databricksScala_spark

```
import org.apache.spark.SparkContext
import org.apache.spark.SparkConf

import org.apache.spark.SparkContext
import org.apache.spark.SparkConf

val sc = SparkContext.getOrCreate()

sc: org.apache.spark.SparkContext = org.apache.spark.SparkContext@a2d48ac

val dataFrame =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/FileStore/tables/fruits.text")


dataFrame: org.apache.spark.sql.DataFrame = [id: int, name: string]

print(dataFrame)

[id: int, name: string]

dataFrame.collect()

res22: Array[org.apache.spark.sql.Row] = Array([1,Apple], [2,Banana], [3,Ora
nge], [4,Grapes])

display(dataFrame)
```

	id	name
1	1	Apple
2	2	Banana
3	3	Orange
4	4	Grapes

Showing all 4 rows.

```
val dF =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/FileStore/tables/animal-1.text","/FileStore/tables/country-
1.text","/FileStore/tables/fruits-2.text","/FileStore/tables/city-1.text")

dF: org.apache.spark.sql.DataFrame = [id: int, name: string]

display(dF)
```

	id	name
1	1	Monkey
2	2	Donkey
3	3	Horse
4	4	Gorilla
5	1	Apple
6	2	Banana
7	3	Orange

Showing all 16 rows.

```
dF.collect()
```

```
res37: Array[org.apache.spark.sql.Row] = Array([1,Monkey], [2,Donkey], [3,Ho
rse], [4,Gorilla], [1,Apple], [2,Banana], [3,Orange], [4,Grapes], [1,Texas],
[2,Dallas], [3,Miami], [4,Chicago], [1,UK], [2,USA], [3,Mexico], [4,Canada])
```

```
dF.show()
```

```
+---+-------+
| id|   name|
+---+-------+
|  1| Monkey|
|  2| Donkey|
|  3|  Horse|
|  4|Gorilla|
|  1|  Apple|
|  2| Banana|
|  3| Orange|
|  4| Grapes|
|  1|  Texas|
|  2| Dallas|
|  3|  Miami|
|  4|Chicago|
|  1|     UK|
|  2|    USA|
|  3| Mexico|
|  4| Canada|
+---+-------+
```

```
val data=sc.textFile("/FileStore/tables/Names-1.text")
```

```
data: org.apache.spark.rdd.RDD[String] = /FileStore/tables/Names-1.text MapP
artitionsRDD[38] at textFile at command-3355079518026507:1
```

```
data.collect()
```

```
res48: Array[String] = Array(Arpit, Sowmya, Jigyasa, Thiru, Mahesh, Arun, Sa
urav, Lalith, Keerthana, Nikita, Aiswarya, Nency, Kalyan)
```

```
val splitdata = data.flatMap(line => line.split(" "));
```

```
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[129] at flatM
ap at command-2826642787694262:1
```

```
val output = splitdata.filter{word=> word.startsWith("A")}
```

```
output: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[130] at filter a
t command-2826642787694263:1
```

```
 output.collect()
```

```
res49: Array[String] = Array(Arpit, Arun, Aiswarya)
```

```
val mapdata = output.map(word => (word,1));
mapdata.collect()
```

```
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[131] at
map at command-2826642787694265:1
res50: Array[(String, Int)] = Array((Arpit,1), (Arun,1), (Aiswarya,1))
```

```
val count=mapdata.count()
```

```
count: Long = 3
```

```
res51: Array[(String, Int)] = Array((Arun,1), (Arpit,1), (Aiswarya,1))
```

```
val order=sc.textFile("/FileStore/tables/orders.text")
```

```
order: org.apache.spark.rdd.RDD[String] = /FileStore/tables/orders-1.text Ma
pPartitionsRDD[142] at textFile at command-2826642787694268:1
```

```
order.collect()
```

```
res56: Array[String] = Array(102,2009-10-08 00:00:00,3,3000, 100,2009-10-08
00:00:00,3,1500, 101,2009-11-20 00:00:00,2,1560, 103,2008-05-20 00:00:00,4,2
060)
```

```
val order =
spark.read.format("csv").option("header","true").option("inferSchema",
"true").load("/FileStore/tables/orders.text")
```

```
order: org.apache.spark.sql.DataFrame = [OrderId: int, OrderDate: string ...
