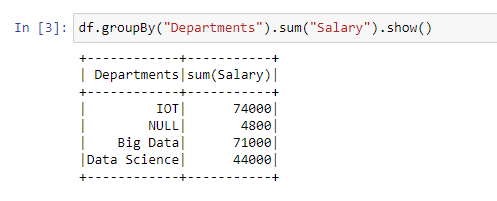
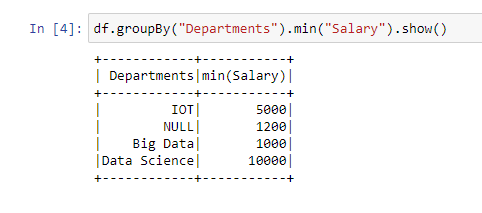
## Aggregations

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, and max functions on the grouped data.

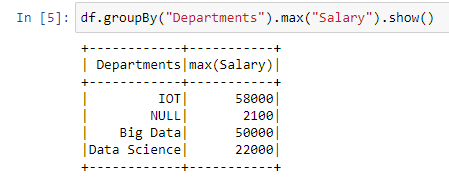
* groupBy() to find sum



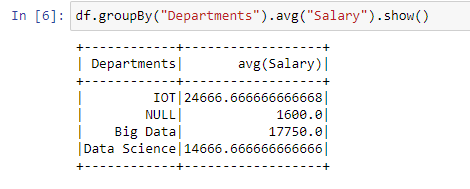
* groupBy() to find min



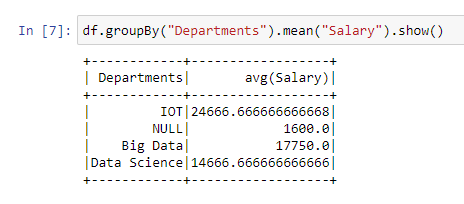
* groupBy() to find max



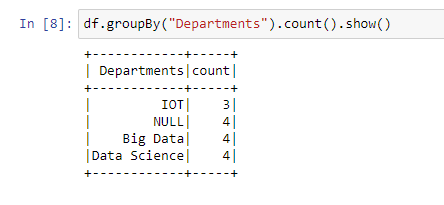
* groupBy() to find avg (average)



* groupBy() to find mean

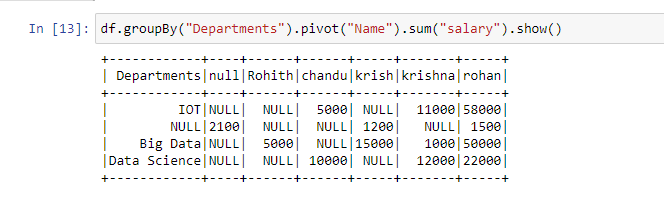


* groupBy() to find count

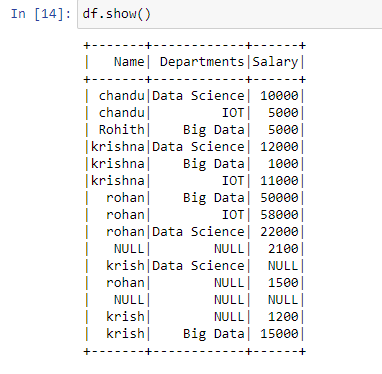


## Pivot

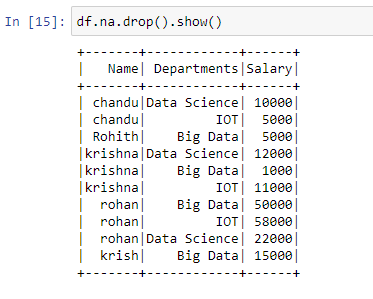
Using Pivot/ UnPivot — Spark SQL provides pivot() function to rotate the data from one column into multiple columns (transpose row to column). It is an aggregation where one of the grouping column values is transposed into individual columns with distinct data



