Hadoop File Formats & Data Ingestion Hands On

Exercise 1

Use Kite SDK to demonstrate copying of various file formats to Hadoop

Step 1) Download the MovieLens Dataset

```
curl http://files.grouplens.org/datasets/movielens/ml-latest-small.zip -o movies.zip
unzip movies.zip
cd ml-latest-small/
```

Step 2) Load the Dataset into Hadoop in Avro format

```
-- infer the schema
kite-dataset csv-schema ratings.csv --record-name ratings -o ratings.avsc
cat ratings.avsc
-- create the schema
kite-dataset create ratings --schema ratings.avsc
-- load the data
kite-dataset csv-import ratings.csv --delimiter ',' ratings
```

Step 3) Load the Dataset into Hadoop in Parquet format

```
kite-dataset csv-schema ratings.csv --record-name ratingsp -o ratingsp.avsc
cat ratingsp.avsc
kite-dataset create ratingsp --schema ratingsp.avsc --format parquet
kite-dataset csv-import ratings.csv --delimiter ',' ratingsp
```

Step 4) Run a sample query to compare the elapsed time between Avro & Parquet

```
hive
select avg(rating) from ratings;
select avg(rating) from ratingsp;
exit;
```

Exercise 2

Use Sqoop to copy an Oracle table to Hadoop

Step 1) Download the Oracle jdbc driver

```
sudo su -
cd /var/lib/sqoop
curl -L https://pkothuri.web.cern.ch/pkothuri/ojdbc6.jar -o ojdbc.jar
exit;
```

Step 2) Run sqoop to copy a table from Oracle

```
sqoop import \
--connect jdbc:oracle:thin:@devdb11-s.cern.ch:10121/devdb11_s.cern.ch \
--username hadoop_tutorial \
-P \
--num-mappers 1 \
--target-dir visitcount_rfidlog \
--table VISITCOUNT.RFIDLOG
```

Check the size and number of files

```
hdfs dfs -ls visitcount_rfidlog/
```

Exercise 3

Use Sqoop to copy an Oracle table to Hadoop, multiple mappers

```
sqoop import \
--connect jdbc:oracle:thin:@devdb11-s.cern.ch:10121/devdb11_s.cern.ch \
--username hadoop_tutorial \
-P \
--num-mappers 2 \
--split-by alarm_id \
--target-dir lemontest_alarms \
--table LEMONTEST.ALARMS \
--as-parquetfile
```

Check the size and number of files

```
hdfs dfs -ls lemontest_alarms/
```

Exercise 4

Use Sqoop to make incremental copy of an Oracle table to Hadoop

Step 1) Initial load of the table; create sqoop job and execute

```
-- Create the Job

sqoop job \
--create alarms \
-- \
import \
--connect jdbc:oracle:thin:@devdb11-s.cern.ch:10121/devdb11_s.cern.ch \
--username hadoop_tutorial \
-P \
--num-mappers 1 \
--target-dir lemontest_alarms_i \
--table LEMONTEST.ALARMS \
--incremental append \
--check-column alarm_id \
--last-value 0 \

-- Execute the Job
sqoop job --exec alarms
```

Step 2) Incremental load of the table

```
sqoop import \
--connect jdbc:oracle:thin:@devdb11-s.cern.ch:10121/devdb11_s.cern.ch \
--username hadoop_tutorial \
-P \
--num-mappers 1 \
--table LEMONTEST.ALARMS \
--target-dir lemontest_alarms_i \
--incremental append \
--check-column alarm_id \
--last-value 47354 \
```

Step 3) Check the Hadoop files

```
hdfs dfs -ls lemontest_alarms_i/
```