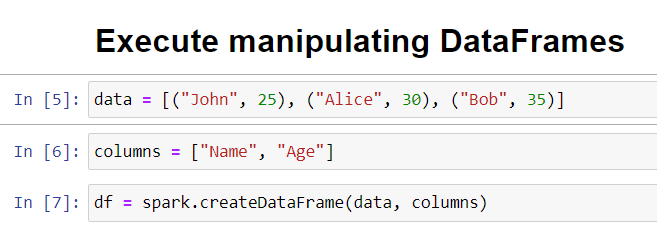
NAME - APARNA BBHARTI

BATCH - DATA ENGINEERING BATCH - 1

**PYSPARK ASSESSMENT**

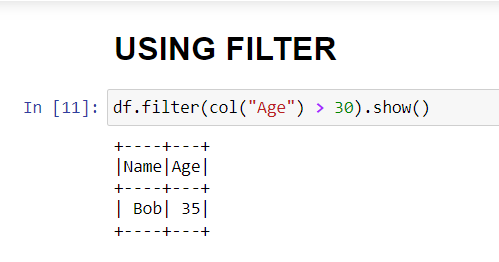
**Que 1) Execute**

1. Manipulating DataFrames



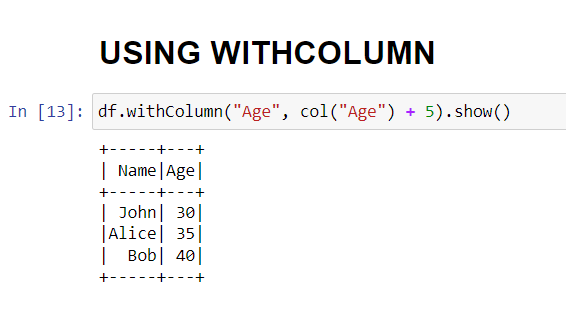
**FILTER**

PySpark filter() function is used to filter the rows from RDD/DataFrame based on the given condition or SQL expression, you can also use where() clause instead of the filter() if you are coming from an SQL background, both these functions operate exactly the same. filter() function returns a new DataFrame or RDD with only the rows that meet the condition specified.



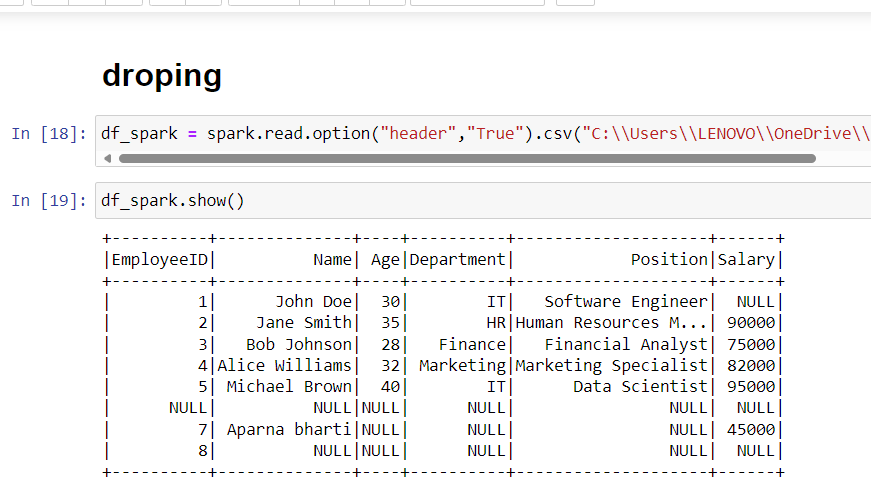
**WITHCOLUMN**

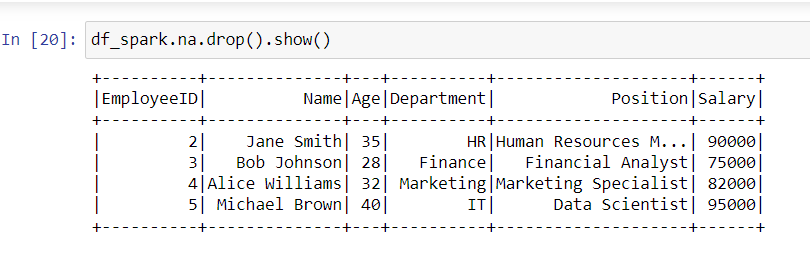
Spark withColumn() is a DataFrame function that is used to add a new column to DataFrame, change the value of an existing column, convert the datatype of a column, derive a new column from an existing column

