
              Rose
                                                                                                            20
      3|Williams|
                                                                                                             10
       4 Jones
                                                                                                            10
       5 Brown
                                                                                                            40
                                                                                   -1| null| null|
       6| Brown|
```

```
empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "leftouter").show()
(6) Spark Jobs
        name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
emp id
                              2018
                                                  M 3000 Finance
   1 Smith
                                          10
                                                                      10
                              2010
                                          20
                                                 M| 4000|Marketing|
   2
        Rose
                                                                      20
   3|Williams|
                      1
                              2010
                                          10
                                                  M 1000 Finance
                                                                      10
                       2
                              2005
                                          10
                                                  F 2000 Finance
                                                                      10
   4 Jones
                        2
   5
       Brown
                                2010
                                           40
                                                       -1
                                                               IT|
                                                                      40
                        2
                                                   | -1|
                                2010
                                           50
                                                              null|
                                                                    null|
       Brown
```

```
Right, Right Outer
    12:41 PM (1s)
  empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "right").show()
▶ (6) Spark Jobs
|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
                -----
                   2| 2005| 10| F| 2000| Finance|
1| 2010| 10| M| 1000| Finance|
    4 Jones
    3|Williams|
                                        10| M| 1000| Finance|
                     -1 2018
1 2010
null null
2 2010
                                        10 | M| 3000 | Finance
    1 Smith
                                                                   10|
                                        20
                                              M| 4000|Marketing|
    2
         Rose
                                                                   20
                                        null null null Sales
  null
                                                                   30
        null
    5 Brown
                    2
                                       40
                                              | -1|
                                                           IT|
                                                                   40
```

```
#right outer join
  empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "rightouter").show()
         name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
emp id
                                2005
                                            10|
                                               F| 2000| Finance|
                                                                       10|
    4
        Jones
                      1|
    3|Williams|
                                2010
                                           10|
                                                  M| 1000| Finance|
                       -1|
                                           10
                                                  M 3000 Finance
    1 Smith
                                2018
                                                                       10
                       1
                                          20
    2
                                2010
                                                  M 4000 Marketing
                                                                       20
         Rose
                    null|
  null|
        nul1
                                          null null null Sales
                                                                       30|
                                null
    5 Brown
                      2|
                                2010
                                          40
                                                | -1|
                                                                IT|
                                                                       40
```

LeftSemi LeftAnti V 12:42 PM (1s) #leftsemijoin empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "leftsemi").show() ▶ (3) Spark Jobs **4-----**|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary| **+-----**10| M| 3000| 10| M| 1000| 10| F| 2000| 20| M| 4000| 1 Smith -1| 2018 1 2010 | 2010 | 2 | 2005 | 1 | 2010 | 3|Williams| 4 Jones 2 Rose 2 40 | -1| 5| Brown| 2010 ▶ ✓ 12:42 PM (2s) empDF.join(deptDF,empDF.emp_dept_id == deptDF.dept_id, "leftanti").show() ▶ (6) Spark Jobs +-----+ |emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary| 50 l | -1| 2 2010

Spark SQL using Loan Data (Clean) Create View ▶ ∨ ✓ 06:45 PM (<1s) loan_clean.createOrReplaceTempView("loan") loan_clean.printSchema() root |-- Loan_ID: string (nullable = true) |-- Gender: string (nullable = true) |-- Married: string (nullable = true) |-- Dependents: string (nullable = true) |-- Education: string (nullable = true) |-- Self_Employed: string (nullable = true) |-- ApplicantIncome: integer (nullable = true) |-- CoapplicantIncome: double (nullable = true) |-- LoanAmount: integer (nullable = true) |-- Loan_Amount_Term: integer (nullable = true) |-- Credit_History: integer (nullable = true) |-- Property_Area: string (nullable = true) |-- Loan_Status: string (nullable = true)