## Handling Missing Values Pyspark



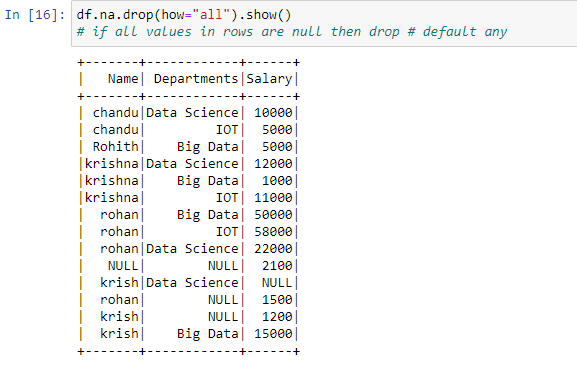
Dropping rows based on null values



drop() has the following parameters — how, thresh, and subset

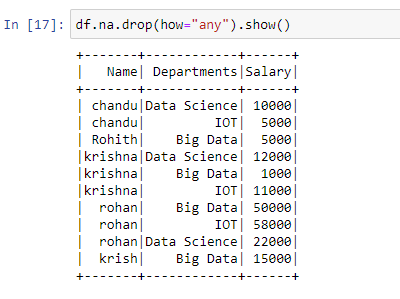
1. df\_pyspark1.na.drop(how="all").show()

# if all values in rows are null then drop # default any



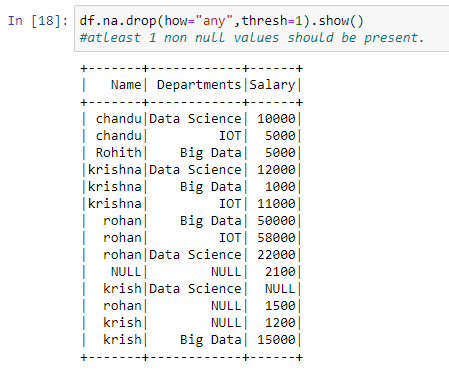
1. df\_pyspark1.na.drop(how="any",thresh=2).show()

#atleast 2 non null values should be present.

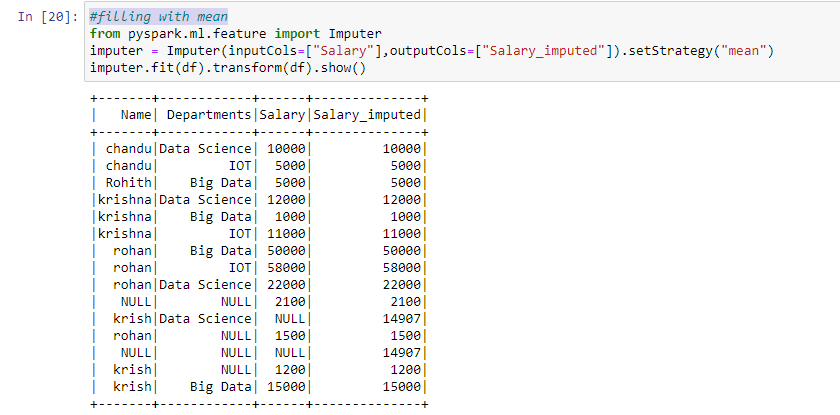


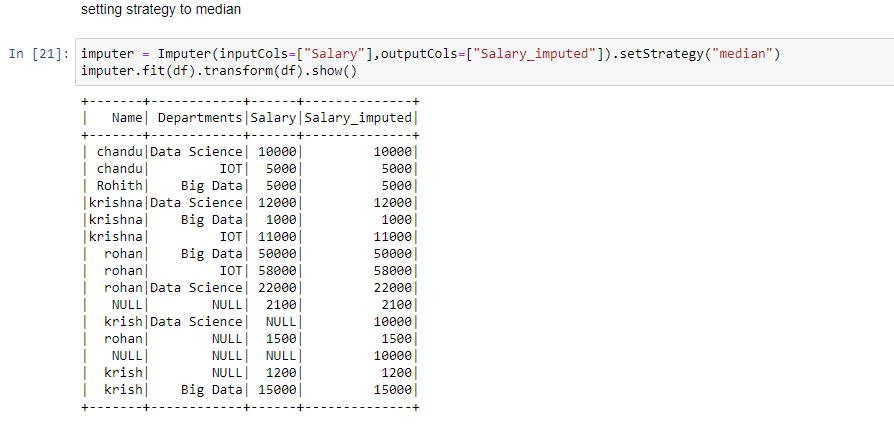
1. df\_pyspark1.na.drop(how="any",subset=["salary"]).show()

