DICE ANALYTICS

Apache Sqoop Tutorial

Prepared by: BILAL MAQSOOD & NABEEL NAQEEBI

SQOOP

Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. Developed by Cloudera the name is actually a contraction of ‘SQL-to-Hadoop’ and it works on MapReduce framework. It works with the help of connectors. You can use different connectors to connect to RDBMS like MySQL, Oracle etc.

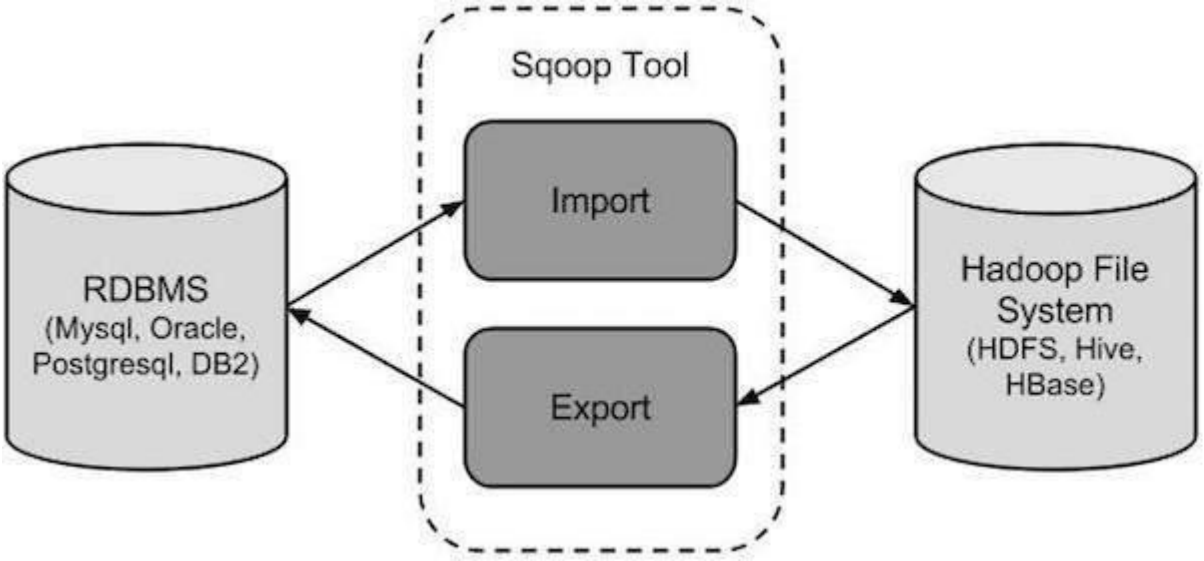
Sqoop has two major tools.

1. Sqoop Import:
2. Sqoop Export:

The documented is consisted of 4 section that are as follows:

1. Import data from MySQL to HDFS
2. Import data from MySQL to Hive
3. Export data from HDFS to MySQL
4. Export data from Hive to MySQL