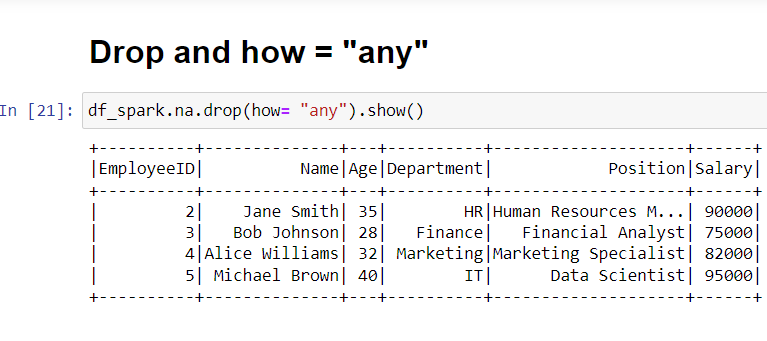


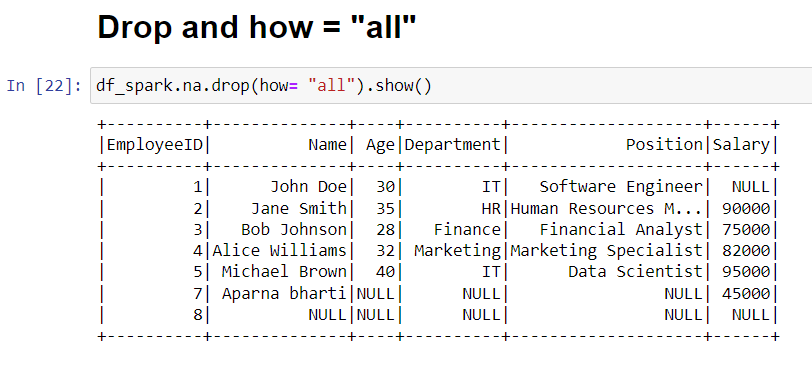
1. Dropping DataFrames

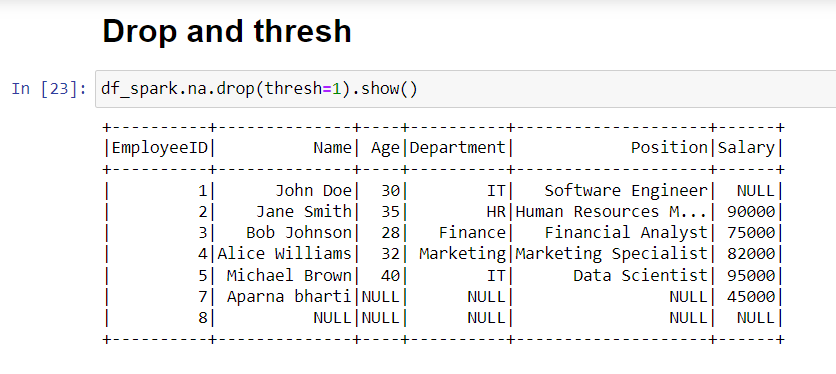
PySpark DataFrame provides a drop() method to drop a single column/field or multiple columns from a DataFrame/Dataset