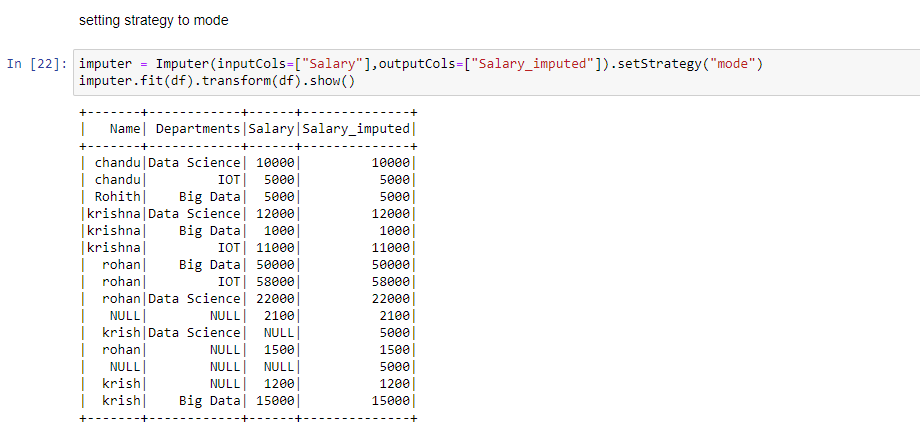
# only in that column rows get deleted



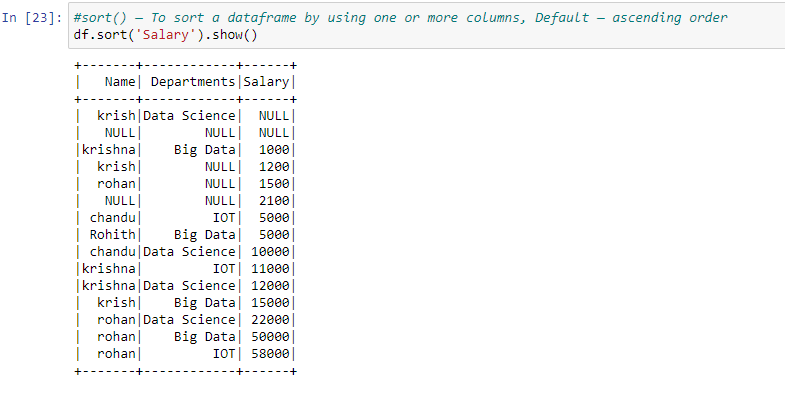
## Filling missing values using Mean, Median, or Mode with help of the Imputer function

