**Submit Sample Spark Job.**

./bin/spark-submit --class org.apache.spark.examples.SparkPi --master local D:\spark\examples\jars\spark-examples\_2.11-2.3.1.jar

**RDD Transformations:**

Read it Through File:

val rdd1 = sc.textFile("file:///d:/sample.txt")

val rdd2 = rdd1.map(x => x+"aaa")

rdd2.collect

val rdd3 = rdd1.flatMap (line => line.split(" "))

val rdd3 = rdd1.flatMap (line => line.split(" ")).map(x => (x,1)).reduceByKey((x,y) => x+ y)

val rdd3 = rdd1.flatMap (line => line.split(" ")).map(x => (x,1)).reduceByKey(\_ + \_)

rdd3.take(2)

rdd3.filter(x=>x.\_2>1)

**map():**

val input = sc.parallelize(List(7, 3, 1))

val result = input.map(x => x +5)

result.collect()

**flatmap():**

val lines = sc.parallelize(List("hello world", “how is spark"))

val words = lines.flatMap(line => line.split(" "))

words.first()

**filter():**

val result = input.filter(x => x > 4)

result.collect()

**Union():**

val rdd1 = sc.parallelize(List(1,2,3))

val rdd2 = sc.parallelize(List(3,4,5))

val result = rdd1.union(rdd2)

result.collect()

**Intersection():**

val result = rdd1.intersection(rdd2)

result.collect()

**distinct():**

val result = rdd1.distinct()

**Join():**

val data = sc.parallelize(Array(("John",1000),("James",2000),("Rob",3000)))

val data2 =sc.parallelize(Array(("John",3000),("Bobby",5000),("James",7000),("Rob",3000)))

val result = data.join(data2)

result.collect()

**sortbyKey():**

val data = sc.parallelize(Array(("Computer Science",90), ("Electronics",75), ("Electrical",80), ("Mechanical",60), ("Chemical",85)))

val sorted = data.sortByKey()

sorted.collect()

**groupByKey() &reduceByKey():**

val rdd3 = sc.textFile("file:///d:/sample.txt").flatMap (line => line.split(" ")).map(x => (x,1)).reduceByKey(\_ + \_)

**RDD Actions:**

**Count():**

val input = sc.parallelize(List(7, 3, 1))

input.count()

**top():**

input.top(2)

**reduce()**

val rdd1 = sc.parallelize(List(10,20,40,30))

val sum = rdd1.reduce(\_+\_)

println(sum)

**Dataframes:**

Read a file:

scala> val df = spark.read.json("file:///d:/employee.json")

scala> df.show()

scala> df.printSchema()

scala> df.select("salary").show()

df.select($"salary"+100).show()

**Dataframe filter and groupby:**

scala> df.filter(df("salary") > 100000).show()

val df1=df.filter($"salary" > 120000).show

scala> df.groupBy("deptid").count().show()

**Dataframe Join:**

val df1 = spark.read.json("file:///d:/employee.json")

val df2 = spark.read.json("file:///d:/dept.json")

df1.join(df2,Seq("deptid")).show

**Create Temp view on dataframe:**

scala> df.createOrReplaceTempView("employee")

scala> val sqlDF = spark.sql("SELECT \* FROM people").show()

scala> val sqlDF = spark.sql("SELECT \* FROM employee where deptid=101")

**Case Class in Dataset:**

case class Employee(deptid:Long,name:String,salary:Long)

val ds=spark.read.json("file:///d:/employee.json").as[Employee]

ds.show

ds.filter(\_.deptid>102).show

ds.filter(col("deptid")>102).show