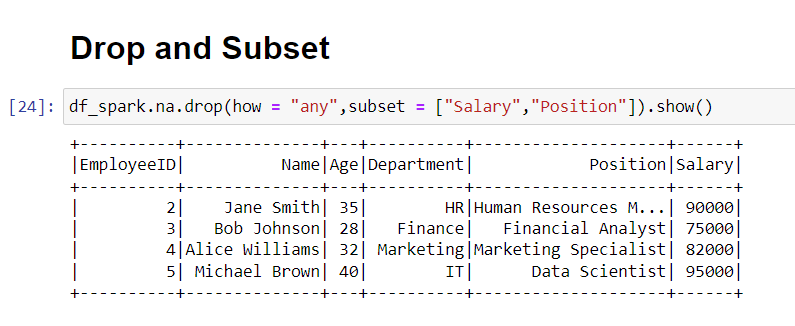






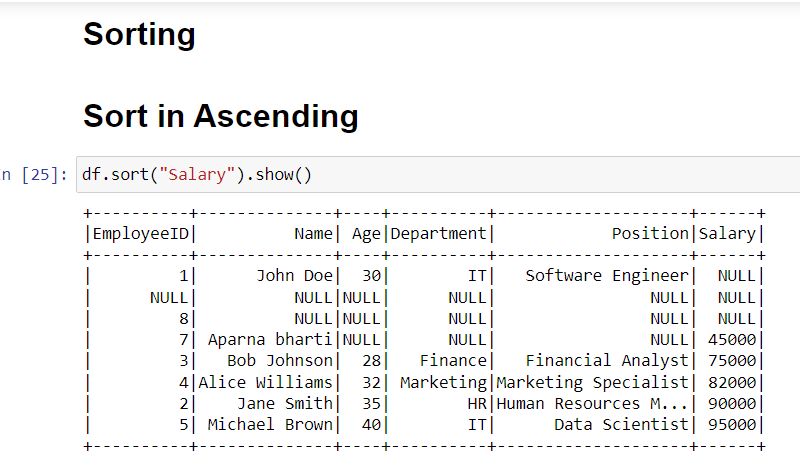


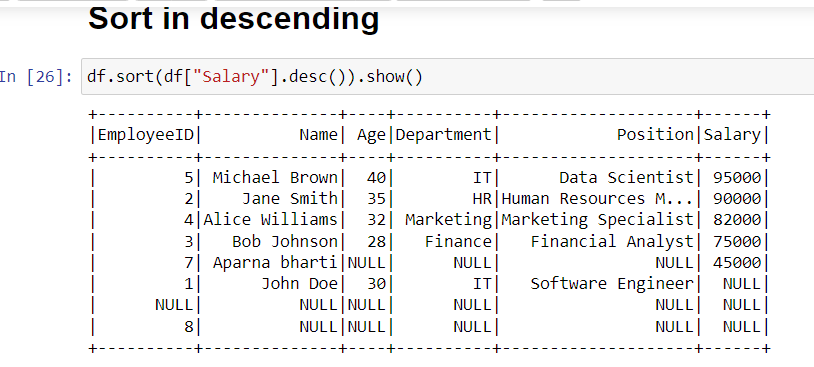




c) Sorting DataFrames

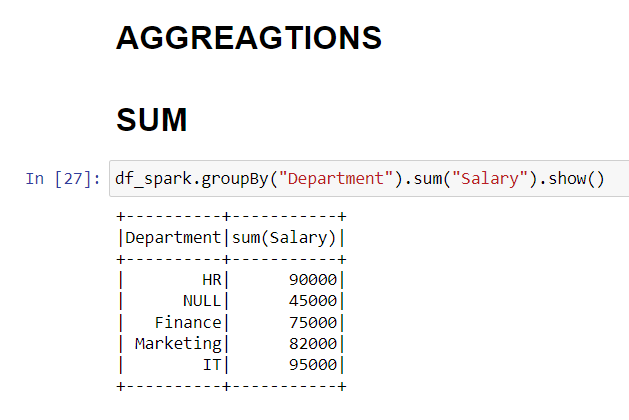
You can use either sort() or orderBy() function of PySpark DataFrame to sort DataFrame by ascending or descending order based on single or multiple columns. Both methods take one or more columns as arguments and return a new DataFrame after sorting

