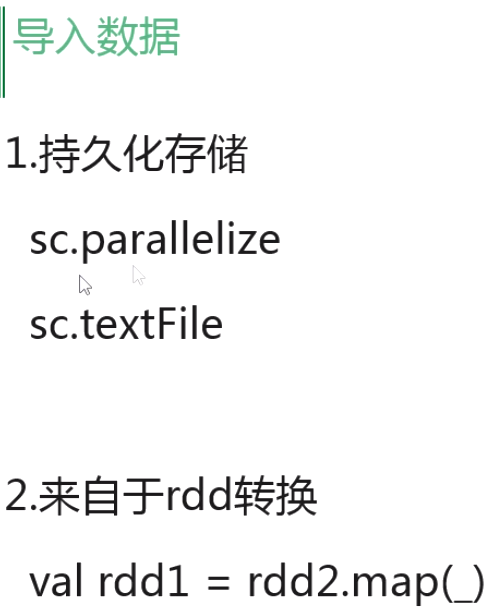
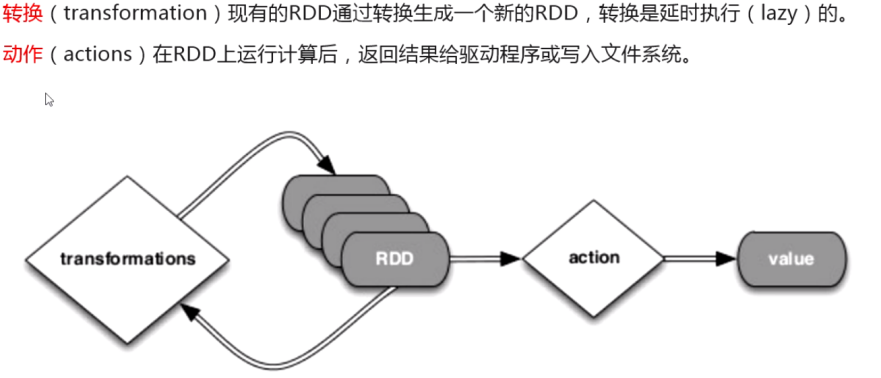
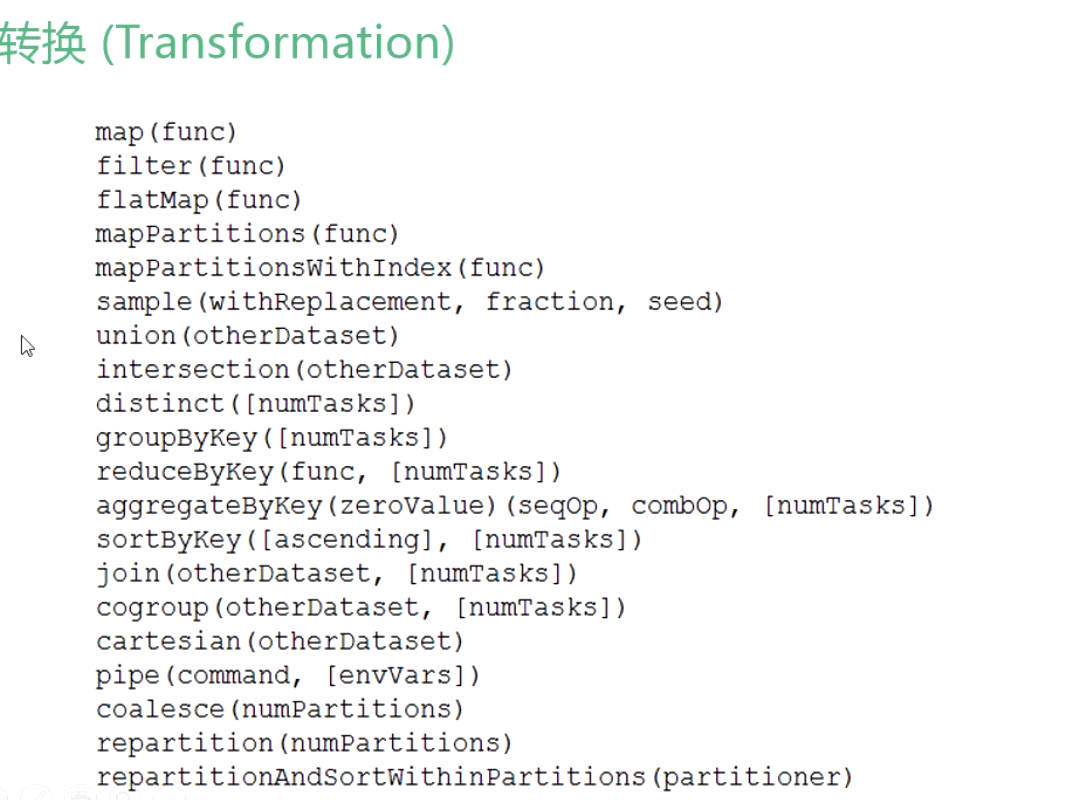
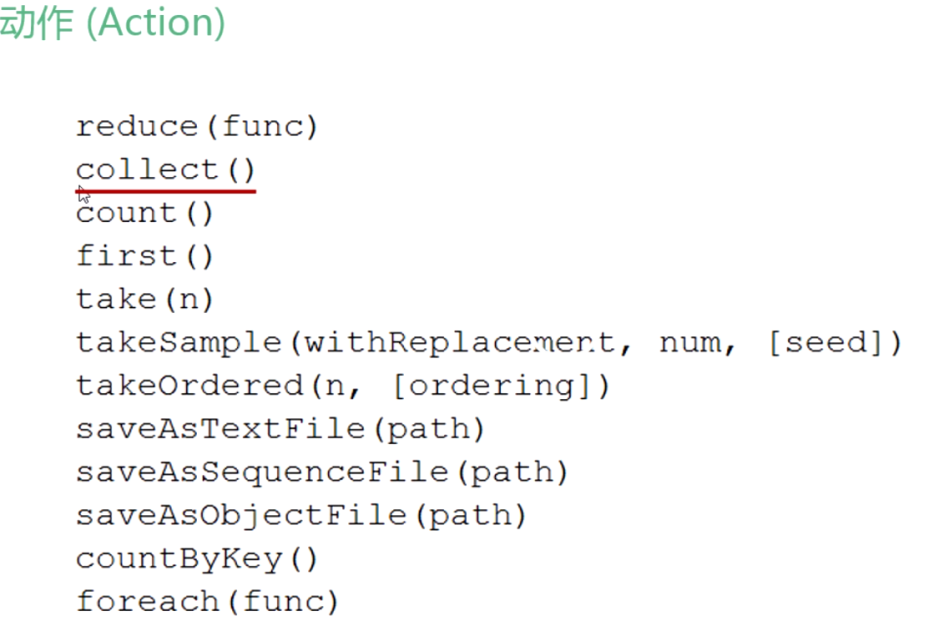
导入数据