2 more fields]
```

```
order.show()
```

```
+-------+-------------------+------+------+
|OrderId|          OrderDate|CustId|Amount|
+-------+-------------------+------+------+
|    102|2009-10-08 00:00:00|     3|  3000|
|    100|2009-10-08 00:00:00|     3|  1500|
|    101|2009-11-20 00:00:00|     2|  1560|
|    103|2008-05-20 00:00:00|     4|  2060|
+-------+-------------------+------+------+
```

```
order.registerTempTable("order")
sqlContext.sql("SELECT CustId, COUNT(*) AS cnt FROM order  GROUP BY
custId").show()
```

```
+------+---+
|CustId|cnt|
+------+---+
|     3|  2|
|     4|  1|
|     2|  1|
+------+---+
```

```
command-2826642787694272:1: warning: method registerTempTable in class Datas
et is deprecated (since 2.0.0): Use createOrReplaceTempView(viewName) instea
d.
order.registerTempTable("order")
      ^
```

```
val states = Map(("BR","BIHAR"),("MH","MAHARASHTRA"),("TN","TAMILNADU"),
("AP","ANDRAPRADESH"),("KL","KERALA"))
  val broadcastStates = spark.sparkContext.broadcast(states)
  val data = Seq(
    ("Amit","Mishra","BR"),
    ("Nitin","Kulkarni","MH"),
    ("Ram","Kumar","TN"),
    ("Kesav","Prasad","AP"),
    ("Biju","Joseph","KL"),
    ("Ravi","Teja","AP")
  )
  val rdd = spark.sparkContext.parallelize(data)

  val rdd2 = rdd.map(f=>{
    val state = f._3
    val fullState = broadcastStates.value.get(state).get
    (f._1,f._2,fullState)
  })

  println(rdd2.collect().mkString("\n"))
```

```
(Amit,Mishra,BIHAR)
(Nitin,Kulkarni,MAHARASHTRA)
(Ram,Kumar,TAMILNADU)
(Kesav,Prasad,ANDRAPRADESH)
(Biju,Joseph,KERALA)
(Ravi,Teja,ANDRAPRADESH)
states: scala.collection.immutable.Map[String,String] = Map(MH -> MAHARASHTR
A, TN -> TAMILNADU, KL -> KERALA, BR -> BIHAR, AP -> ANDRAPRADESH)
broadcastStates: org.apache.spark.broadcast.Broadcast[scala.collection.immut
able.Map[String,String]] = Broadcast(21)
data: Seq[(String, String, String)] = List((Amit,Mishra,BR), (Nitin,Kulkarn
i,MH), (Ram,Kumar,TN), (Kesav,Prasad,AP), (Biju,Joseph,KL), (Ravi,Teja,AP))
rdd: org.apache.spark.rdd.RDD[(String, String, String)] = ParallelCollection
RDD[26] at parallelize at command-2826642787694273:11
rdd2: org.apache.spark.rdd.RDD[(String, String, String)] = MapPartitionsRDD
[27] at map at command-2826642787694273:13
```

```
+-----+--------+---+
| Amit|  Mishra| BR|
+-----+--------+---+
|Nitin|Kulkarni| MH|
|  Ram|   Kumar| TN|
|Kesav|  Prasad| AP|
| Biju|  Joseph| KL|
| Ravi|    Teja| AP|
+-----+--------+---+
```

```
broadcastStates: org.apache.spark.broadcast.Broadcast[Array[String]] = Broad
cast(20)
```

```
accum: org.apache.spark.util.LongAccumulator = LongAccumulator(id: 490, nam
e: Some(My Accumulator), value: 0)
```

```
val data=Array(1, 2, 3, 4)
val accum = sc.longAccumulator("My Accumulator")
sc.parallelize(data).foreach(x => accum.add(x))
accum.value
```

```
data: Array[Int] = Array(1, 2, 3, 4)
accum: org.apache.spark.util.LongAccumulator = LongAccumulator(id: 522, nam
e: Some(My Accumulator), value: 10)
res23: Long = 10
```

```
val data =Seq (("Chennai", "20000000"), ("Mumbai", "50000000"), ("Delhi",
"40000000"))
val columns = Seq("city","humanCount")
  import spark.sqlContext.implicits._
  val df = data.toDF(columns:_*)
```

```
data: Seq[(String, String)] = List((Chennai,20000000), (Mumbai,50000000), (D
