**ASSIGNMENT**

**1. We have NYSE(New York Stock Exchange data) data into our oracle database,updated daily(using some schedular.Autosys,Cron Job)**

****

**How to insert csv data into database?**

**2.Upload the data from database to HDFS (sqoop import) (Using Java way)**

sqoop import \

--options-file /home/hduser/myfile.txt \

--table nyse\_daily\_prices -m 1 \

--columns "STOCK\_EXCHANGE,STOCK\_SYMBOL,STOCK\_DATE,STOCK\_PRICE\_OPEN, \

STOCK\_PRICE\_HIGH,STOCK\_PRICE\_LOW,STOCK\_PRICE\_CLOSE,STOCK\_VOLUME,\

STOCK\_PRICE\_ADJ\_CLOSE" \

--hive-database default

--hive-table nyse\_daily\_prices\_sqoop

--hive-import

The above command will import the data from oracle to hive directly.sqoop will take care of creating the table with the same schema as we have in the oracle.

Loading the data into HDFS using Sqoop Command.

sqoop import --options-file /home/hduser/myfile.txt --table nyse\_daily\_prices -m 1 --columns "STOCK\_EXCHANGE,STOCK\_SYMBOL,STOCK\_DATE,STOCK\_PRICE\_OPEN,STOCK\_PRICE\_HIGH,STOCK\_PRICE\_LOW,STOCK\_PRICE\_CLOSE,STOCK\_VOLUME,STOCK\_PRICE\_ADJ\_CLOSE" --target-dir /user/hduser/nyse\_daily\_prices/ --delete-target-dir

**3.Write Map-Reduce program to select only those records with stock\_volume>5000**

**Select \* from nyse\_daily\_price\_sqoop where stock\_volume > 5000;**

**4.Load that output to external hive table(sqoop java client/hive java client)**

create external table nyse\_daily\_prices\_sqoop\_external(stock\_exchange string,stock\_symbol string,stock\_date string,stock\_price\_open double,stock\_price\_high double,stock\_price\_low double,stock\_price\_close double,stock\_volume double,stock\_price\_adj\_close double) row format delimited fields terminated by ',' location '/user/hduser/nyse\_daily\_prices';

5.Findout the top 10 nyse\_data with highest volume and write them to hdfs file.(hive client)

insert overwrite directory '/user/hduser/nyse/' select \* from nyse\_daily\_prices\_sqoop order by stock\_volume desc limit 3;

**6.Write sqoop command to create another table which will hold top 10 details.**

sqoop eval --options-file /home/hduser/myfile.txt --query "CREATE table nyse\_daily\_prices\_top\_stk\_vol(stock\_exchange varchar(20),stock\_symbol varchar(20),stock\_date varchar(20),stock\_price\_open number(5,15),stock\_price\_high number(5,15),stock\_price\_low number(5,15),stock\_price\_close number(5,15),stock\_volume number,stock\_price\_adj\_close number(5,15))

**7.Export the data from hdfs file(step 5) into Database table(step-6).**

**Create an table in hive**

create table nyse\_daily\_prices\_sqoop\_analz(stock\_exchange string,stock\_symbol string,stock\_date string,stock\_price\_open double,stock\_price\_high double,stock\_price\_low double,stock\_price\_close double,stock\_volume double,stock\_price\_adj\_close double) row format delimited fields terminated by ',';

load the hdfs file to table;

load data inpath '/user/hduser/nyse/\*' overwrite into table nyse\_daily\_prices\_sqoop\_analz;

create table in mysql to hold the resultant data.

create table nyse\_daily\_prices\_sqoop\_analz(stock\_exchange varchar(20),stock\_symbol varchar(20),stock\_date varchar(20),stock\_price\_open double,stock\_price\_high double,stock\_price\_low double,stock\_price\_close double,stock\_volume double,stock\_price\_adj\_close double);

Export the data from hive to mysql database.

sqoop --connect jdbc:mysql://localhost:3306/test --username root --password root --table nyse\_daily\_prices\_sqoop\_analz -m 1 --columns "stock\_exchange,stock\_symbol,stock\_date,stock\_price\_open,stock\_price\_high,stock\_price\_low,stock\_price\_close,stock\_volume,stock\_price\_adj\_close" --export-dir /user/hive/warehouse/nyse\_daily\_prices\_sqoop\_analz --input-fields-terminated-by '\001'

**Using JavaClient for sqoop**

**https://sqoop.apache.org/docs/1.99.6/ClientAPI.html**

**http://devslogics.blogspot.in/2013/09/sqoop-java-client.html**

**https://community.cloudera.com/t5/Data-Ingestion-Integration/How-to-run-a-sqoop-command-inside-the-java-code/td-p/2985**

**https://sqoop.apache.org/docs/1.99.6/ClientAPI.html**