## Transfer data between Mysql and HDFS (Import and Export) using Sqoop.

Importing Data from Mysql to HDFS

Create table in Hive:

create database acad;

create table company

(

id int,

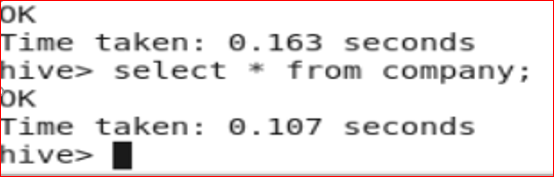
name string,

location string

);

This table will be stored in hive deault location : /user/hive/warehouse/acad.db/company

Check data in hive table(Data does not exist)



Sqoop Command to import data:

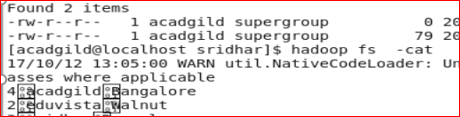
sqoop import --connect jdbc:mysql://localhost/b1 \

--username 'root' -P --table 'company' \

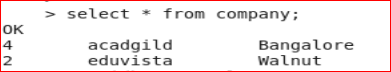
--hive-import --hive-table 'company' -m 1 \

--warehouse-dir /user/hive/warehouse/acad.db;

Check data in warehouse directory of hdfs:



Check data in hive table company using select \* from company (data got loaded)



Exporting Data from HDFS to Mysql

Create table in Mysql using create statement

create table b1.company

(

id int,

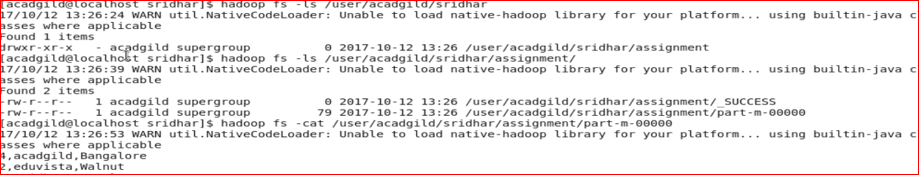
name varchar(20),

location varchar(20)

);

Check HDFS data to be exported:

hadoop fs -cat /user/acadgild/charan /assignment/part-m-00000



Sqoop Export:

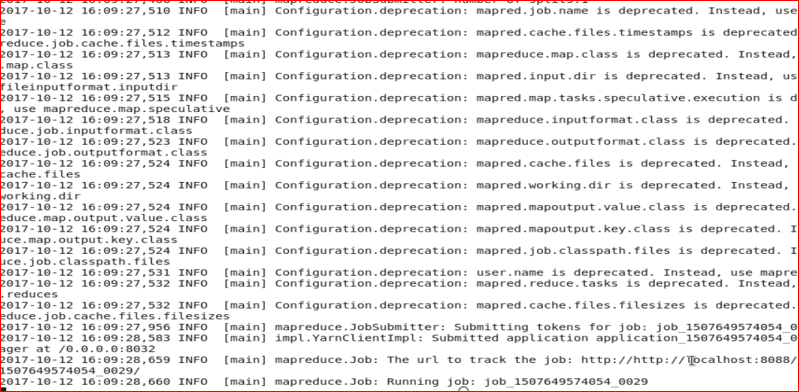
sqoop export --connect jdbc:mysql://localhost/b1 \

--username 'root' -P --table 'company' --export-dir '/user/acadgild/ charan /assignment' \

--input-fields-terminated-by ',' -m 1 --columns id,name,location \

Explanation(--connect to specify connection string, --username to specify the username of the db, -P to specify the password at run time, --export-dir to specify the directory from where the data will be exported,--input-fields-terminated-by ',' to specify fields are terminated by ‘,’, -m 1 to specify the no of mapper tasks, --columns to specify the columns to be exported

)



