**Just Keep Sqoopin’ - 1**

**WHAT HAPPENS if:**

1. Target Directory ---> HDFS specified directory;

2. No target directory ---> root

3. Hive import ---> root(temp) -->apps/hive/warehouse

4. targetDirectory+Hive ---> HDFS specified directory(temp) -->apps/hive/warehouse

**Use Cases -**

* Sqoop import table from mySQL to HDFS
* Sqoop import table from mySQL into a target-dir in HDFS
* Sqoop import using --query into a target-dir into HDFS
* Hive import
* Hive import with create table option (imports to warehouse)
* Hive import using --query into target-dir
* **Sqoop import table from mySQL to HDFS**

Check to make sure your mySQL has the database movielens configured. We are going to import a table called genres from this database.

At the terminal run the following sqoop command -

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver --username root --password password  -m 1  --table genres; |

Once the above command runs, check in HDFS to locate where this table was imported to. It will create a folder name with the same name as the table.

This query will give you a table in mess in your HDFS (as a text as 1 row, without splitting).

* **Sqoop import table from mySQL into a target-dir in HDFS**

Run the same sqoop command as above but specify a target directory to import into

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver --username root --password password  --table genres  -- target-dir /user/maria\_dev/target  --fields-terminated-by ‘,’; |

**If you sqooping into target directory that doesn't exist, in first time this directory will be created with this query. Than if you want to sqoop file into this directry as a target dir and give the path for it in your query, you'll have an error as this directory already exist. So. you need to mantion the name of file for sqooping in this target dir.**

**Also can be a problem if you mention in query username root, password password - delete it.**

**Also use mapper.**

* **Sqoop import using --query into a target-dir into HDFS**

To select only a subset of rows/columns from the table genres, we use the --query

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver  --username root --password password  --query "SELECT id, name FROM genres WHERE \$CONDITIONS and name like 'A%' "  --split-by id --target-dir /user/maria\_dev/usingquery  --fields-terminated-by ','; |

Check the output at the above path in HDFS

**This way gives a structured table with field that you can make by like SQL queries.**

* **Hive import**

Run the following query to check what happens

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver --username root --password password  --table occupations  --hive-import; |

The table is again created in HDFS under /apps/hive/warehouse/occupations

* **Hive import with create table option (imports to warehouse)**

This sqoop command does a Hive import into ‘default’ database with the table name given. After execution, check Hive warehouse /apps/hive/warehouse

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver --username root --password password  --table occupations  -m 1  --hive-database default  --hive-table occu  --create-hive-table  --fields-terminated-by '\t'  --hive-import; |

* **Hive import using --query into target-dir**

This command creates a table with the given name in the default database and data is stored in HDFS at the specified path

|  |
| --- |
| sqoop import --connect jdbc:mysql://localhost/movielens  --driver com.mysql.jdbc.Driver --username root --password password  --query "SELECT \* FROM movies WHERE \$CONDITIONS limit 100"  --split-by id  --target-dir /user/maria\_dev/hiveimportmovies/  --hive-import  --hive-database default  --hive-table non\_movies  --create-hive-table  --fields-terminated-by '\t' |