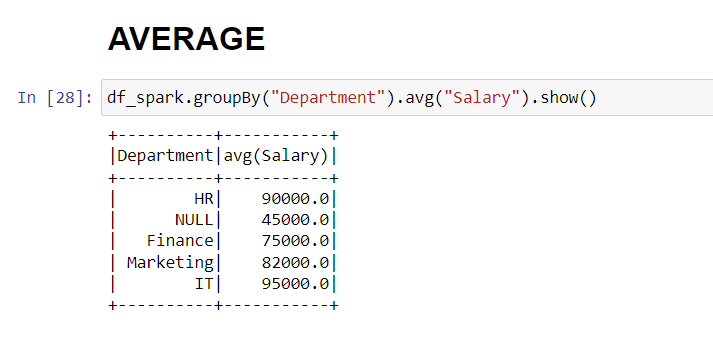


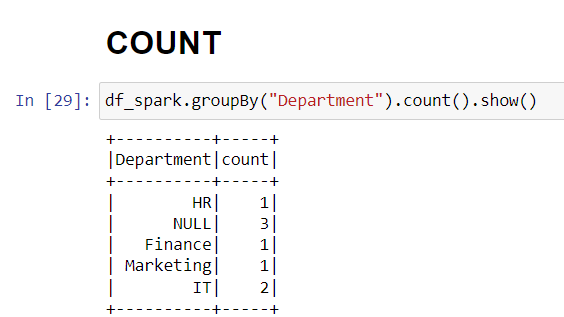


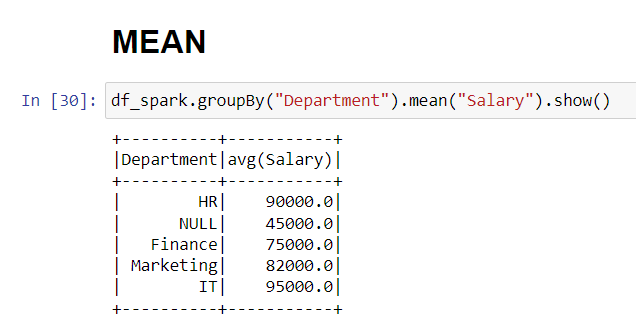
d)Aggregations

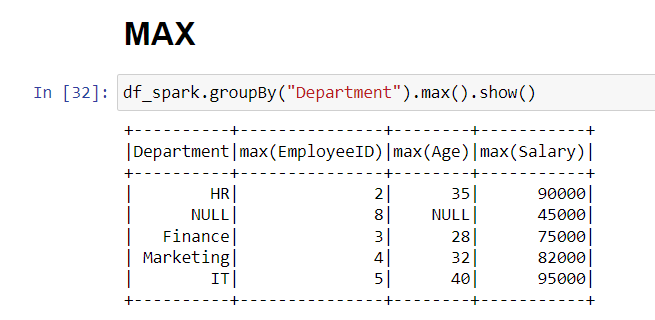
PySpark provides built-in standard Aggregate functions defined in DataFrame API, these come in handy when we need to make aggregate operations on DataFrame columns. Aggregate functions operate on a group of rows and calculate a single return value for every group

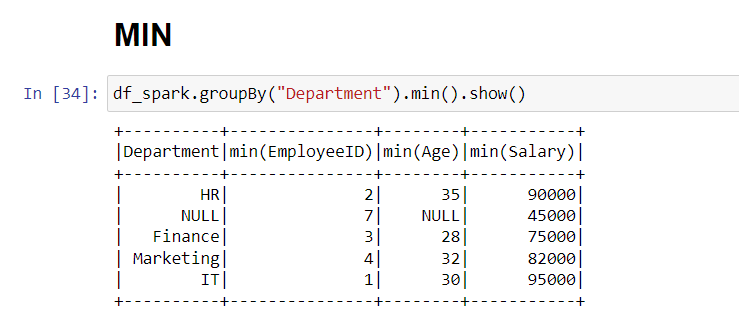






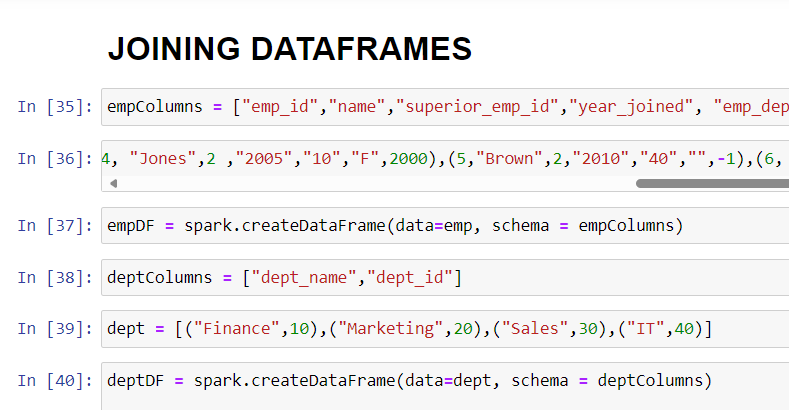






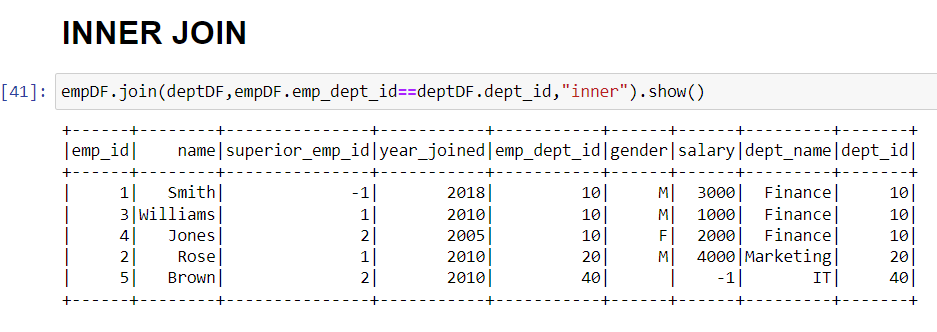
e)Joining DataFrames

**PySpark Join** is used to combine two DataFrames and by chaining these you can join multiple DataFrames; it supports all basic join type operations available in traditional SQL like INNER, LEFT OUTER, RIGHT OUTER, LEFT