The following image describes the workflow of Sqoop

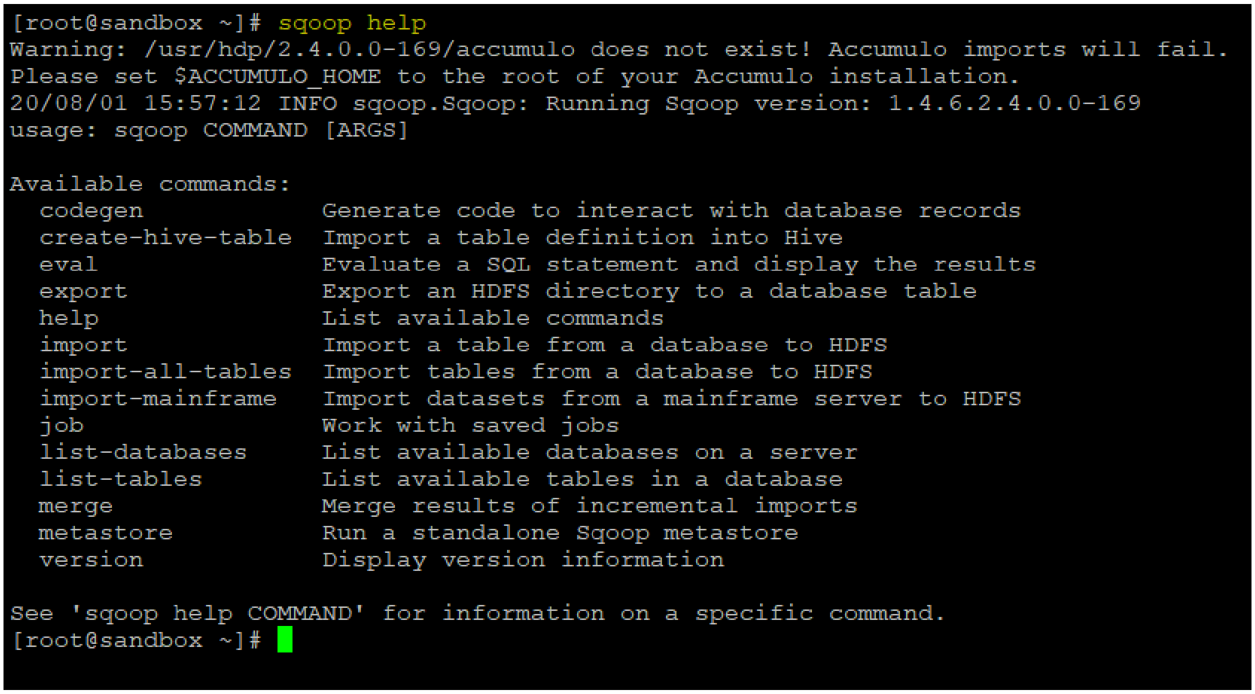


Implementation:

1- Import data from MySQL to HDFS

Open shell on either PuTTY or by going to the <ip of your vm>:4200.

sqoop help:



```
[root@sandbox ~]# sqoop help
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/01 15:57:12 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  import-mainframe   Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information

See 'sqoop help COMMAND' for information on a specific command.
[root@sandbox ~]#
```

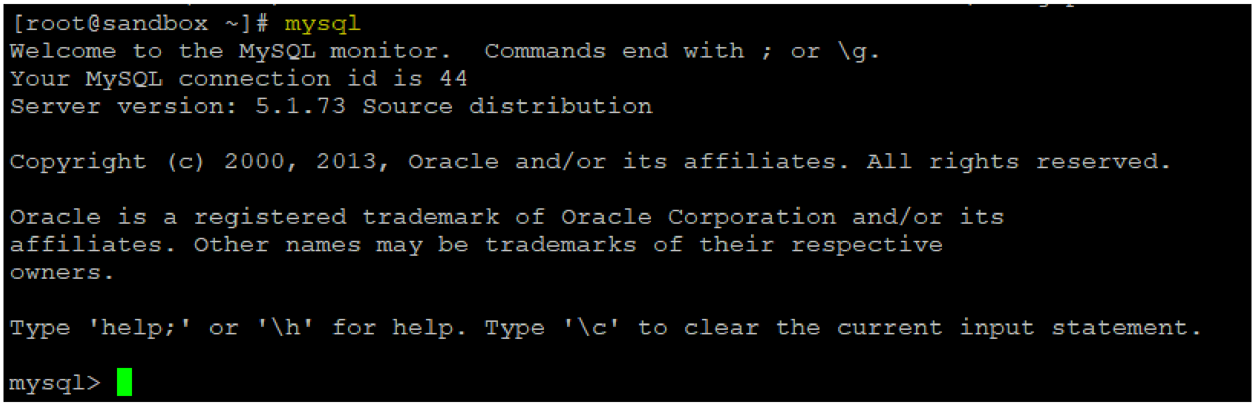
With the help of sqoop help <command> you can see all the functionality that command provides you further commands that you can use with it to refine the data even further.

