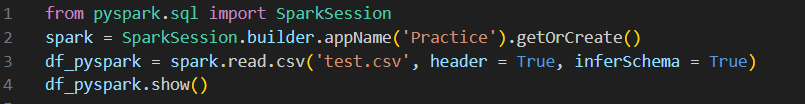
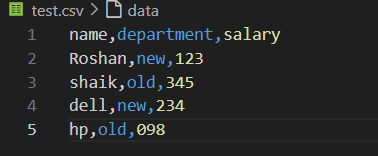
**Group By aggregations**



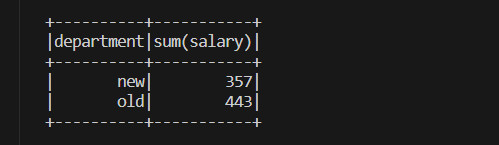
Test.csv



Sum() – to calculate sum of data in a column



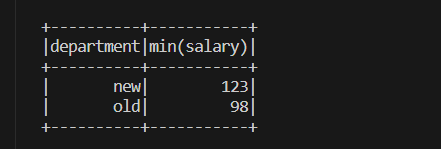
Output:



Min() - to calculate min of data in a column



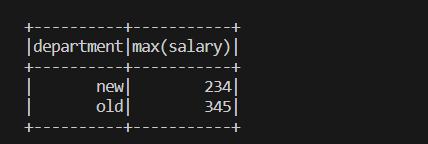
Output:



Max() - to calculate max of data in a column



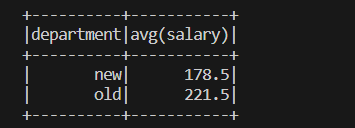
Output:



Avg() - to calculate average of data in a column



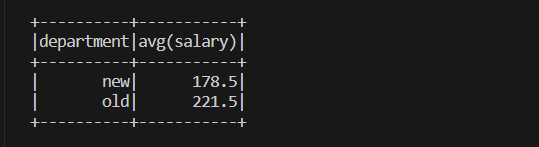
Output:



Mean() - to calculate mean of data in a column



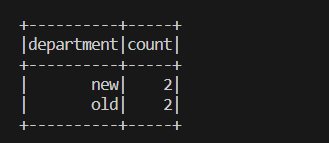
Output:



Count() - to calculate count of records in a column



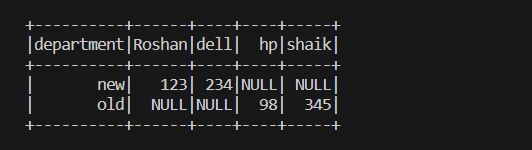
Output:



**Pivot()** - 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. Similary UnPivot can be used.

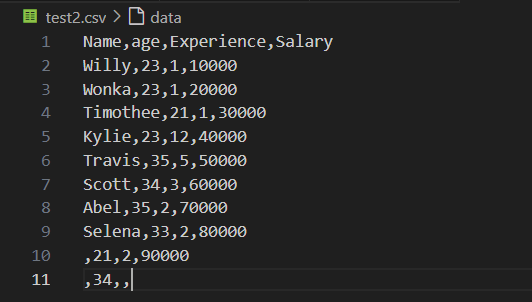


Output:



Handling Missing Values Pyspark

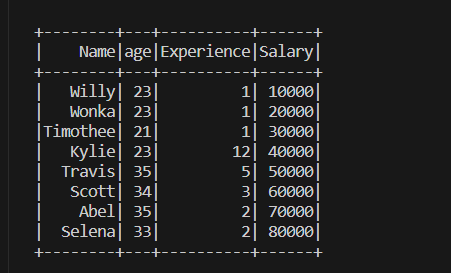
Test2.csv



Dropping rows based on null values



Output:

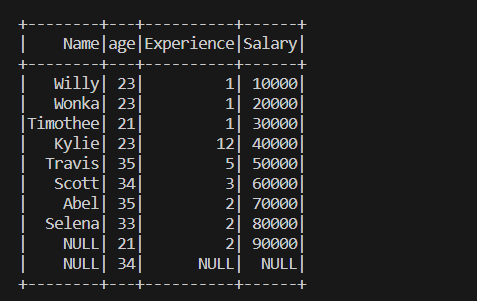


drop() has the following parameters — how, thresh, and subset

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



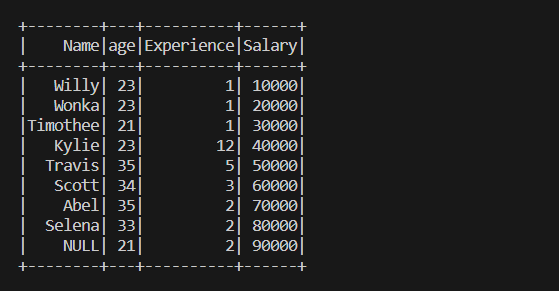
Output:



atleast 2 non null values should be present.



