Intro to Big Data - Lab 3 (Data Ingestion with Apache Sqoop)

What we will cover in this lab:

- **Connecting to a database via Sqoop**
- **Importing data from the DB tables into HDFS**
- **Basic data manipulation, basic queries**

Useful links:

https://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html

Preamble:

- Basic information:

  Sqoop is a tool designed to transfer data between Hadoop and relational databases or mainframes. You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle or a mainframe into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.

- Sqoop 'import'
  Sqoop 'import' is the most important Sqoop command. It uses map reduce framework to connect to database and copy data in parallel into HDFS. The Sqoop 'import' tool imports an individual table from an RDBMS to HDFS. Each row from a table is represented as a separate record in HDFS. Records can be stored as text files (one record per line), or in binary representation as Avro or SequenceFiles.

Your tasks:

1. Connect to a publicly available database

- Connect to a publicly available database server using the following jdbc URL: "jdbc:mysql://quickstart.cloudera:3306" and list all the available databases:

```
$ sqoop list-databases \
  --connect "jdbc:mysql://quickstart.cloudera:3306" \
  --username=retail_dba \
  --password=cloudera
```

- List all tables from the **retail_db** database:

```
$ sqoop list-tables \
  --connect"jdbc:mysql://quickstart.cloudera:3306/retail_db" \
```

```
  --username=retail_dba \
  --password=cloudera
```

- Import all tables from the **retail_db** database and store them into the target directory „lab3”:

```
$ sqoop import-all-tables \
  --connect”jdbc:mysql://quickstart.cloudera:3306/retail_db” \
  --username=retail_dba \
  --password=cloudera \
  --warehouse-dir /lab3
```

Remark: /lab3 is the base directory under which the imported tables will be created (/lab 3 is created on the root)

- Verify the imported tables in HDFS:

```
$ hadoop fs –ls /lab3


$ hadoop fs –ls –R /lab3
```

- Verify the imported data of table **orders** in HDFS:

```
$ hadoop fs -cat /lab3/orders/part-m-*
```

- Import all tables from the **retail_db** database *as avro datafiles* and store them into the base directory ”/lab3”:

```
$ hadoop fs –rm -R /lab3
```

```
$ sqoop import-all-tables \
  -m 12
  --connect”jdbc:mysql://quickstart.cloudera:3306/retail_db” \
  --username=retail_dba \
  --password=cloudera \
  --as-avrodatafile
  --warehouse-dir /lab3
```

Remark: -m or --num-mapers signifies parallel threads

- Verify the imported data as avro files in HDFS:

```
$ hadoop fs -ls -R /lab3
```

- Verify that .avsc files are created in the directory from where the above command is run:

```
$ ls -ltr *.avsc
```

Remark: Avro files generate metadata for the structure of data.

- The following command is used to import a *subset* of the table **orders**. The subset query is to retrieve the orders with status CLOSED:

```
$ sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username=retail_dba \
--password=cloudera \
--table orders \
--where "order_status ='CLOSED'" \
--target-dir /wherequery
```

Remarks:

  - The column name order_status is visible as metadata in the .avsc file associated to the table orders
  - The value 'CLOSED' is visible in the imported (text) table data.

Assignments:

- Import only the column **order_status** from the table **orders**, as text files (default). Use the argument --columns (see the sqoop documentation).
- Using pyspark with (one of) the resulted data files, print how many times each order_status appears in that file.

2. Let us take the example of two tables named **Movie** and **Has_played_in** which are in a database called **userdb** in a MySQL database server. The structure of the two tables is as follows:

movie (digital_visa, title, year, type, nb_spec, budget) - describing for each movie its digital visa, its title, its year, its type – comedy, etc., the number of spectators having watched the movie ever since it was released and the production budget.

has_played_in (first_name_actor, last_name_actor, digital_visa) - describing the main actors for each of the movies.

- Use sqoop import to import a table into HDFS
  Sqoop tool 'import' is used to import table data from the table to the Hadoop file system as a text file or a binary file.

  The following command is used to import the movie table from MySQL database server to HDFS.
  Remark: if you are using a different database server, please provide the appropriate *connect string* that describes how to connect to the database. The *connect string* is similar to a URL, and is communicated to Sqoop with the --connect argument. This describes the server and database to connect to; it may also specify the port.

```
$ sqoop import \
--connect "jdbc:mysql://localhost/userdb" \
--username root \
--table movie --m
--target-dir /lab3
```

- Verify the imported data in HDFS:

```
$ hadoop fs -ls /lab3
```

```
$ hadoop fs -cat /lab3/movie/part-m-*
```

  It shows you the movie table data and fields are separated with comma (,)
- Importing into target directory: import has_played_in table data into '/lab3' directory:

```
$ sqoop import \
--connect "jdbc:mysql://localhost/userdb" \
--username root \
--table has_played_in \
```

```
--m 1 \
--target-dir /lab3
```

- Verify the imported data in /lab3 directory from has_played_in table:

```
hadoop fs -cat /lab3/has_played_in/part-m-*
```

It will show you the has_played_in table data with comma (,) separated fields

- Import subset of table data:
  We can import a subset of a table using the 'where' clause in Sqoop import tool. It executes the corresponding SQL query in the respective database server and stores the result in a target directory in HDFS.
  The syntax for where clause is as follows:

```
--where <condition>
```

- The following command is used to import a subset of movie table data. The subset query is to retrieve the movies released in 2015:

```
$ sqoop import \
--connect "jdbc:mysql://localhost/userdb" \
--username root \
--table movie \
--m 1 \
--where "year = 2015" \
--target-dir /wherequery
```

Assignment:

- Execute all the above commands listed in exercise 1 and 2.
- In your database, create the two tables in exercise 2 and populate them. You may use any kind of database, but have to provide an appropriate URL for the server in order to connect to the DB.
- Import the movies released in the 90s.
- Import the movies having a production budget less than 1.5 million dollars and a number of spectators greater than 10 million.
- Import the animation movies released in 2012 and in 2013.
- Import the table **has_played_in** into HDFS.
- Using pyspark with the files resulted from the previous import (table **has_played_in**), count how many times Johnny Depp has played in a movie.