Try the following 2 commands now:

- Sqoop help import
- Sqoop help export

Open another shell through PuTTY and write mysql and hit enter. Now you have 2 shell windows open: one where you can use mysql and other which is at root.

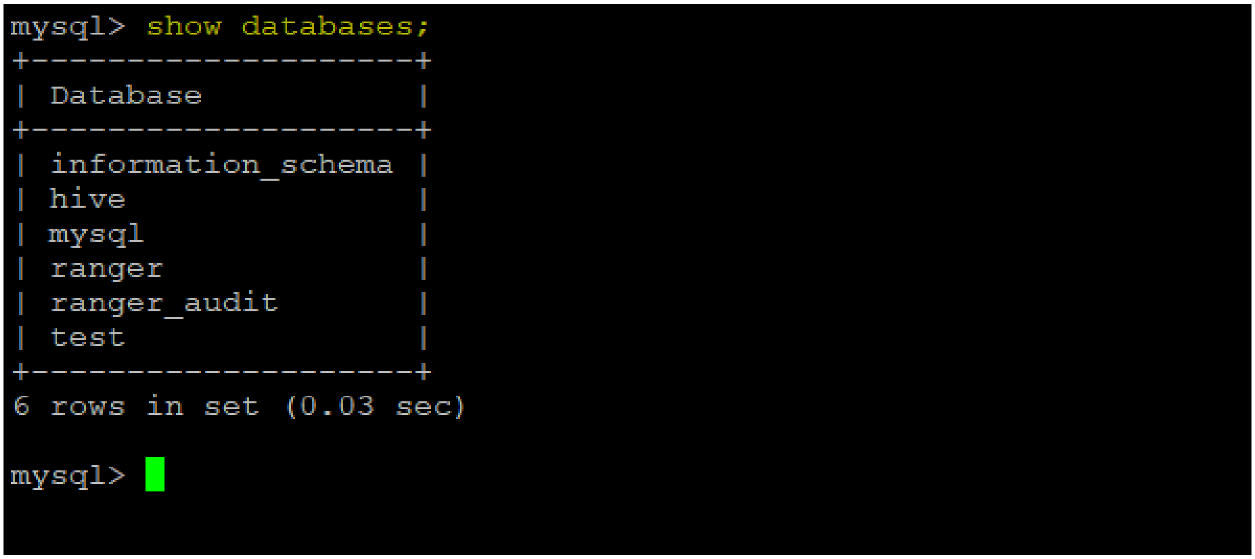
Type mysql to open mysql shell



```
[root@sandbox ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

In the mysql shell:

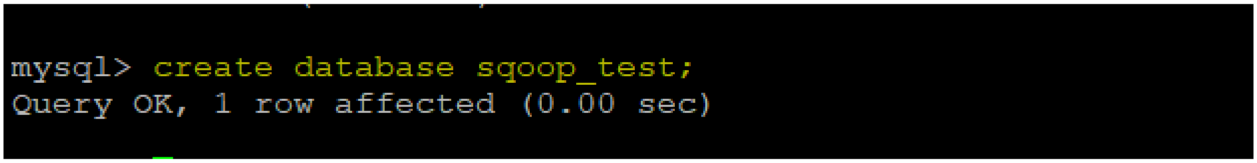
show databases;



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hive               |
| mysql              |
| ranger             |
| ranger_audit       |
| test               |
+--------------------+
6 rows in set (0.03 sec)

mysql>
```