count判断数据的行数

first 输入数据的前一行

collect 查看变量

hadoop fs -mkdir /sparksql

[kjk@hadoop15 resources]$ hadoop fs -put ./\* /sparksql/

spark-shell --master spark://192.168.242.10:7077

告诉集群主节点是谁

***第一种***

val sqlContext = new org.apache.spark.sql.SQLContext(sc)

import sqlContext.implicits.\_

case class People(name:String, age:Int)



val dfpeople = sc.textFile("/sparksql/people.txt").map(\_.split(",")).map(p => People(p(0), p(1).trim.toInt)).toDF()

dfpeople.registerTempTable("people")



查看表格

sqlContext.sql("show tables").map(t => "tableName is:" + t).collect().foreach(println)

查看表的第一列数据

sqlContext.sql("select name from people").map(t => "people anme is:" + t(0)).collect().foreach(println)

第二种

import org.apache.spark.sql.\_

import org.apache.spark.sql.types.\_

创建rdd

val rddpeople =sc.textFile("/sparksql/people.txt")

将rdd改成rowrdd

val rowRDD = rddpeople.map(\_.split(",")).map(p => Row(p(0), p(1).trim))

创建俩列

val schemaString = "name age"

读取json文件

val jsonpeople = sqlContext.jsonFile("/sparksql/people.json")

jsonpeople.registerTempTable("jsonpeople")

sqlContext.sql("show tables").map(t => "tableName is:" + t).collect().foreach(println)

sqlContext.sql("select name from jsonpeople").map(t => "tabcollectleName is:" + t).().foreach(println)

读取parquet

<https://blog.csdn.net/xianrenqiu1234/article/details/90484833>

val parquetpeople =sqlContext.parquetFile("/sparksql/users.parquet")

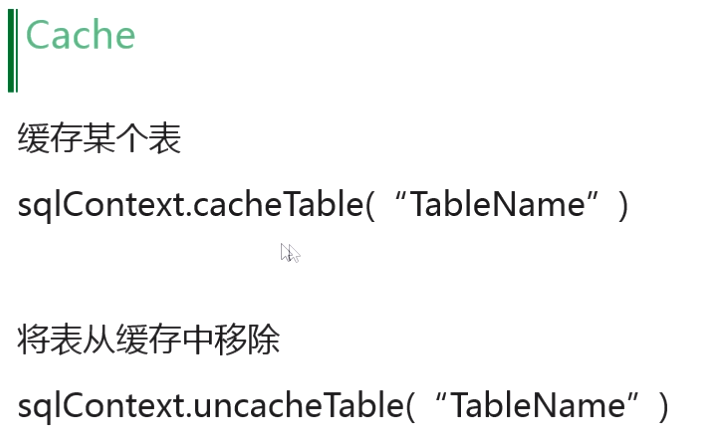
parquetpeople.registerTempTable("parquetpeople")

sqlContext.sql("show tables").map(t => "tableName is:" + t).collect().foreach(println)

