**sqoop**

Bulk data transfer tool / Data ingestion tool

Mysql --------------------> HDFS

Mysql ---------------------> Hive table

--Login to mysql

sudo mysql -u root -p

Passwork for ak : P@55word

pwd(mysql) :root

---------------------------------------

create database sqoopdb;

use sqoopdb;

create table policy (policy\_id int, policy\_holder varchar(10),sum\_insured int, terms\_in\_years int);

insert into policy values(3001,’Anand’,50000,30);

insert into policy values(100,'Ak',50000,10);

insert into policy values(3002,'Gautam',50000,10);

--open new terminal and execute the following sqoop commands

---------------------------------------------------------------------------------------------------

---loading data from mysql table to hdfs file

--/user/ak/ is the sqoop staging area

--After execution , find the folder under /user/ak/policy

**sqoop-import --connect jdbc:mysql://localhost:3306/sqoopdb?useSSL=false -username ak -password ak -table policy -m 1**

------------------------------------------------------------

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword -table policy --where "policy\_holder='bbb'" --append --split-by policy\_id -m 2;

-m is to set number of mappers

this is only for non-primary key tables .

-----------------------------------------------------------------------------------------

--loading data to user defined directory

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword -table policy --where "policy\_holder='anand'" --target-dir /yourname/mysqlload/ --append -m 1;

----to list all the tables from database

sqoop-list-tables --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword

sqoop-list-databases --connect jdbc:mysql://localhost/ -username hiveuser -password hivepassword

--transfer only particular columns from the table to hdfs

sqoop-import --connect jdbc:mysql://localhost/sqoopdb -username hiveuser -password hivepassword -table policy -columns policy\_holder --append -m 1

-----------------------------------------------------------------------------------------

--to store data in other file formats

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword -table policy --as-sequencefile --append -m 1;

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword -table policy --as-textfile --append -m 1;

-----------------------------------------------------------------------------

---Exporting data from hdfs file to mysql table . First, create the mysql table based on schema on the hdfs file and then perform export

--in mysql

--create table patmysql (pid int ,pname varchar(10),drug varchar(10),gender varchar(10),amt int);

sqoop export -connect jdbc:mysql://localhost/sqoopdb?useSSL=false --table patmysql --export-dir /lti/patient.txt --username hiveuser --password hiveuser -m 1 --input-fields-terminated-by ','

----------------------------------------------------------------------------------------

//  **Hive**

---Download and copy hive -common.jar file to sqoop-1.4.7 / lib

------------<https://repo1.maven.org/maven2/org/apache/hive/hive-common/2.3.7/hive-common-2.3.7.jar>

Loading data from Mysql to Hive table

---command creates and load the data

---duplicate policy table in mysql or remove /user/ak/policy folder in hdfs

--In -hive-table database.tablename

create table policy1 as select \* from policy;

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --table emp -create-hive-table -hive-table ak.empsqoop -hive-import --fields-terminated-by ',' -m 1;

---to load to existing hive table

sqoop import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --table policy --hive-import -hive-table lti.policy\_hivetable --fields-terminated-by ',' -m 1;

---------------------------------------------------------------

---exporting data from hive hdfs location to mysql table

sqoop export --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --table policy --export-dir /user/hive/warehouse/lti.db/patient/patient.txt --input-fields-terminated-by ',';

===========to load all table -->bulk load

sqoop-import-all-tables --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --hive-import -m 1;

------incremental load → to load only new records after first execution

Last modified ⇒ to apply this source table must have date column →

Last values ⇒ 1----------------10

1 -100

→

101 --

---------------------------------------increment using last value-----------------

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --table policy --incremental append --check-column policy\_id -last-value 3066 -m 1

------------------------Increment using last modified ------------------------------------

----Using data source date column

====Apply on your project dataset==========================

sqoop-import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username hiveuser -password hivepassword --hive-import –incremental append –last-modified orderdate –last value 20-05-1990 -m 1

Emp.txt --1---100 -----------1-1000

Full import -- delta load

1001 --2000 --- > 1-1000 → 1---2000

For Automatic last-value create sqoop job without last-value parameter

sqoop job --create sqjobinc -- import --connect jdbc:mysql://localhost/sqoopdb?useSSL=false -username geoinsys -P --table patmysql --incremental append --check-column pid -target-dir /user/hive/warehouse/anand.db/patsqoop -m 1

Sqoop job --exec sqjobinc