sqoop import --connect jdbc:mysql://localhost/retail\_db --username retail\_dba --password cloudera --table orders --target-dir /user/cloudera/problem5/text --fields-terminated-by \\t --lines-terminated-by \\n

sqoop import --connect jdbc:mysql://localhost/retail\_db --username retail\_dba --password cloudera --table orders --target-dir /user/cloudera/problem5/avro --as-avrodatafile

sqoop import --connect jdbc:mysql://localhost/retail\_db --username retail\_dba --password cloudera --table orders --target-dir /user/cloudera/problem5/parquet --as-parquetfile

val dataDF = sqlContext.read.avro("/user/cloudera/problem5/avro")

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

dataDF.write.parquet("/user/cloudera/problem5/parquet-snappy-compress")

dataDF.map(rec => (rec(0)+"\t"+rec(1)+"\t"+rec(2)+"\t"+rec(3)).saveAsTextFile("/user/cloudera/problem5/text-gzip-compress",classOf[org.apache.hadoop.io.compress.gzipCodec])

sqlContext.setConf("spark.sql.text.compression.codec","gzip")

dataDF.saveAsTextFile("/user/cloudera/problem5/text-gzip-compress")

val dfRDD = dataDF.map(rec => rec(0) + "," + rec(1) + "," + rec(2) + "," + rec(3))

dataDF.map(rec => (rec(0)+"\t"+rec(1)+"\t"+rec(2)+"\t"+rec(3))).saveAsTextFile("/user/cloudera/problem5/text-gzip-compress",classOf[org.apache.hadoop.io.compress.GzipCodec])

dfRDD.map(rec => (rec(0).toInt,rec(0)+"\t"+rec(1)+"\t"+rec(2)+"\t"+rec(3))).saveAsSequenceFile("/user/cloudera/problem5/sequence")

(OR)

dataDF.map(rec => (rec(0).toString,rec(0)+"\t"+rec(1)+"\t"+rec(2)+"\t"+rec(3))).saveAsSequenceFile("/user/cloudera/problem5/sequence1")

val dataDF1 = sqlContext.read.parquet("/user/cloudera/problem5/parquet-snappy-compress")

sqlContext.setConf("spark.sql.parquet.compression.codec","uncompressed")

dataDF1.write.parquet("/user/cloudera/problem5/parquet-no-compress")

sqlContext.setConf("spark.sql.avro.compression.codec","snappy")

dataDF1.write.avro("/user/cloudera/problem5/avro-snappy")

val avroDataFile = sqlContext.read.avro("/user/cloudera/problem5/avro-snappy")

avroDataFile.toJSON.saveAsTextFile("/user/cloudera/problem5/json-no-compress")

avroDataFile.toJSON.saveAsTextFile("/user/cloudera/problem5/json-gzip",classOf[org.apache.hadoop.io.compress.GzipCodec])

val jsonDataFrame = sqlContext.read.json("/user/cloudera/problem5/json-gzip")

jsonDataFrame.map(rec => (rec(0) +","+ rec(1) +","+ rec(2) +","+ rec(3))).saveAsTextFile("/user/cloudera/problem5/csv-gzip",classOf[org.apache.hadoop.io.compress.GzipCodec])

val seqData = sc.sequenceFile("/user/cloudera/problem5/sequence", classOf[Int], classOf[String])

(OR)

var seqData1 = sc.sequenceFile("/user/cloudera/problem5/sequence1", classOf[org.apache.hadoop.io.Text], classOf[org.apache.hadoop.io.Text])