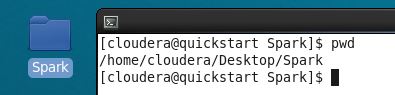
1. **Spark所讀取的資料位置**

$ cd Desktop/Spark

****

1. **進入 Spark**

$ spark-shell



**2-1. Using Spark SQL by spark-shell**

#-- SQL Context

Entry point for all SQL functionality

Wraps/extends existing SparkContext

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

Scale > import sqlContext.\_

#-- 定義資料類別boxLevel, 設定輸入資料型態

Scale > :paste

case class boxLevel(pkno: String, budget: Long, currency4wg: String, weekendgross: Int, date: Int, screen: Int, level: Int)

val boxlevel = sc

.textFile("./data/boxLevelComplete.txt")

.map(\_.split(","))

.map(p => boxLevel(p(0), p(1).trim.toLong, p(2), p(3).trim.toInt, p(4).trim.toInt, p(5).trim.toInt, p(6).trim.toInt))



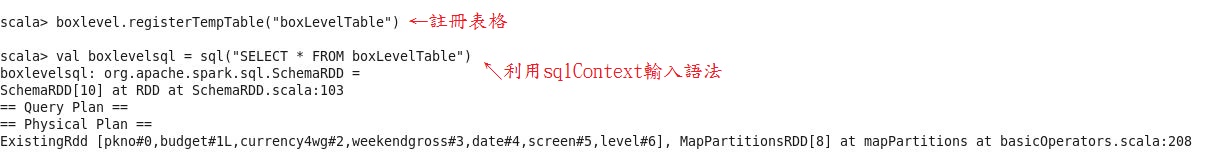
#-- boxlevel 為 boxLevel類別內的變數名稱, 並給定表名稱為boxLevelTable

Scale > boxlevel.registerTempTable("boxLevelTable")

#-- SQL statements can be run by using the sql methods provided by

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

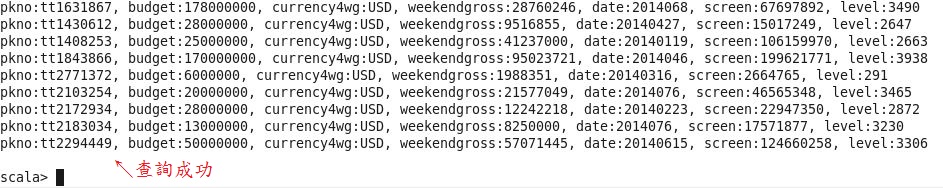
Scale > val boxlevelsql = sql("SELECT \* FROM boxLevelTable")



#-- The results of SQL queries are SchemaRDDs and support all the normal RDD operations.

The columns of a row in the result can be accessed by ordinal.

Scala > boxlevelsql.map(t => "pkno:" + t(0)+ ", budget:" + t(1) + ", currency4wg:" + t(2) + ", weekendgross:" + t(3) + ", date:" + t(4) + ", screen:" + t(5) + ", level:" + t(6)).collect().foreach(println)



**2-2. Hive table**

To be continues…

#-- ver 20150113