① For querring the SQL Syntax in Pyspark use temp view

② ASC, DESC → df.Sort(df.Name asc()).Show(

③ CAST → Convert data type

df.Select(df.Salary.Cast('int'))

④ Like → operator

df. filter.(df.name ('S%'))

⑤ where → Condition's

df.filter((df.Id =='1') & (df.name.like('S%')).Show

⑥ Distinct

df.distinct().Show()

⑦ Drop Duplicates

df.dropDuplicates().Show()

ⓧ from whole table

for one Column

df.dropDuplicates(['Emp_ID']).show()

⑧ union → remove duplicates from rows

Combine two Data frames

unionAll → Don't remove the dublicates

Combine two Data frames

Combine two Data frame

```
df3 = df1. union ALL (df2)
```

union By name → Combine two Data frame based on their Column name

(9) Joins → Types of Joins

```
Df1 . Join (Df2 , Df1 . dep == Df2 . ID, 'Inner') . Show()
```

(10) upper → function

```
df . withColumn ('Name' , upper (df. name)). Show()
```

(11) Temp View → we can use this with in a session

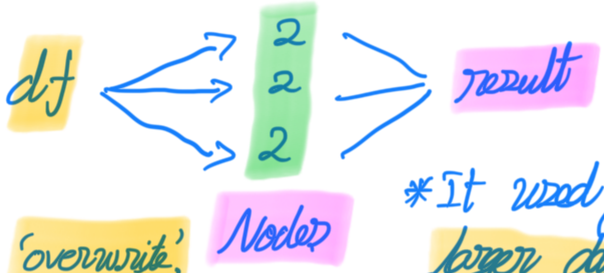
```
df. CreateOrReplaceTempView ('Employee')
df1 = Spark . Sql ("Select * from Employee")
```

v/s

(12) Global temp View → we can use this across the session

```
df. Create or Replace Global Temp View ('Global_Emp')
df1 = Spark. Sql ("Select * from Global_temp. Global_Emp")
```

(13) partition by →



```
df. write. parquet ('path', mode = 'overwrite',
            partitionby = ['Dep'])
```

*It used for larger data set to process fast

(14) Date format →

```
df.withColumn('date_format', date_format(lit('2024-12-21'),
                                  'yyyy.MM.dd')).show()
```