## ● Transfer data between Mysql and Hive (Import and Export only selected columns) using Sqoop.

Importing data from Mysql to Hive Table

First create company Table in mysql using create statement:

create table company

(

id int,

name varchar(20),

location varchar(20)

);

Load data into company table:

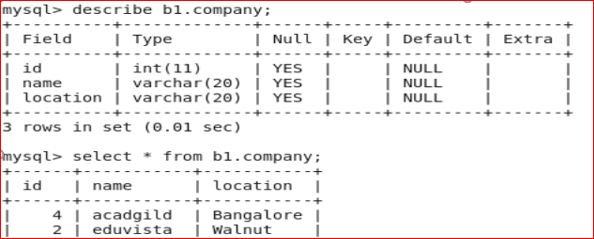
insert into company values(4, 'acadgild','Bangalore');

insert into company values(2, 'eduvista','Walnut');

Check data in mysql table company using

select \* from b1.comapny ;

B1 is the database used



Next Create table in Hive :

Using create statement

create database acad; //database acad

use acad;

create table company

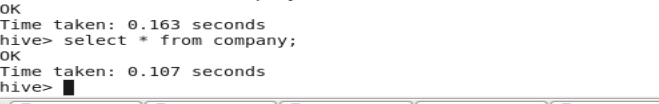
(

id int,

name string,

location string

Check data do not exist on hive using select \* from company



Run the Sqoop Import :

sqoop import --connect jdbc:mysql://localhost/b1 --username 'root' -P

--table 'company' --hive-import --hive-table 'company' -m 1

--warehouse-dir /user/hive/warehouse/acad.db;

--columns "id,name"

Explanation( --connect to specify the connection string,

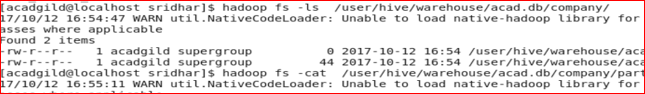
--username to specify the username of the db, -P to provide password during run time so to maintain security, use –hive-import and provide –hive-table <<table\_name>>

--warehouse-dir is the parent directory in hdfs where the data will be stored

--columns “id,name” to import only selected columns)

Check data on warehouse directory

hadoop fs -cat /user/hivewarehouse/acad.db/comapny/part-m-00000



Check data on hive after import using query select \* from company

Only data for id and name got imported



Exporting data from Hive to Mysql:

Create table in Hive using create statement:

create database acad;

use acad;

create table company

(

id int,

name string,

location string

Check data in mysql using create statement:

create table company

(

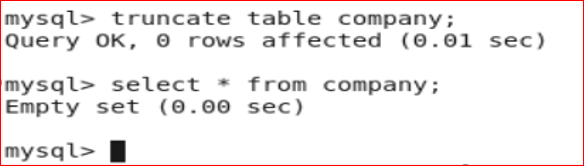
id int,

name varchar(20),

location varchar(20)

);

Check data in mysql (Data do not exist)



Run sqoop export:

sqoop export --connect jdbc:mysql://localhost/b1 \

--username 'root' \

-P \

--table company \

--export-dir /user/hive/warehouse/acad.db/company \

--input-fields-terminated-by '\001' \

--input-lines-terminated-by '\n' \

--num-mappers 2 \

--batch \

--input-null-string nvl \

--input-null-non-string -1

--columns "id,location"

Explanation( --connect to specify the connection string,

--username to specify the username of the db, -P to provide password during run time so to maintain security, --columns “id,name” to export only selected columns ,

--num-mappers to specify mapper task, --batch to be executed in batch mode,

--export-dir to specify the directory from where data will be exported, -–table to specify the mysql table name where data will be stored, --input-lines-terminated-by to specify input fields are terminated by new line

--input-fields-terminated-by '\001' to specify fields terminated by '\001'

--input-null-non-string :The string to be interpreted as null for non-string columns

--input-null-string nvl : The string to be interpreted as null for string columns )