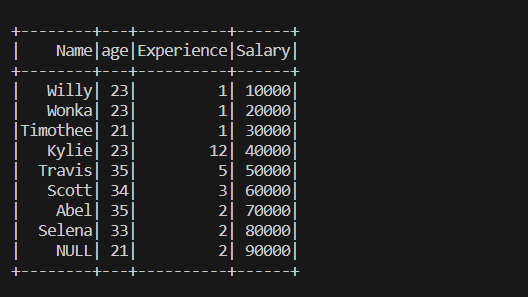
Output:



only in salary column rows get deleted



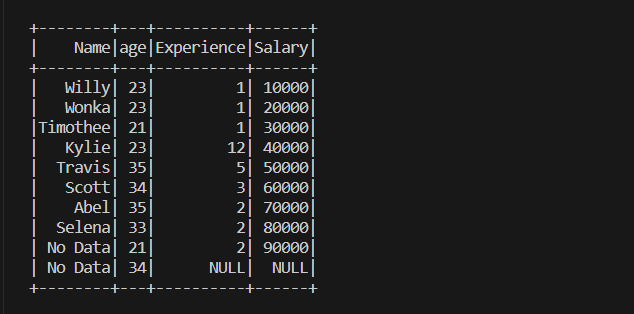
Output:



string values will get replaced as string is given as input



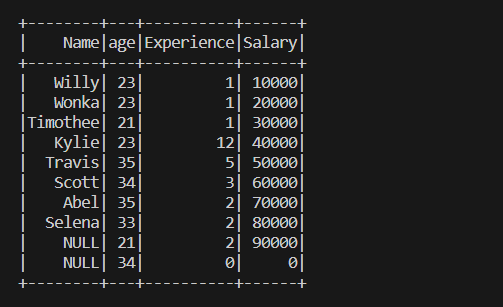
Output:



integer values will get replaced as integer is given as input



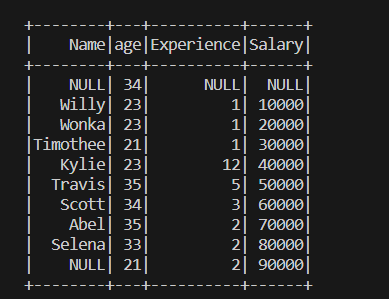
Output:



sort() in Pyspark DataFrame



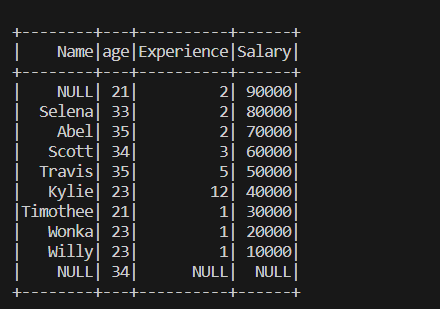
Output:



sort based on descending order



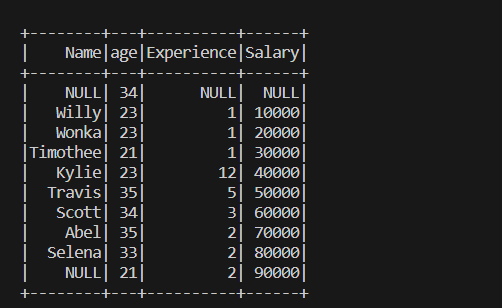
Output:



Sort based on first column then second column



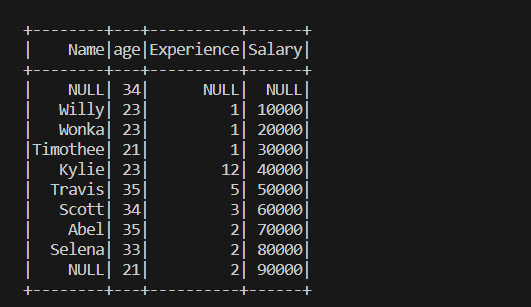
Output:



Sort based on single column and using orderby

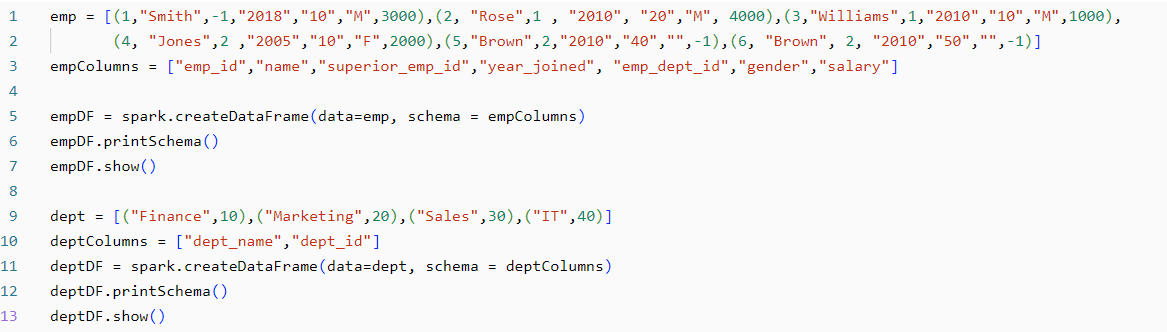


Output:

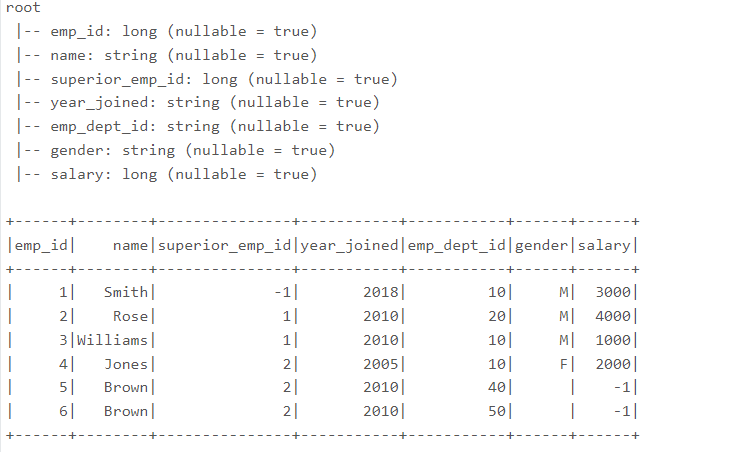


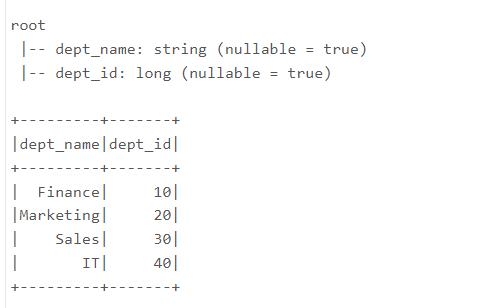
**Joins**

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.



Output:

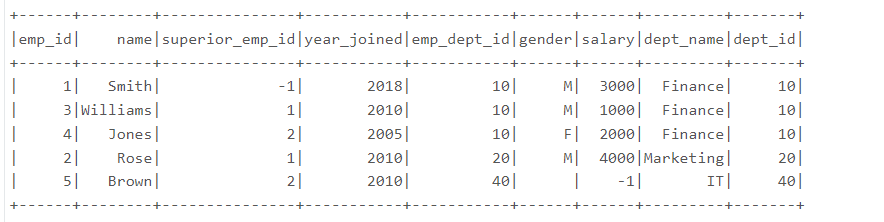




Inner join - Matches and retrieves common values from two tables



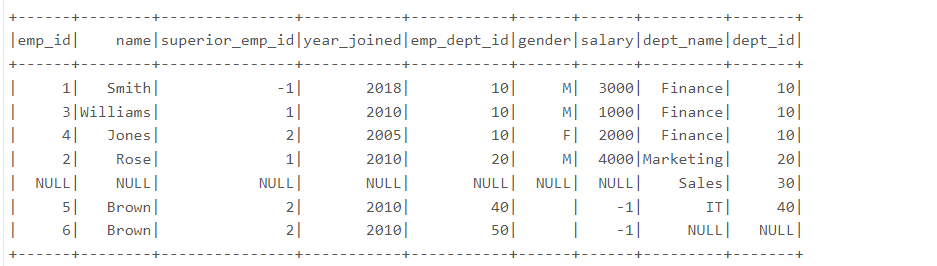
Output:



Outer join - Retrieves matched and unmatched values from two tables.



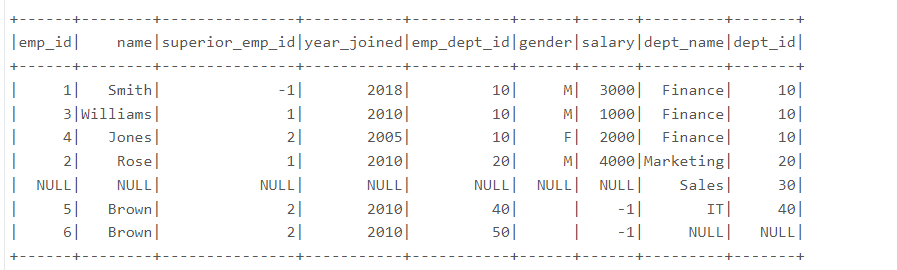
Output:



Full join - Retrieves all values from both tables, matching and non-matching



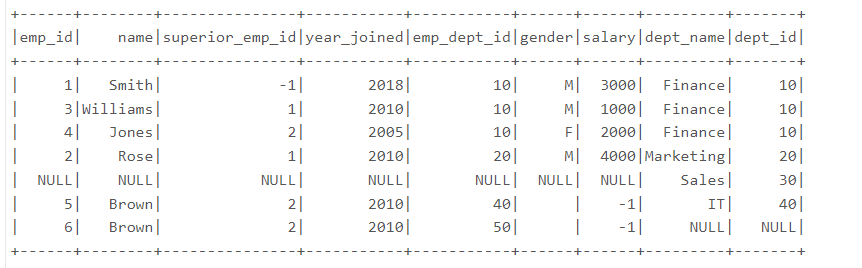
Output:



Full outer join - Retrieves all values, matching and non-matching, from both tables.



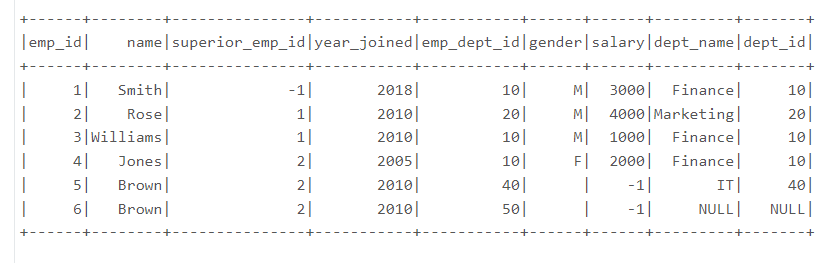
Output:



Left join - Retrieves all values from the left table and matching values.



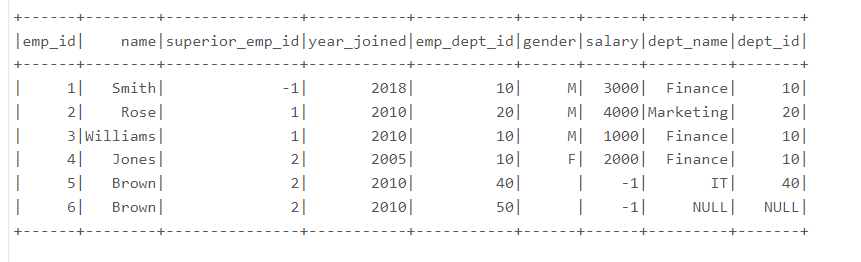
Output:



Left outer join - Retrieves all values from the left table and matching values.



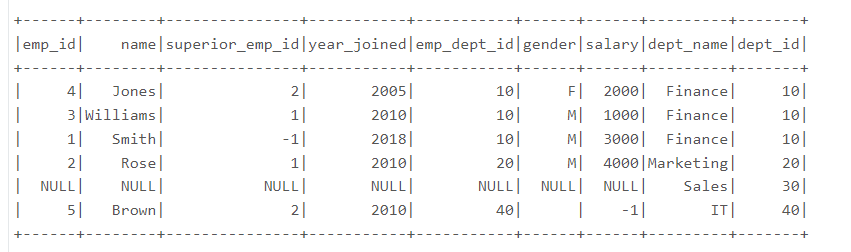
Output:



Right join - Retrieves all values from the right table and matching values.



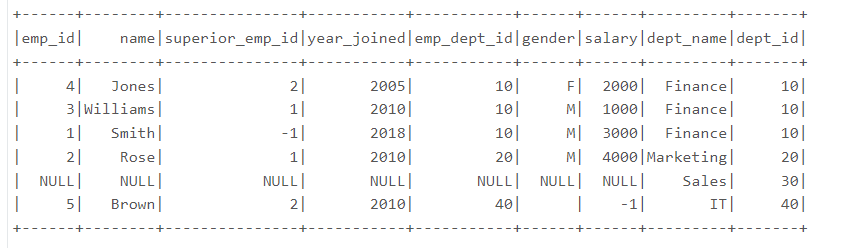
Output:



Right outer join - Retrieves all values from the right table and matching values.



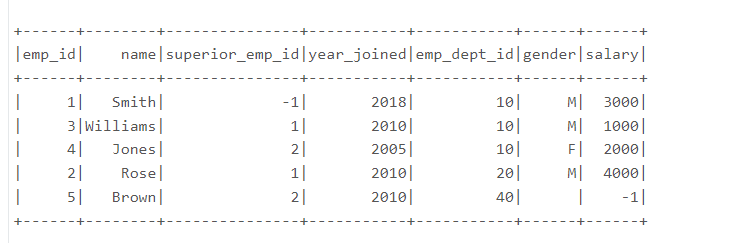
Output:



Left semi join - Returns rows from the left table with matching right table.



Output:



Left anti join - Returns rows from the left table with non-matching right table.



Output:

