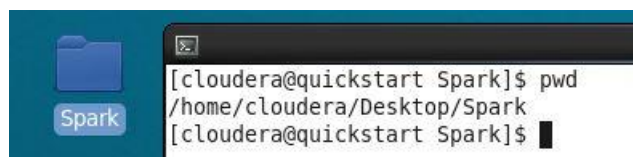


1-0. Spark 所讀取的資料位置

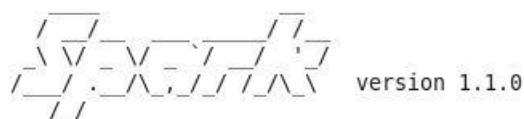
```
$ cd Desktop/Spark
```



2-0. 進入 Spark

```
$ spark-shell
```

Welcome to



Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)

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))
```

```
scala> :paste
// Entering paste mode (ctrl-D to finish)
```

```
// Define the schema using a case class.
case class boxLevel(pkno: String, budget: Long, currency4wg: String, weekendgross: Int, date: Int, screen: Int, level: Int) ←資料型態要正確
// Create an RDD of Person objects and register it as a table
// data source by cloudera
val boxlevel = sc
.textFile("./data/boxLevelComplete.txt") ←資料存放位置
.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))
```

```
// Exiting paste mode, now interpreting. ←資料型態轉換
```

```
2015-01-13 00:24:26,809 INFO [main] storage.MemoryStore (Logging.scala:logInfo(59)) - ensureFreeSpace(177621) called with curMem=0, maxMem=278302556
2015-01-13 00:24:26,811 INFO [main] storage.MemoryStore (Logging.scala:logInfo(59)) - Block broadcast_0 stored as values in memory (estimated size 173.5 KB, free 265.2 MB)
defined class boxLevel ←類別定義成功
boxlevel: org.apache.spark.rdd.RDD[boxLevel] = MappedRDD[3] at map at <console>:16
```

```

#-- 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")

scala> boxlevel.registerTempTable("boxLevelTable") ←註冊表格
scala> val boxlevelsql = sql("SELECT * FROM boxLevelTable")
boxlevelsql: org.apache.spark.sql.SchemaRDD =
SchemaRDD[10] at RDD at SchemaRDD.scala:103
== Query Plan ==
== Physical Plan ==
ExistingRDD [pkno#0,budget#1L,currency4wg#2,weekendgross#3,date#4,screen#5,level#6], MapPartitionsRDD[8] at mapPartitions at basicOperators.scala:208

#-- 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)

pkno:tt1631867, budget:178000000, currency4wg:USD, weekendgross:28760246, date:2014068, screen:67697892, level:3490
pkno:tt1430612, budget:280000000, currency4wg:USD, weekendgross:9516855, date:20140427, screen:15017249, level:2647
pkno:tt1408253, budget:250000000, currency4wg:USD, weekendgross:41237000, date:20140119, screen:106159970, level:2663
pkno:tt1843866, budget:1700000000, currency4wg:USD, weekendgross:95023721, date:2014046, screen:199621771, level:3938
pkno:tt2771372, budget:60000000, currency4wg:USD, weekendgross:1988351, date:20140316, screen:2664765, level:291
pkno:tt2103254, budget:200000000, currency4wg:USD, weekendgross:21577049, date:2014076, screen:46565348, level:3465
pkno:tt2172934, budget:280000000, currency4wg:USD, weekendgross:12242218, date:20140223, screen:22947350, level:2872
pkno:tt2183034, budget:130000000, currency4wg:USD, weekendgross:82500000, date:2014076, screen:17571877, level:3230
pkno:tt2294449, budget:500000000, currency4wg:USD, weekendgross:57071445, date:20140615, screen:124660258, level:3306

scala> █ ↗查詢成功

```

```

#-- ver 20150113

```