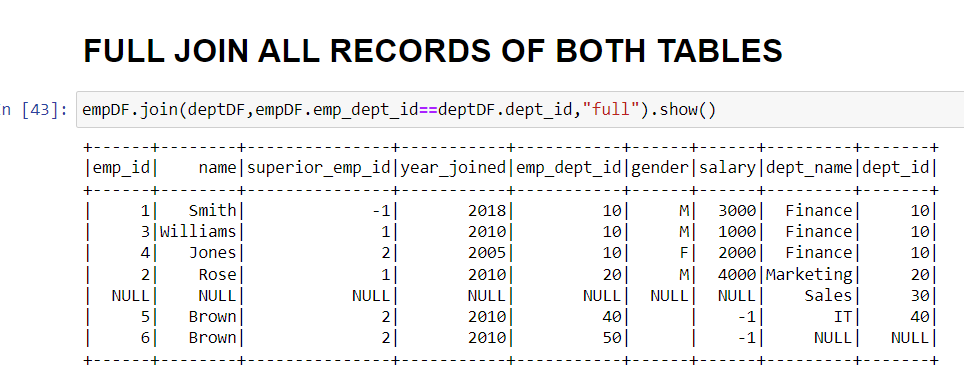
**INNER JOIN**

Returns only the rows with matching keys in both DataFrames.



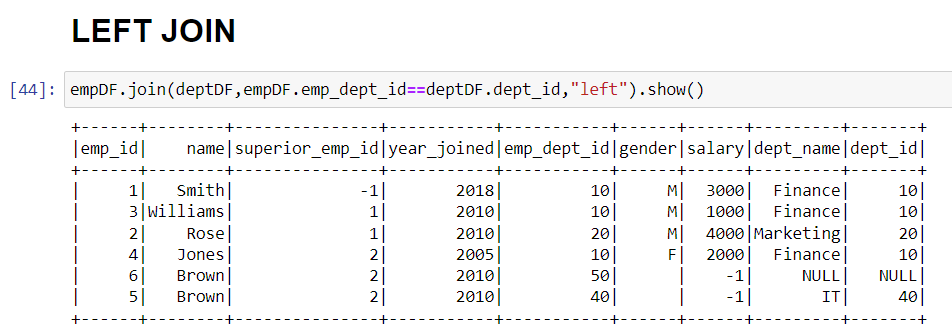
**FULL JOIN**

Returns all rows from both DataFrames, including matching and non-matching rows.

****

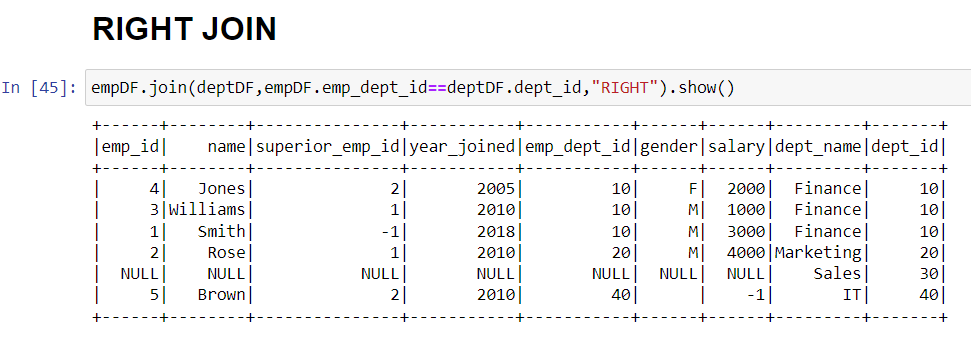
**LEFT JOIN**

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.

****

f)Group by

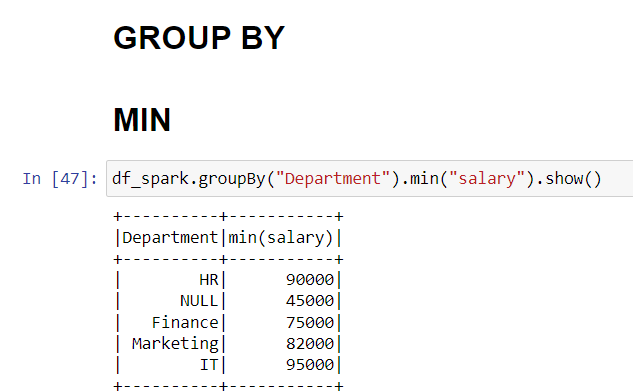
Similar to SQL GROUP BY clause, PySpark groupBy() function is used to collect the identical data into groups on DataFrame and perform count, sum, avg, min, max functions on the grouped data. In this article, I will explain several groupBy() examples using PySpark (Spark with Python).

