## File 1:

**val** sc = spark.sparkContext  
**val** sqlContext = **new** org.apache.spark.sql.SQLContext(sc)  
**val** data = sc.textFile(**"/user/horton/task1/"**)  
**val** dfdata = data.map(line => line.split(**","**)).map(row => *flights*(row(1), row(2), row(3), row(4), row(5).toInt, row(6).toInt))  
**import** sqlContext.implicits.\_  
dfdata.toDF().createOrReplaceTempView(**"dfdatatbl"**)  
**val** finaldata = spark.sql(**""** +  
 **"select year,day,month,deparloc,"** +  
 **"deprDelay,ArrDelay "** +  
 **"from dfdatatbl where deprDelay>=10 order by ArrDelay asc"**)  
  
finaldata.*rdd*.map(line => line.mkString(**","**)).repartition(1).saveAsTextFile(**"/user/horton/solutions/task1"**)

## File 2:

**import** spark.implicits.\_  
  
**val** data =sqlContext.read.json(**"/user/horton/task2"**)  
data.toDF().registerTempTable(**"dfdatatbl"**)  
**val** finaldata= sqlContext.sql(**"select state,avg(balance) from dfdatatbl where balance>2000 group by state"**)  
finaldata.*rdd*.map(line=>line.mkString(**"\t"**)).repartition(1).saveAsTextFile(**"/user/horton/solutions/task2"**)

## File 3:

**val** nErrors=sc.accumulator(0.0)  
**val** logs = sc.textFile(**"/user/horton/task6/output.log"**)  
logs.filter(line =>line.contains(**"ERROR"**)).foreach(x=>nErrors+=1)  
*println*(nErrors.value)

## File 4: (hive table)

**import** spark.implicits.\_  
**val** data = sqlContext.sql(**"select snow\_depth,date from blizzards where cast(snow\_depth as int) >=10"**)  
data.*rdd*.map(line=>line.mkString(**"\t"**)).repartition(1).saveAsTextFile(**"/user/horton/task4"**)

## File 5: (hive table)

**val** data = sqlContext.sql(**"select avg(euro),avg(dollar) from exchangedata where substring(date,0,4)='2012'"**)  
data.*rdd*.map(line =>line.mkString(**","**)).saveAsTextFile(**"/user/horton/task5"**)

## File 6:

**case class** weather(stationid: String, elevation: String, windspeed: String, temp: String)  
**case class** sation(stationid: String, name: String)

**val** weaterdata = sc.textFile(**"/user/horton/task6/weather.txt"**)  
**val** dfweather2 = weaterdata.map(line => line.split(**","**))

**val** dfweather3= dfweather2.map(row => *weather*( row(0), row(1), row(2), row(3)))  
  
**val** stationdata = sc.textFile(**"/user/horton/task6/station.txt"**)  
**val** dfstation = stationdata.map(line => line.split(**","**)).map(  
 row => *sation*(row(0), row(3)))  
  
**import** spark.implicits.\_  
dfweather3.toDF().registerTempTable(**"dfweathertbl"**)

dfweather4 = dfweather3.sql(select \* from **dfweathertbl where temp** != **"\*\*\*\*"**)

dfweather4.toDF().registerTempTable(**"dfweathertbl4"**)

dfweather5 = dfweather3.sql(select \* from **dfweathertbl where cast(temp as int) >=32"**)

dfweather5.toDF().registerTempTable(**"dfweathertbl5"**)

dfstation.toDF().registerTempTable(**"dfstationtbl"**)  
  
**val** finaldata = sqlContext.sql(**"select a.stationid,a.elevation,a.windspeed,b.name "** +  
 **"from dfweathertbl5 a join dfstationtbl b on (a.stationid=b.stationid) "** +  
 **"order by a.temp asc"**)  
  
finaldata.*rdd*.map(line => line.mkString(**"\t"**)).repartition(1).saveAsTextFile(**"/user/horton/solutions/task6"**)

## File7:

*spark-submit --jars SparkYarn.jar --class SparkYarn --master yarn-client SparkYarn.jar*