sqoop import --connect jdbc:mysql://nn01.itversity.com:3306/retail\_db --username retail\_dba --password itversity --table orders --target-dir /user/rss161030/cloudera/problem1/orders --as-avrodatafile --compress --compression-codec org.apache.hadoop.io.compress.SnappyCodec

sqoop import --connect jdbc:mysql://nn01.itversity.com:3306/retail\_db --username retail\_dba --password itversity --table order\_items --target-dir /user/rss161030/cloudera/problem1/order-items --as-avrodatafile --compress --compression-codec org.apache.hadoop.io.compress.SnappyCodec

val ordersDF = sqlCon.read.avro("/user/rss161030/cloudera/problem1/orders")

val orderItemsDF = sqlCon.read.avro("/user/rss161030/cloudera/problem1/order-items")

ordersDF.registerTempTable("orders")

orderItemsDF.registerTempTable("orderItems")

val sqlResult = sqlCon.sql("select to\_date(from\_unixtime(cast(order\_date/1000 as bigint))) order\_formatted\_date,order\_status,COUNT(distinct(order\_id)) total\_orders, CAST(SUM(order\_item\_subtotal) as DECIMAL(10,2)) total\_amount from orders join orderItems on order\_id = order\_item\_order\_id group by to\_date(from\_unixtime(cast(order\_date/1000 as bigint))),order\_status order by order\_formatted\_date desc,order\_status,total\_amount desc,total\_orders")

sqlCon.setConf("spark.sql.parquet.compression.codec","gzip")

sqlResult.write.parquet("/user/rss161030/cloudera/problem1/result4b-gzip")

sqlCon.setConf("spark.sql.parquet.compression.codec","snappy")

sqlResult.write.parquet("/user/rss161030/cloudera/problem1/result4b-snappy")

sqlResult.map(rec => rec.toString()).saveAsTextFile("/user/rss161030/cloudera/problem1/result4b-csv")

sqlResult.map(rec => rec(0) + "," + rec(1) + "," + rec(2) + "," + rec(3)).saveAsTextFile("/user/rss161030/cloudera/problem1\_2/result4b-csv")

sqoop export --connect jdbc:mysql://nn01.itversity.com/retail\_export --username retail\_dba --password itversity --table zrss161030result --export-dir /user/rss161030/cloudera/problem1\_2/result4b-csv