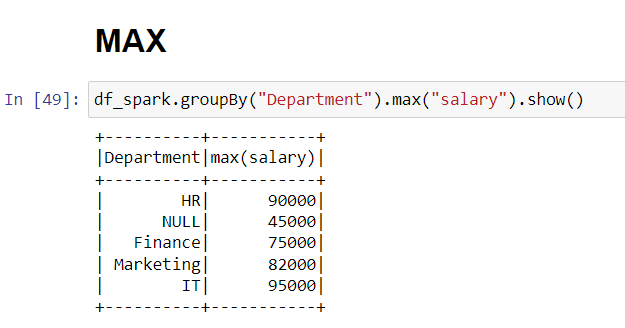
# Syntax

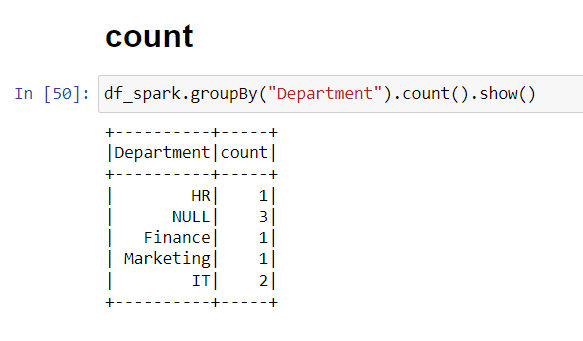
DataFrame.groupBy(\*cols)

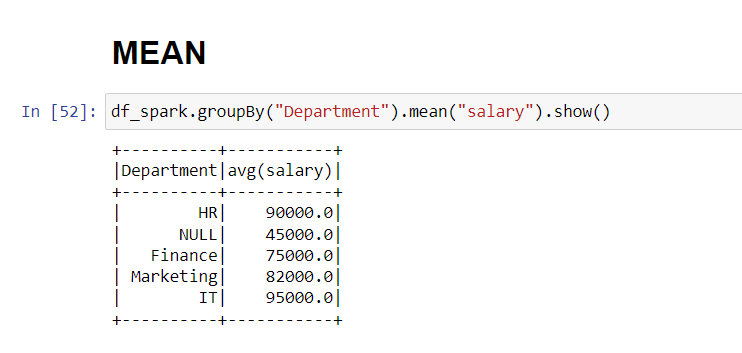
#or

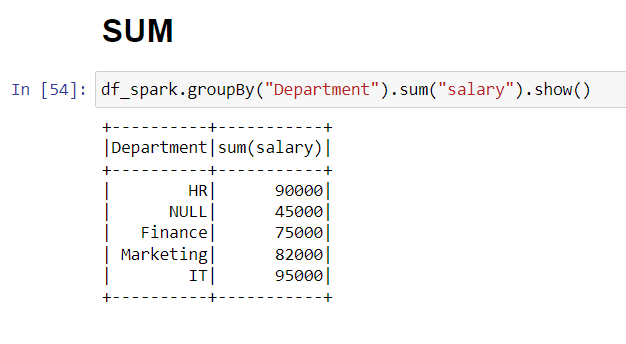
DataFrame.groupby(\*cols)

