```
grant all on *.* to 'root'@'localhost' with grant option;

sudo service mysqld start
mysql -u root -p

create database db3;

use hive;

create table company
(
id int,
name varchar(20),
location varchar(20)
);

insert into company values(1, 'ineuron','Bangalore');

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

insert into company values(3, 'eduvista','Bangalore');



flush privileges;

commit;

exit;

hadoop fs -ls /sqoopout
====================================================


 //check connection
 sqoop import --connect jdbc:mysql://localhost:3306 --username root --password hortonworks1

 //list databases
 sqoop list-databases --connect jdbc:mysql://localhost:3306 --username root --password
hortonworks1

// list-tables

  sqoop list-tables --connect jdbc:mysql://localhost:3306/hive --username root --password
```

hortonworks1

Single Mapper Import

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --username root --password
hortonworks1 --table TBLS --target-dir '/tmp/sqoophivetbls/' -m 1
```

============================

Multi-Mapper Import

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --username root --password
hortonworks1 --table TBLS --target-dir '/tmp/sqoophivetblsmultimap/' -m 5
```

===================================

Hive import

```
sudo su hive

sqoop import --connect jdbc:mysql://localhost:3306/hive --username root --password
hortonworks1 --table TBLS --target-dir '/tmp/sqoophivetblsHIVENEW/' -m 5 --hive-import  -m 5;
```

Go to hive and check table structure and content

==

Sqoop Export

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --username root --password
hortonworks1 --table company --target-dir '/tmp/company/' --split-by "id" -m 3

mysql -u root -p

use hive;
delete from company;
commit;

sqoop export --connect jdbc:mysql://localhost:3306/hive --username 'root' -password
hortonworks1 --table 'company' --export-dir '/tmp/company/' --input-fields-terminated-by ',' -m 3 -
```

```
-columns id,name,location
```

==

Using Stage Table

```
use db2;

describe company;

create table staging_company
(
id int(11),
name varchar(20),
location varchar(20)
);

sqoop export --connect jdbc:mysql://localhost/db2 --username 'root' -P --table 'company' --
export-dir '/sqoopout15' --input-fields-terminated-by ',' -m 1 --columns id,name,location --
staging-table staging_company
```

Sqoop requires that the structure of the staging table be the same as that of the target table. The number of columns and their types must be the same; otherwise, the export operation will fail.

When using a staging table, Sqoop will first export all data into this staging table instead of the main table that is present in the parameter --table. Sqoop opens a new transaction to move data from the staging table to the final destination, if and only if all parallel tasks successfully transfer data.

As Sqoop will export data into the staging table and then move it to the final table, there is a period of time where all your data is stored twice in the database.

As the data is first loaded somewhere else and then moved to the final table, using a staging table will always be slower than exporting directly to the final table.

Incremental Mode
----

use hive;

```
CREATE TABLE test_new (

emp_id int,

name varchar(100),

salary int,

update_time timestamp,

PRIMARY KEY (emp_id) )
```

```
INSERT INTO test_new values (1,"ABC",100,'2018-01-20 00:00:00');

INSERT INTO test_new values (2,"ABC",200,'2018-01-20 00:00:00');

INSERT INTO test_new values (3,"ABC",300,'2018-01-20 00:00:00');

INSERT INTO test_new values (4,"ABC",400,'2018-01-20 00:00:00');
```

Run Normal sqoop import ==========>

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --username 'root' --password hortonworks1 --table 'test_new' --target-dir '/tmp/test_new/'
```

Check the data that it has 4 records======================

Now insert a record to the table===========================

```
INSERT INTO test_new values (5,"ABC",500,'2018-01-20 00:00:00');
```

Run Incremental Import Append Mode ===============================

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --incremental append --check-column emp_id --last-value 4 --username root --password hortonworks1 --table test_new --target-dir '/tmp/test_new/'
```

CHECK THAT ONE MORE FILE GOT CREATED IN OUTPUT DIRECTORY AND record WITH emp_id **5** IS PRESENT

RUN BELOW SQL IN MYSQL to insert a new record=================================

```
INSERT INTO test_new values (6,'ABC',600,'2018-01-22 00:00:00');
```

```
update test_new set name='XYZ' , update_time='2018-01-22 00:00:00' where emp_id=4;
```

HERE WE HAVE INSERTED 1 NEW RECORD with emp_id 7 and UPDATED 1 RECORD for emp_id 4 . BOTH THE RECORDS NOW HAVE TIMESTAMP AS 2018-01-22 00:00:00

Run Incremental Import LastModified Mode To identify the Updates==============================

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --incremental lastmodified --check-column update_time --last-value 2018-01-21 --merge-key emp_id --username root --password hortonworks1 --table test_new --target-dir '/tmp/test_new/'
```

[--merge-key is required if we are doing the incremental import in already existing folder. This column will be used to merge the records for Update cases]

CHECK THE FOLDER DATA. IT should have now the previous data + updated data

Change file format of the target HDFS

Avro Data	**--as-avrodatafile**
Parquet File	**-as-parquetfile**

```
sqoop import --connect jdbc:mysql://localhost:3306/hive --username 'root' --password hortonworks1 --table 'test_new' --as-avrodatafile --target-dir '/tmp/test_new_avro/'
```