누구나 빅데이터 처리 할 수 있는

APACHE PySpark™

Part 3

groupBy (Max, Min, Count,Sum,Avg)

df.groupBy("department").**count().**show()

df.groupBy("department").**sum("salary").**show()

df.groupBy("department").**min("salary").**show()

df.groupBy("department").**max("salary").**show()

df.groupBy("department").**avg("salary").**show()

df.join

```
from pyspark.sql import SparkSession

# df1하고 df2에 있는 column1 합치기

joined_df = df1.join(df2, df1.column1 == df2.column1)

joined_df
```

df.join

Join String	Equivalent SQL Join
inner	INNER JOIN
outer, full, fullouter, full_outer	FULL OUTER JOIN
left, leftouter, left_outer	LEFT JOIN
right, rightouter, right_outer	RIGHT JOIN
cross	
anti, leftanti, left_anti	
semi, leftsemi, left_semi	

df.union

```
from pyspark.sql import SparkSession

# df1하고 df2 데이터프레임 합치기

union_df = df1.union(df2)

union_df.show()
```

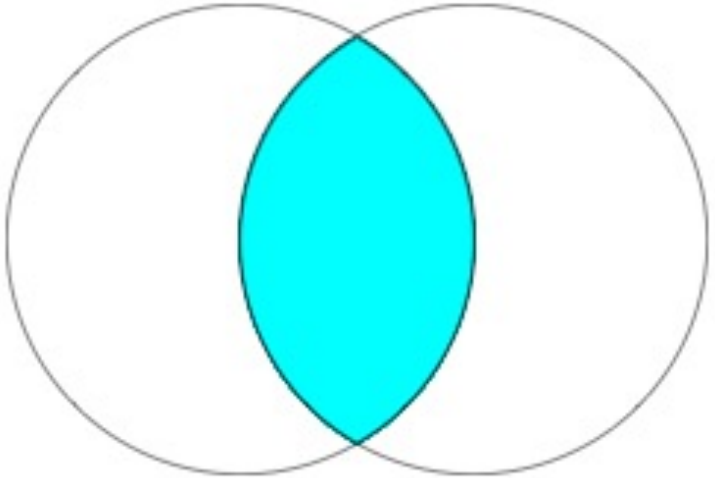
df.intersect

```
from pyspark.sql import SparkSession

# df1하고 df2 교차하기

intersect_df = df1.intersect(df2)

Intersect_df
```



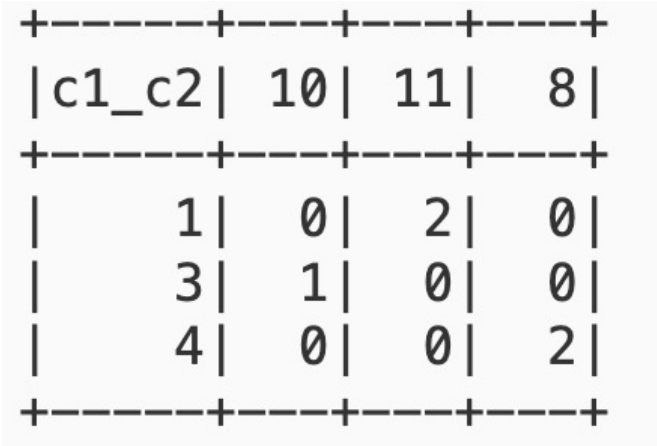
df.crosstab

```
from pyspark.sql import SparkSession

# df1하고 df2 교차표

crosstab_df = df.crosstab("column1","column2")

crosstab_df.show()
```



c1_c2	10	11	8
1	0	2	0
3	1	0	0
4	0	0	2

df.dtypes & columns

```
from pyspark.sql import SparkSession
# 데이터프레임의 데이터 type
dtypes = df.dtypes
print(dtype)

# 컬럼 이름 리스트
columns = df.columns
print(columns)
```