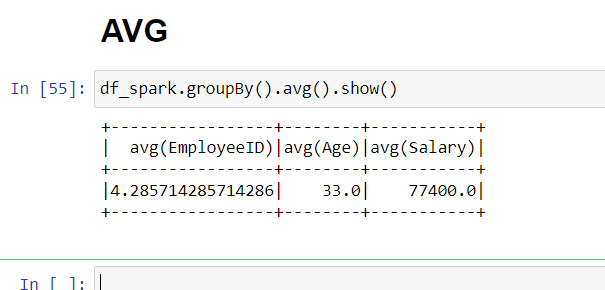






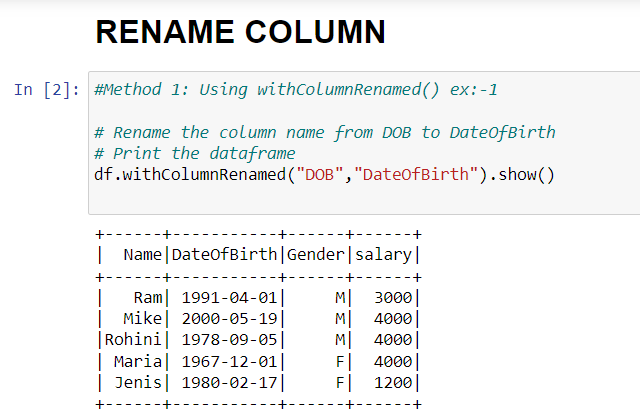


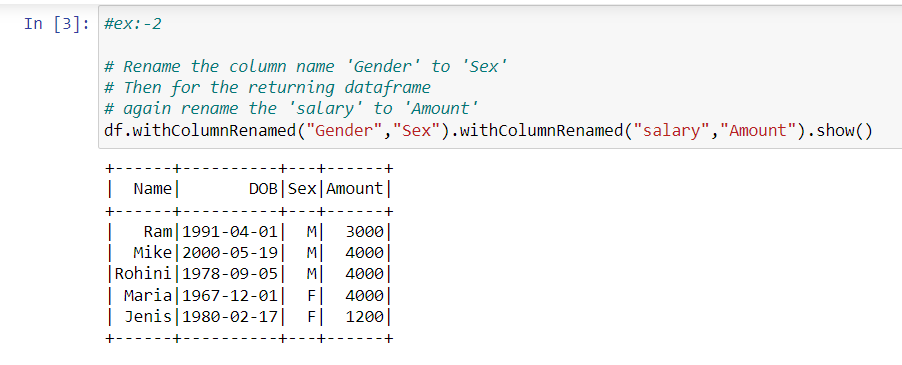


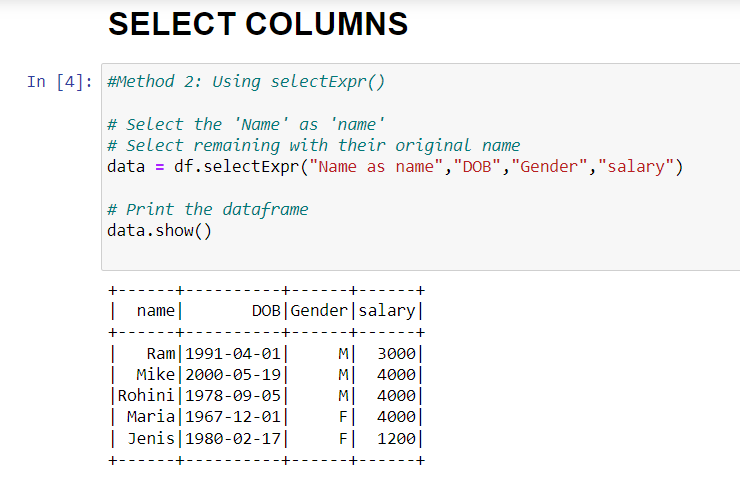


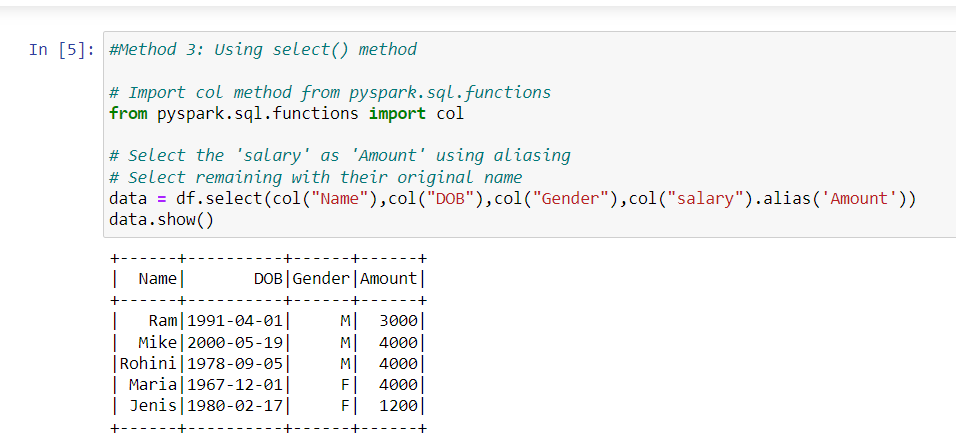
**Que 2) Applying Functions in a Pandas DataFrame**

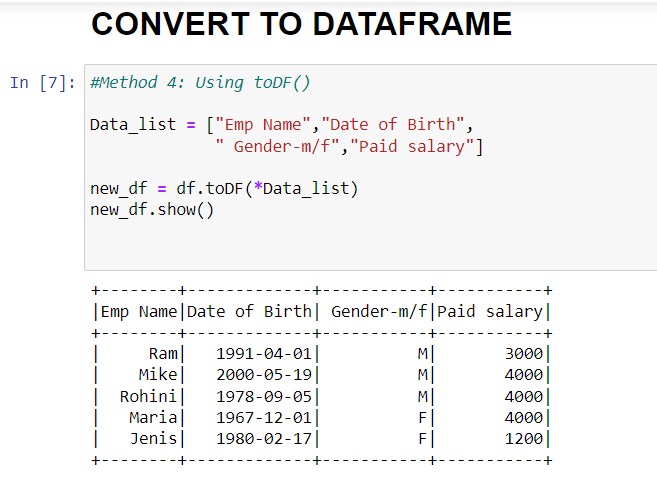












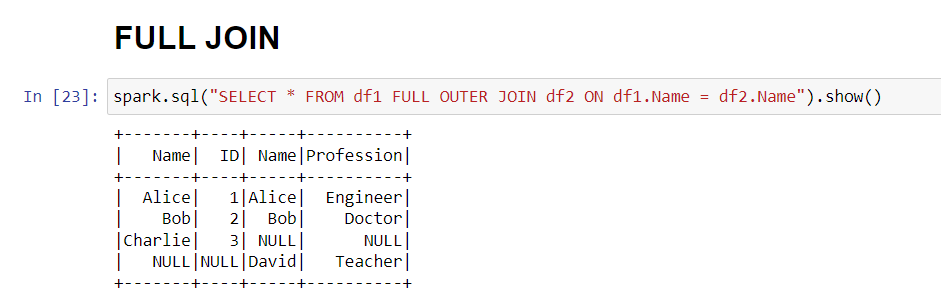
**Que 3) Execute Pyspark -sparksql joins**

## 

## 

## PySpark Full Outer Join

full, full outer join returns all rows from both datasets, where the join expression doesn’t match it returns null on respective record columns.

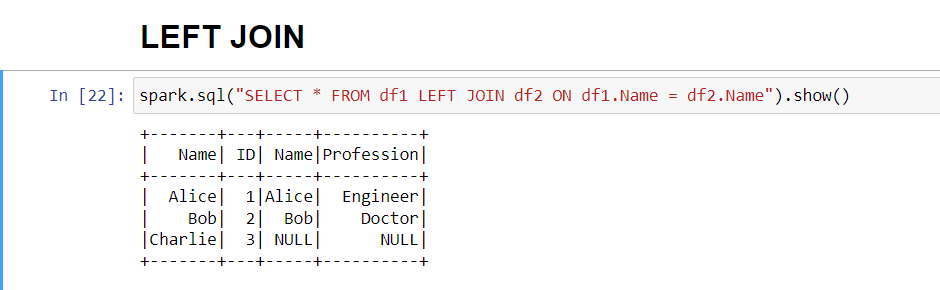


## 

## 

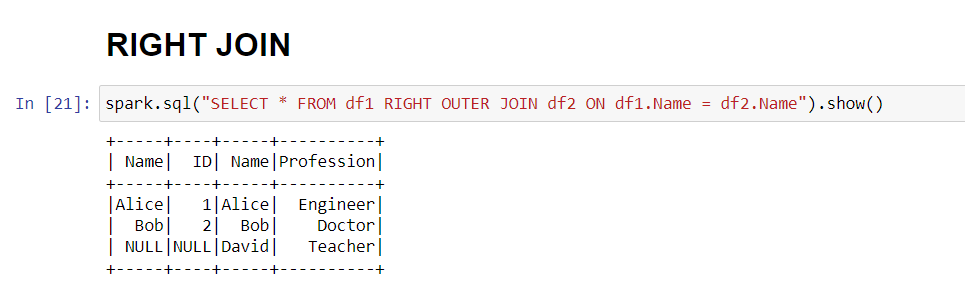
## PySpark left Join

Leftouter join returns all rows from the left dataset regardless of match found on the right dataset when join expression doesn’t match, it assigns null for that record and drops records from right where match not found.



## PySpark right Join

Rightouter join is opposite of left join, here it returns all rows from the right dataset regardless of match found on the left dataset, when join expression doesn’t match, it assigns null for that record and drops records from left where match not found.



## PySpark inner Join

Inner join is the default join in PySpark and it’s mostly used when you want to retrieve data from two or more DataFrames based on a shared key.