elhi,40000000))
columns: Seq[String] = List(city, humanCount)
import spark.sqlContext.implicits._
df: org.apache.spark.sql.DataFrame = [city: string, humanCount: string]
```

```
df.show()
```

```
+-------+----------+
|   city|humanCount|
+-------+----------+
|Chennai|  20000000|
| Mumbai|  50000000|
|  Delhi|  40000000|
+-------+----------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Name", StringType,true),
                 StructField("Skillset", ArrayType(StringType),true),
                 StructField("State", StringType,true),
             ))
val data =Seq(
    Row("Amit,,Mishra",List("Java","Scala","C++"),"UP"),
    Row("Prabhu,Ram,",List("Spark","Java","C++"),"TN"),
    Row("Ramesh,Kumar",List("CSharp","VB"),"TN"))
var dfFromData3 =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData3.show()
```

```
+------------+------------------+-----+
|        Name|          Skillset|State|
+------------+------------------+-----+
|Amit,,Mishra|[Java, Scala, C++]|   UP|
| Prabhu,Ram,|[Spark, Java, C++]|   TN|
|Ramesh,Kumar|      [CSharp, VB]|   TN|
+------------+------------------+-----+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Name,
StringType,true), StructField(Skillset,ArrayType(StringType,true),true), Str
uctField(State,StringType,true))
data: Seq[org.apache.spark.sql.Row] = List([Amit,,Mishra,List(Java, Scala, C
++),UP], [Prabhu,Ram,,List(Spark, Java, C++),TN], [Ramesh,Kumar,List(CSharp,
VB),TN])
dfFromData3: org.apache.spark.sql.DataFrame = [Name: string, Skillset: array
<string> ... 1 more field]
```

```
  command-3475293752828625:3: error: not found: value ArrayType
      .add("SkillSet", ArrayType(StringType))
                       ^
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Empno", IntegerType,true),
                 StructField("Ename",StringType,true),
                 StructField("Job", StringType,true),
                 StructField("Sal", IntegerType,true),
                 StructField("Deptno", IntegerType,true),
             ))
val data =Seq(
    Row(7369,"SMITH","CLERK",800,20),
    Row(7499,"ALLEN","SALESMAN",1600,30),
    Row(7521,"WARD","SALESMAN",1250,30),
    Row(7566,"JONES","MANAGER",2975,20),
    Row(7654,"MARTIN","SALESMAN",1400,30),
    Row(7698,"BLAKE","MANAGER",2850,30)
)
var dfFromData =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData.show()
```

```
+-----+------+--------+----+------+
|Empno| Ename|     Job| Sal|Deptno|
+-----+------+--------+----+------+
| 7369| SMITH|   CLERK| 800|    20|
| 7499| ALLEN|SALESMAN|1600|    30|
| 7521|  WARD|SALESMAN|1250|    30|
| 7566| JONES| MANAGER|2975|    20|
| 7654|MARTIN|SALESMAN|1400|    30|
| 7698| BLAKE| MANAGER|2850|    30|
+-----+------+--------+----+------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Empn
o,IntegerType,true), StructField(Ename,StringType,true), StructField(Job,Str
ingType,true), StructField(Sal,IntegerType,true), StructField(Deptno,Integer
Type,true))
data: Seq[org.apache.spark.sql.Row] = List([7369,SMITH,CLERK,800,20], [7499,
ALLEN,SALESMAN,1600,30], [7521,WARD,SALESMAN,1250,30], [7566,JONES,MANAGER,2
975,20], [7654,MARTIN,SALESMAN,1400,30], [7698,BLAKE,MANAGER,2850,30])
dfFromData: org.apache.spark.sql.DataFrame = [Empno: int, Ename: string ...
3 more fields]
```

```
dfFromData.createOrReplaceTempView("EmpDetails")
dfFromData.withColumn("Sal",dfFromData("Sal")+100).show(false)
```

```
+-----+------+--------+----+------+
|Empno|Ename |Job     |Sal |Deptno|
+-----+------+--------+----+------+
|7369 |SMITH |CLERK   |900 |20    |
|7499 |ALLEN |SALESMAN|1700|30    |
|7521 |WARD  |SALESMAN|1350|30    |
|7566 |JONES |MANAGER |3075|20    |
|7654 |MARTIN|SALESMAN|1500|30    |
|7698 |BLAKE |MANAGER |2950|30    |
+-----+------+--------+----+------+
```

```
spark.sql("Select Empno,Ename,Job,Sal+100 as Sal,Deptno from
EmpDetails").show()
```

```
+-----+------+--------+----+------+
|Empno| Ename|     Job| Sal|Deptno|
+-----+------+--------+----+------+
| 7369| SMITH|   CLERK| 900|    20|
| 7499| ALLEN|SALESMAN|1700|    30|
| 7521|  WARD|SALESMAN|1350|    30|
| 7566| JONES| MANAGER|3075|    20|
| 7654|MARTIN|SALESMAN|1500|    30|
| 7698| BLAKE| MANAGER|2950|    30|
+-----+------+--------+----+------+
```

```
dfFromData.withColumn("Bonus",dfFromData("Sal")*0.20).show(false)
```

```
+-----+------+--------+----+------+-----+
|Empno|Ename |Job     |Sal |Deptno|Bonus|
+-----+------+--------+----+------+-----+
|7369 |SMITH |CLERK   |800 |20    |160.0|
|7499 |ALLEN |SALESMAN|1600|30    |320.0|
|7521 |WARD  |SALESMAN|1250|30    |250.0|
|7566 |JONES |MANAGER |2975|20    |595.0|
|7654 |MARTIN|SALESMAN|1400|30    |280.0|
|7698 |BLAKE |MANAGER |2850|30    |570.0|
+-----+------+--------+----+------+-----+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Empno", IntegerType,true),
                 StructField("Ename",StringType,true),
                 StructField("Job", StringType,true),
                 StructField("Sal", IntegerType,true),
                 StructField("Dept", StringType,true),
             ))
val data =Seq(
    Row(7369,"SMITH","CLERK",800,"Operations"),
    Row(7499,"ALLEN","SALESMAN",1600,"Marketing"),
    Row(7521,"WARD","SALESMAN",1250, "Marketing"),
    Row(7566,"JONES","MANAGER",2975, "Operations"),
    Row(7654,"MARTIN","SALESMAN",1400, "Marketing"),
    Row(7698,"BLAKE","MANAGER",2850, "Marketing"),

)
var dfFromData =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData.show()
```

```
+-----+------+--------+----+----------+
|Empno| Ename|     Job| Sal|      Dept|
+-----+------+--------+----+----------+
| 7369| SMITH|   CLERK| 800|Operations|
| 7499| ALLEN|SALESMAN|1600| Marketing|
| 7521|  WARD|SALESMAN|1250| Marketing|
| 7566| JONES| MANAGER|2975|Operations|
| 7654|MARTIN|SALESMAN|1400| Marketing|
| 7698| BLAKE| MANAGER|2850| Marketing|
+-----+------+--------+----+----------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Empn
o,IntegerType,true), StructField(Ename,StringType,true), StructField(Job,Str
ingType,true), StructField(Sal,IntegerType,true), StructField(Dept,StringTyp
e,true))
data: Seq[org.apache.spark.sql.Row] = List([7369,SMITH,CLERK,800,Operation
s], [7499,ALLEN,SALESMAN,1600,Marketing], [7521,WARD,SALESMAN,1250,Marketin
g], [7566,JONES,MANAGER,2975,Operations], [7654,MARTIN,SALESMAN,1400,Marketi
ng], [7698,BLAKE,MANAGER,2850,Marketing])
dfFromData: org.apache.spark.sql.DataFrame = [Empno: int, Ename: string ...