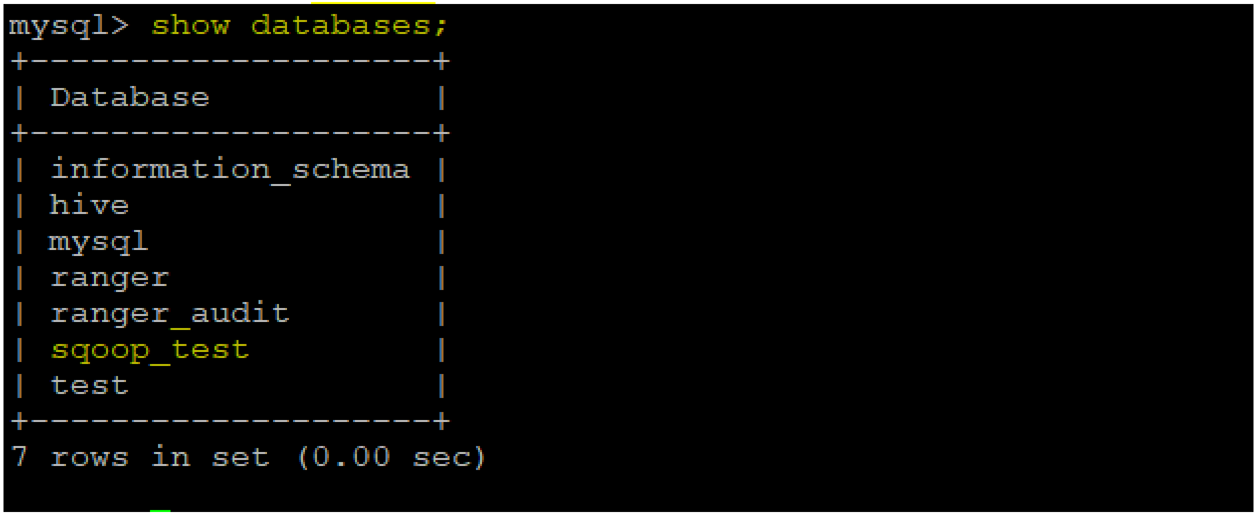
These are the default databases.

create database sqoop_test;



```
mysql> create database sqoop_test;
Query OK, 1 row affected (0.00 sec)
```

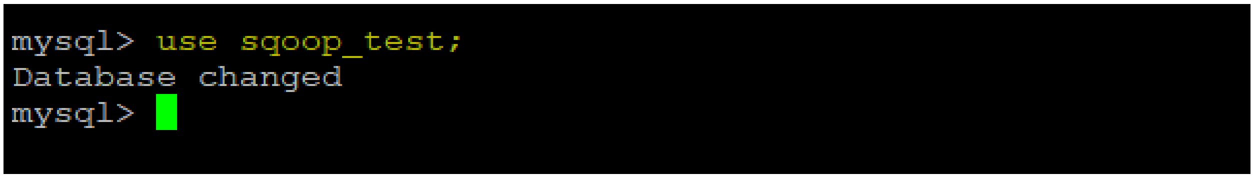
show databases;



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hive               |
| mysql              |
| ranger             |
| ranger_audit       |
| sqoop_test         |
| test               |
+--------------------+
7 rows in set (0.00 sec)
```

Now we will create a new table in our newly created database. There are two methods either change your database to sqoop_test (Database you'll use).

Use <your-database-name>;

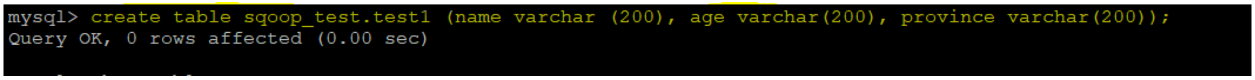