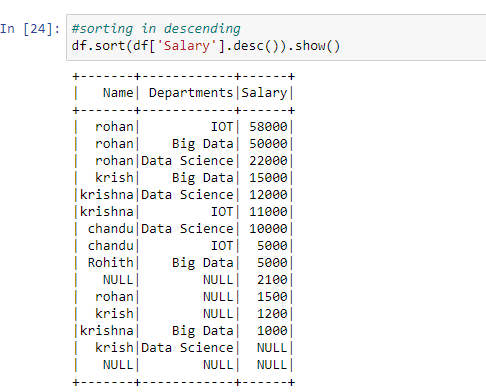


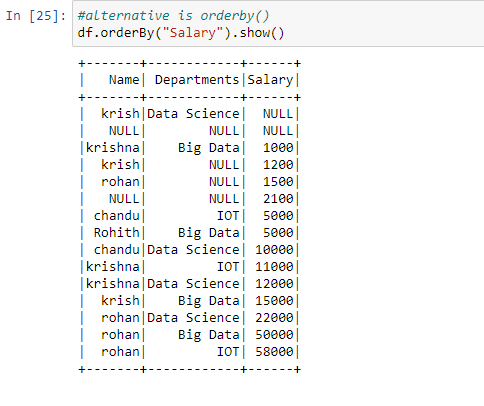




## orderBy() and sort() in Pyspark DataFrame





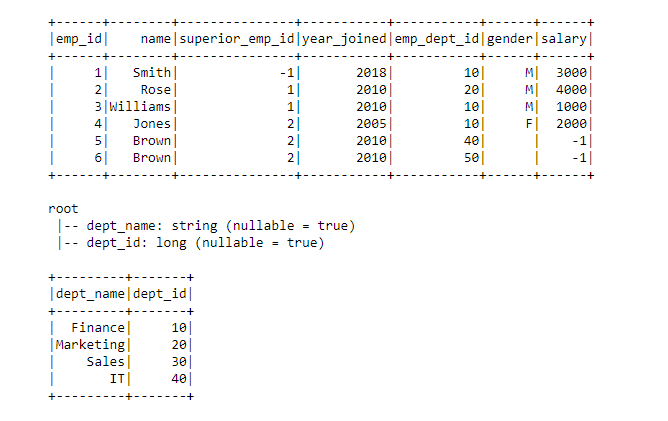


**join() using pyspark:**

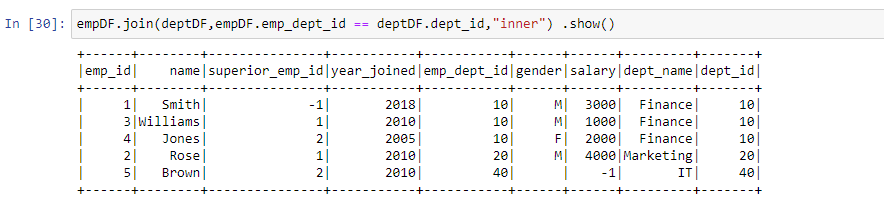
PySpark Join is used to combine two DataFrames and by chaining these you can join multiple DataFrames; it supports all basic join type operations available in traditional SQL like INNER, LEFT OUTER, RIGHT OUTER, LEFT ANTI, LEFT SEMI, CROSS, SELF join.

 creating two dataframes:

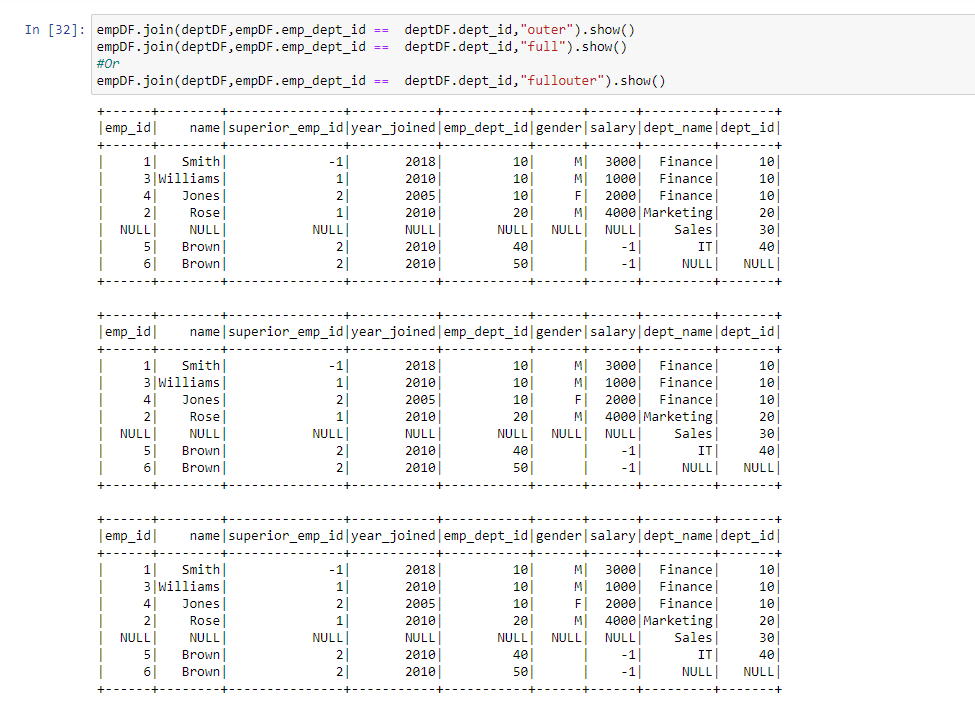




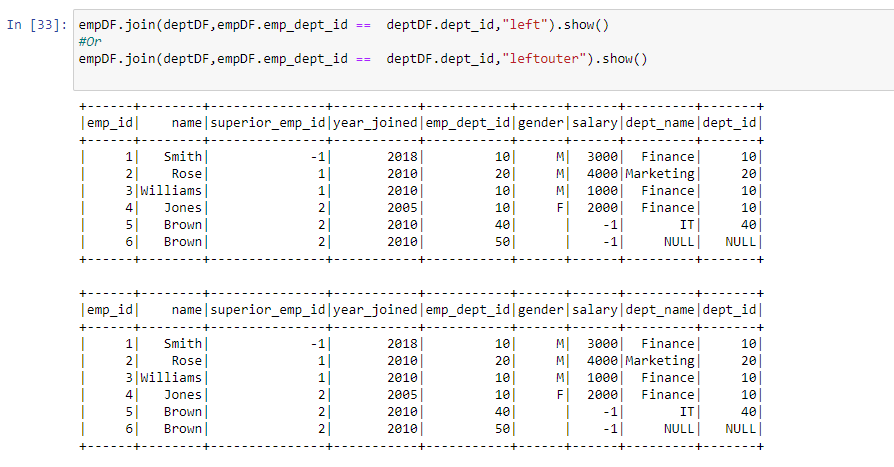
1. Inner join



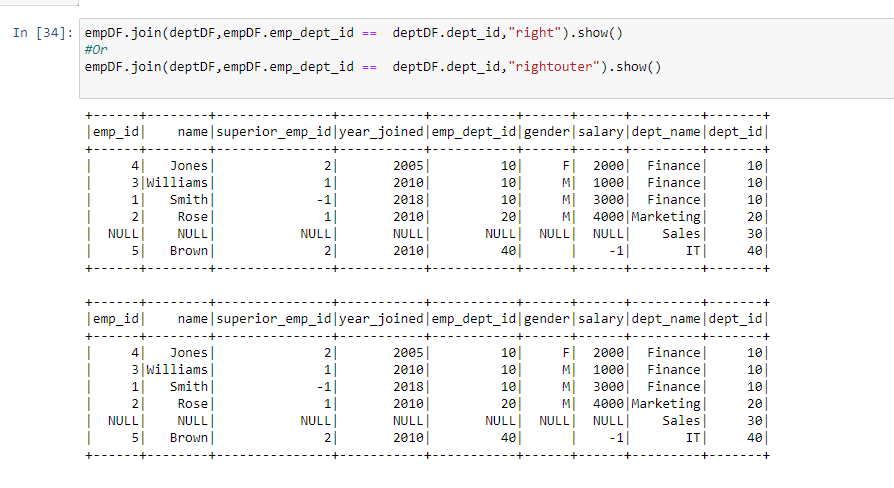
1. Outer / full / fullouter join



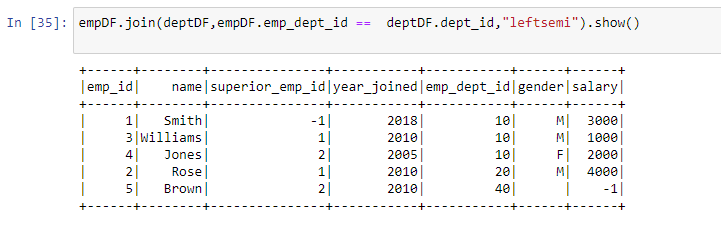
1. left join



1. right join



1. Left semi join



1. Left anti join