3 more fields]
```

```
dfFromData.createOrReplaceTempView("EmpDept")
```

```
spark.sql("Select Dept,sum(Sal) from EmpDept group by Dept").show()
```

```
+----------+--------+
|      Dept|sum(Sal)|
+----------+--------+
|Operations|    3775|
| Marketing|    7100|
+----------+--------+
```

```
spark.sql("Select Dept,Job,sum(Sal) from EmpDept group by Dept,Job").show()
```

```
+----------+--------+--------+
|      Dept|     Job|sum(Sal)|
+----------+--------+--------+
|Operations|   CLERK|     800|
| Marketing|SALESMAN|    4250|
|Operations| MANAGER|    2975|
| Marketing| MANAGER|    2850|
+----------+--------+--------+
```

```
spark.sql("Select Dept,sum(Sal) from EmpDept group by Dept
having(sum(sal)>=5000)").show()
```

```
+---------+--------+
|     Dept|sum(Sal)|
+---------+--------+
|Marketing|    7100|
+---------+--------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Empno", IntegerType,true),
                 StructField("Ename",StringType,true),
                 StructField("Job", StringType,true),
                 StructField("Sal", IntegerType,true),
                 StructField("DeptId", IntegerType,true),
             ))
val data =Seq(
    Row(7369,"SMITH","CLERK",800,20),
    Row(7499,"ALLEN","SALESMAN",1600,30),
    Row(7521,"WARD","SALESMAN",1250,30),
    Row(7566,"JONES","MANAGER",2975,20),
    Row(7654,"MARTIN","SALESMAN",1400,30),
    Row(7698,"BLAKE","MANAGER",2850,30)
)
var dfFromData1 =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData1.show()
```

```
+-----+------+--------+----+------+
|Empno| Ename|     Job| Sal|DeptId|
+-----+------+--------+----+------+
| 7369| SMITH|   CLERK| 800|    20|
| 7499| ALLEN|SALESMAN|1600|    30|
| 7521|  WARD|SALESMAN|1250|    30|
| 7566| JONES| MANAGER|2975|    20|
| 7654|MARTIN|SALESMAN|1400|    30|
| 7698| BLAKE| MANAGER|2850|    30|
+-----+------+--------+----+------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Empn
o,IntegerType,true), StructField(Ename,StringType,true), StructField(Job,Str
ingType,true), StructField(Sal,IntegerType,true), StructField(DeptId,Integer
Type,true))
data: Seq[org.apache.spark.sql.Row] = List([7369,SMITH,CLERK,800,20], [7499,
ALLEN,SALESMAN,1600,30], [7521,WARD,SALESMAN,1250,30], [7566,JONES,MANAGER,2
975,20], [7654,MARTIN,SALESMAN,1400,30], [7698,BLAKE,MANAGER,2850,30])
dfFromData1: org.apache.spark.sql.DataFrame = [Empno: int, Ename: string ...
3 more fields]
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema= StructType( Array(
                 StructField("DeptId", IntegerType,true),
                 StructField("Dept",StringType,true),
                 StructField("Loc", StringType,true),
             ))
val data =Seq(
    Row(10,"ACCOUNTING","NEWYORK"),
    Row(20,"RESEARCH","DALLAS"),
    Row(30,"SALES","CHICAGO"),
    Row(40,"OPERATIONS","BOSTON"))
var dfFromData2 =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData2.show()
```

```
+------+----------+-------+
|DeptId|      Dept|    Loc|
+------+----------+-------+
|    10|ACCOUNTING|NEWYORK|
|    20|  RESEARCH| DALLAS|
|    30|     SALES|CHICAGO|
|    40|OPERATIONS| BOSTON|
+------+----------+-------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(DeptI
d,IntegerType,true), StructField(Dept,StringType,true), StructField(Loc,Stri
ngType,true))
data: Seq[org.apache.spark.sql.Row] = List([10,ACCOUNTING,NEWYORK], [20,RESE
ARCH,DALLAS], [30,SALES,CHICAGO], [40,OPERATIONS,BOSTON])
dfFromData2: org.apache.spark.sql.DataFrame = [DeptId: int, Dept: string ...
1 more field]
```

```
dfFromData1.join(dfFromData2,dfFromData1("DeptId") ===
dfFromData2("DeptId"),"inner").show(false)
```

```
+-----+------+--------+----+------+------+--------+-------+
|Empno|Ename |Job     |Sal |DeptId|DeptId|Dept    |Loc    |
+-----+------+--------+----+------+------+--------+-------+
|7369 |SMITH |CLERK   |800 |20    |20    |RESEARCH|DALLAS |
|7566 |JONES |MANAGER |2975|20    |20    |RESEARCH|DALLAS |
|7499 |ALLEN |SALESMAN|1600|30    |30    |SALES   |CHICAGO|
|7521 |WARD  |SALESMAN|1250|30    |30    |SALES   |CHICAGO|
|7654 |MARTIN|SALESMAN|1400|30    |30    |SALES   |CHICAGO|
```

```
|7698 |BLAKE |MANAGER |2850|30    |30    |SALES   |CHICAGO|
+-----+------+--------+----+------+------+--------+-------+
```

```
dfFromData1.join(dfFromData2,dfFromData1("DeptId") ===
dfFromData2("DeptId"),"leftouter").show(false)
```

```
+-----+------+--------+----+------+------+--------+-------+
|Empno|Ename |Job     |Sal |DeptId|DeptId|Dept    |Loc    |
+-----+------+--------+----+------+------+--------+-------+
|7369 |SMITH |CLERK   |800 |20    |20    |RESEARCH|DALLAS |
|7566 |JONES |MANAGER |2975|20    |20    |RESEARCH|DALLAS |
|7499 |ALLEN |SALESMAN|1600|30    |30    |SALES   |CHICAGO|
|7521 |WARD  |SALESMAN|1250|30    |30    |SALES   |CHICAGO|
|7654 |MARTIN|SALESMAN|1400|30    |30    |SALES   |CHICAGO|
|7698 |BLAKE |MANAGER |2850|30    |30    |SALES   |CHICAGO|
+-----+------+--------+----+------+------+--------+-------+
```

```
dfFromData1.join(dfFromData2,dfFromData1("DeptId") ===
dfFromData2("DeptId"),"rightouter").show(false)
```

```
+-----+------+--------+----+------+------+----------+-------+
|Empno|Ename |Job     |Sal |DeptId|DeptId|Dept      |Loc    |
+-----+------+--------+----+------+------+----------+-------+
|null |null  |null    |null|null  |10    |ACCOUNTING|NEWYORK|
|7369 |SMITH |CLERK   |800 |20    |20    |RESEARCH  |DALLAS |
|7566 |JONES |MANAGER |2975|20    |20    |RESEARCH  |DALLAS |
|7499 |ALLEN |SALESMAN|1600|30    |30    |SALES     |CHICAGO|
|7521 |WARD  |SALESMAN|1250|30    |30    |SALES     |CHICAGO|
|7654 |MARTIN|SALESMAN|1400|30    |30    |SALES     |CHICAGO|
|7698 |BLAKE |MANAGER |2850|30    |30    |SALES     |CHICAGO|
|null |null  |null    |null|null  |40    |OPERATIONS|BOSTON |
+-----+------+--------+----+------+------+----------+-------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Empno", IntegerType,true),
                 StructField("Ename",StringType,true),
                 StructField("Job", StringType,true),
                 StructField("Sal", IntegerType,true),
                 StructField("DeptId", IntegerType,true),
             ))
val data =Seq(
    Row(7369,"SMITH","CLERK",800,20),
    Row(7499,"ALLEN","SALESMAN",1600,30),
    Row(7521,"WARD","SALESMAN",1250,30),
    Row(7566,"JONES","MANAGER",2975,20),
    Row(7654,"MARTIN","SALESMAN",1400,30),
    Row(7698,"BLAKE","MANAGER",2850,30)
)
var dfFromData1 =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData1.show()
```

```
+-----+------+--------+----+------+
|Empno| Ename|     Job| Sal|DeptId|
+-----+------+--------+----+------+
| 7369| SMITH|   CLERK| 800|    20|
| 7499| ALLEN|SALESMAN|1600|    30|
| 7521|  WARD|SALESMAN|1250|    30|
| 7566| JONES| MANAGER|2975|    20|
| 7654|MARTIN|SALESMAN|1400|    30|
| 7698| BLAKE| MANAGER|2850|    30|
+-----+------+--------+----+------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Empn
o,IntegerType,true), StructField(Ename,StringType,true), StructField(Job,Str
ingType,true), StructField(Sal,IntegerType,true), StructField(DeptId,Integer
Type,true))
data: Seq[org.apache.spark.sql.Row] = List([7369,SMITH,CLERK,800,20], [7499,
ALLEN,SALESMAN,1600,30], [7521,WARD,SALESMAN,1250,30], [7566,JONES,MANAGER,2
975,20], [7654,MARTIN,SALESMAN,1400,30], [7698,BLAKE,MANAGER,2850,30])
dfFromData1: org.apache.spark.sql.DataFrame = [Empno: int, Ename: string ...
3 more fields]
```

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window

//row_number
val windowSpec  = Window.partitionBy("DeptId").orderBy("Sal")
dfFromData1.withColumn("rank",row_number.over(windowSpec)).show()
```

```
+-----+------+--------+----+------+----+
|Empno| Ename|     Job| Sal|DeptId|rank|
+-----+------+--------+----+------+----+
| 7369| SMITH|   CLERK| 800|    20|   1|
| 7566| JONES| MANAGER|2975|    20|   2|
| 7521|  WARD|SALESMAN|1250|    30|   1|
| 7654|MARTIN|SALESMAN|1400|    30|   2|
| 7499| ALLEN|SALESMAN|1600|    30|   3|
| 7698| BLAKE| MANAGER|2850|    30|   4|
+-----+------+--------+----+------+----+
```

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
windowSpec: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.s
ql.expressions.WindowSpec@e7464a
```

```
+-----+------+--------+----+------+----+
|Empno| Ename|     Job| Sal|DeptId|rank|
+-----+------+--------+----+------+----+
| 7369| SMITH|   CLERK| 800|    20|   1|
| 7566| JONES| MANAGER|2975|    20|   2|
| 7521|  WARD|SALESMAN|1250|    30|   1|
| 7654|MARTIN|SALESMAN|1400|    30|   2|
| 7499| ALLEN|SALESMAN|1600|    30|   3|
| 7698| BLAKE| MANAGER|2850|    30|   4|
+-----+------+--------+----+------+----+
```

```
import org.apache.spark.sql.functions._
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType,
ArrayType,IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
val schema = StructType( Array(
                 StructField("Empno", IntegerType,true),
                 StructField("Ename",StringType,true),
                 StructField("Job", StringType,true),
                 StructField("Sal", IntegerType,true),
                 StructField("DeptId", IntegerType,true),
             ))
val data =Seq(
    Row(7369,"SMITH","CLERK",800,20),
    Row(7499,"ALLEN","SALESMAN",1600,30),
    Row(7521,"WARD","SALESMAN",1250,30),
    Row(7566,"JONES","MANAGER",2975,20),
    Row(7654,"MARTIN","SALESMAN",1400,30),
    Row(7698,"BLAKE","MANAGER",2850,30)
)
var dfFromData1 =
spark.createDataFrame(spark.sparkContext.parallelize(data),schema)
dfFromData1.show()
```

```
+-----+------+--------+----+------+
|Empno| Ename|     Job| Sal|DeptId|
+-----+------+--------+----+------+
| 7369| SMITH|   CLERK| 800|    20|
| 7499| ALLEN|SALESMAN|1600|    30|
| 7521|  WARD|SALESMAN|1250|    30|
| 7566| JONES| MANAGER|2975|    20|
| 7654|MARTIN|SALESMAN|1400|    30|
| 7698| BLAKE| MANAGER|2850|    30|
+-----+------+--------+----+------+
```

```
import org.apache.spark.sql.types.{StringType, StructField, StructType, Arra
yType, IntegerType}
import org.apache.spark.sql.Row
import scala.collection.JavaConversions._
schema: org.apache.spark.sql.types.StructType = StructType(StructField(Empn
o,IntegerType,true), StructField(Ename,StringType,true), StructField(Job,Str
ingType,true), StructField(Sal,IntegerType,true), StructField(DeptId,Integer
Type,true))
data: Seq[org.apache.spark.sql.Row] = List([7369,SMITH,CLERK,800,20], [7499,
ALLEN,SALESMAN,1600,30], [7521,WARD,SALESMAN,1250,30], [7566,JONES,MANAGER,2
975,20], [7654,MARTIN,SALESMAN,1400,30], [7698,BLAKE,MANAGER,2850,30])
dfFromData1: org.apache.spark.sql.DataFrame = [Empno: int, Ename: string ...
3 more fields]
```

```
dfFromData1.createOrReplaceTempView("EmpDetails")
spark.sql("SELECT Ename,Sal,LAG(sal) OVER (PARTITION BY DeptId Order BY Sal)
As PreviousSal FROM EmpDetails").show(false)
```

```
+------+----+-----------+
|Ename |Sal |PreviousSal|
+------+----+-----------+
|SMITH |800 |null       |
|JONES |2975|800        |
|WARD  |1250|null       |
|MARTIN|1400|1250       |
|ALLEN |1600|1400       |
|BLAKE |2850|1600       |
+------+----+-----------+
```

```
+------+----+-------+
|Ename |Sal |NextSal|
+------+----+-------+
|SMITH |800 |2975   |
|JONES |2975|null   |
|WARD  |1250|1400   |
|MARTIN|1400|1600   |
|ALLEN |1600|2850   |
|BLAKE |2850|null   |
+------+----+-------+
```