```
mysql> use sqoop_test;
Database changed
mysql>
```

Or

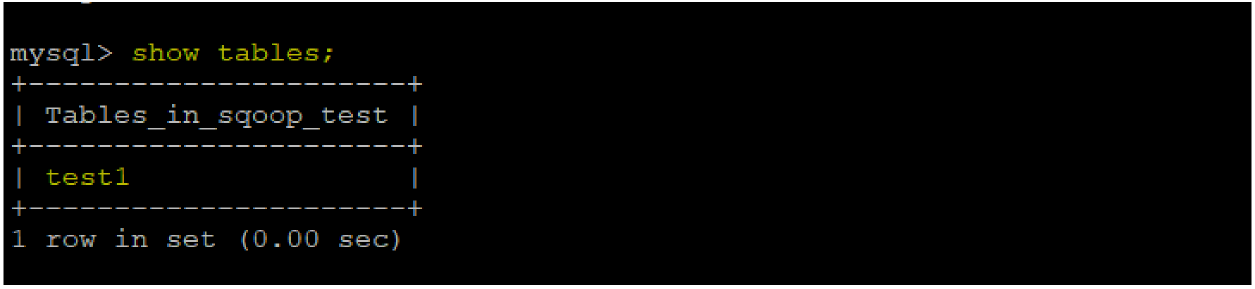
Use the following command this way it will create a table in database sqoop_test by the name of test1. The format for this method is you specify **<database-name>.<table-name>**. If you are in a different database and want to create table in another it will create it and your present working database won't change as well.

create table sqoop_test.test1 (name varchar (200), age varchar(200), province varchar(200));



```
mysql> create table sqoop_test.test1 (name varchar (200), age varchar(200), province varchar(200));
Query OK, 0 rows affected (0.00 sec)
```

Now list table using **show tables** command.

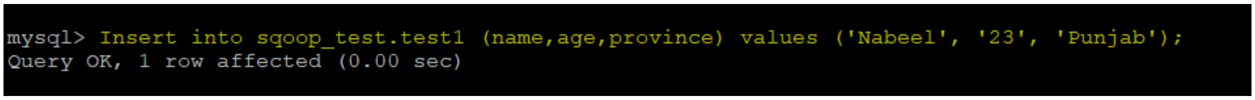