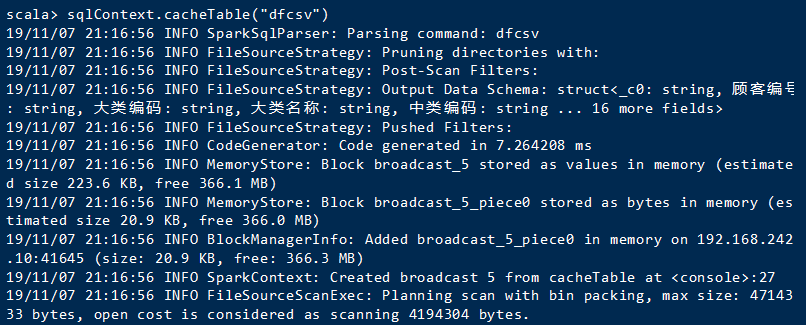
csv读取

val df=spark.read.format("csv").option("header", "true").option("mode", "DROPMALFORMED")

.csv("/sparksql/task1\_1.csv")



sqlContext.cacheTable("df")



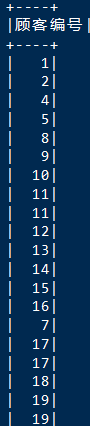
val sqlContext = new org.apache.spark.sql.SQLContext(sc)

将表缓存

Sql查询

df.select("顾客编号").show()

查询某一列数据



df.select("大类名称").where($"大类名称"==="日配").show()

所有大类名称为日配的提取出来

df.select("\*").where($"大类名称"==="日配").show()

单价大于5的提取出来

df.select("\*").where($"商品单价">5).show()

orderBy/sort排序

df.select("\*").sort($"商品单价".desc).show()

对商品单价进行降序排序

df.groupBy("商品单价").sum($"商品单价").show()



SparkSq|基础语法一上

通过方法来使用

(1)查询

tslelte't",'),showl);

(2)带条件的查询

= "bb"),show()

dieelel'""m,","age" ).Where($"age">20).show

(3)排序查询

orderBy/sort($"咧名") 升序排列

orderey/ort($列名".desc)降序排列

orderBy/sort($列1" , $"列2".desc)按两列排序

dieeliei'i',"'ame".rderey($" name" desc).show

df,selte(s"id", s"'ame"5or(s"name" desc),show

tabslect(""5"ame,"sts""ame"desc).show

(4)分组查询

groupyY“"列名", ..max列名)求最大值

groupBy("列名", ..min列名)求最小值

groupBy("列名", ..avg/列名)平均值

groupByV"列名", ..um列名)求和

groupBy("列名", .cout个数

groupBy("列名", ..g可以将多个方法进行聚合

scal>val rdd = s.aeDi(t(1," ")" 10,",",",0,1," ","45);)

scal>val df= rdotd',"a",",dr" "cre";

sarfoupuyad"contQ.sho.w)

sala>froupBya'dr"aggaxl'scor'", min'S"score"), cut\*(""),show

(5)连接查询

scala>val dtstscasos:,aobu",",afau1",frt'trtid',"deptnaie’"’

scala>val empsclralel(s(:,o,ang,2.,,,",,0,a)1l)F(","i","name"’"

scala>dept.oinemp,s"deptid" === $"d).show

第三个参数:①left②right③inner full

scala>deptjoin(emp,$"deptid" === s"ird","'f'" ).show

左向外联接的结果集包括LEFT OUTER子句中指定的左表的所有行，而不仅仅是联接列所匹配的行。如果左表的某行在右表中没

表列均为

scala>deptjoin(emp,$"deptid" === $"did'"ight").show

(6)执行运算

df.select($" num" \* 100).show

(7)使用列表

deslelextrerner,"rs([)".show"

(8)使用结构体

{"name"."陈晨,"res:"ct,":"西安t","tete."南二环甲字1号"]

m"ama"她娜""res"("ty:"西安" ,"treet:"南二环甲字2号”}

val df= sqyhcet.anio/f,sle///woro/osersriro’

sselel"'me,","arsttshows

(9)其他

df.count/获取记录总数

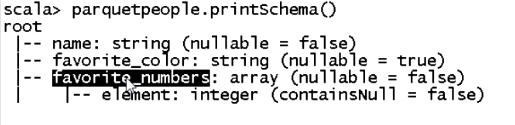
val row= drfrt/获取第一记录

val tae-df.take(2) /获取前n条记录

val value = rw.tetrin1//取该行指定列的值

df.ollect /获取当前df对象中的所有数据为一个Array 其实就是调用了df对象对应的底层的dd的lesct方法nevineniu1234.

用saveAsParquetFile(“文件路径”)对文件进行格式转换



对parquet文件进行查看

toJSON.savaASTextFile(“文件路径.json”)

保存df文件：

sc.textFile("/sparksql/people.txt").flatMap(line => line.split(" ")).map(word =>(word,1)) .reduceByKey(\_ + \_).repartition(1).saveAsTextFile("sparksql/df.txt")