```
mysql> show tables;
+----------------------+
| Tables_in_sqoop_test |
+----------------------+
| test1                |
+----------------------+
1 row in set (0.00 sec)
```

Insert Data in your table:

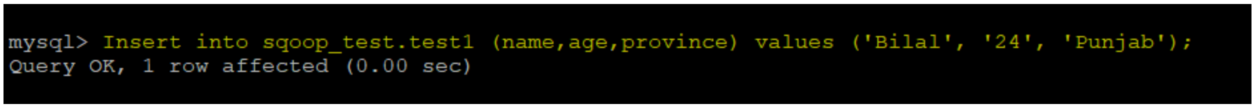
Now we need to add some data to our newly created table. Again if you have used ‘use <your-database-name>’ you won’t need to specify the database name otherwise you’ll have to.

Insert into sqoop_test.test1 (name,age,province) values (‘Nabeel’, ‘23’, ‘Punjab’);



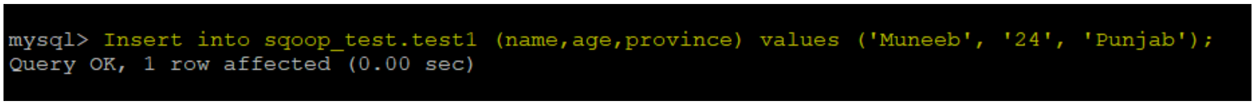
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Nabeel', '23', 'Punjab');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Bilal’, ‘24’, ‘Punjab’);



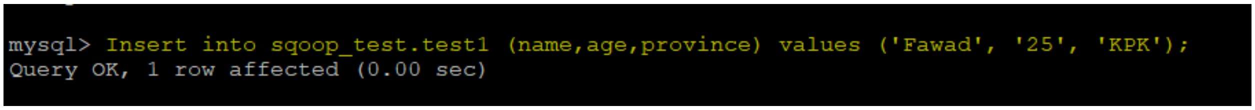
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Bilal', '24', 'Punjab');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Muneeb’, ‘24’, ‘Punjab’);



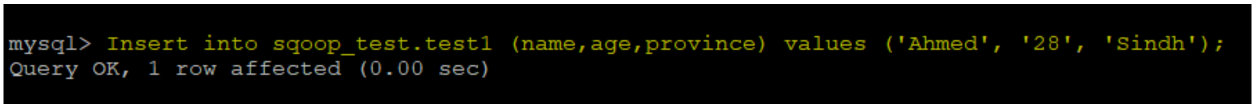
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Muneeb', '24', 'Punjab');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Fawad’, ‘25’, ‘KPK’);



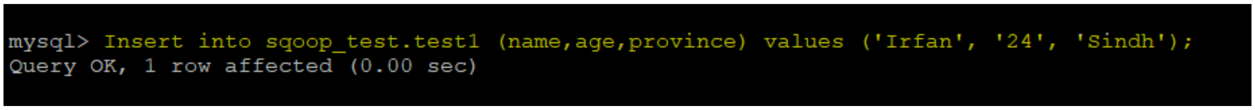
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Fawad', '25', 'KPK');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Ahmed’, ‘28’, ‘Sindh’);



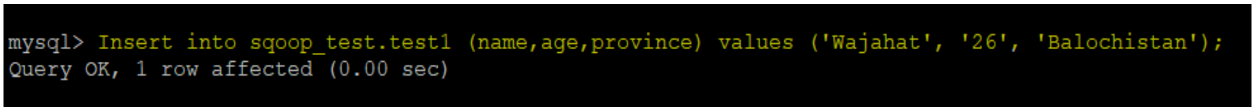
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Ahmed', '28', 'Sindh');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Irfan’, ‘24’, ‘Sindh’);



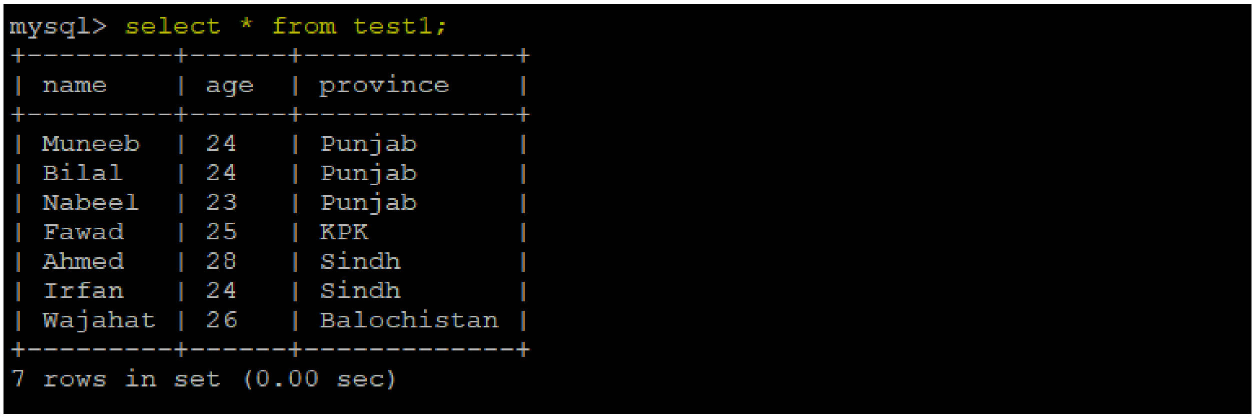
```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Irfan', '24', 'Sindh');
Query OK, 1 row affected (0.00 sec)
```

Insert into sqoop_test.test1 (name,age,province) values (‘Irfan’, ‘24’, ‘Sindh’);



```
mysql> Insert into sqoop_test.test1 (name,age,province) values ('Wajahat', '26', 'Balochistan');
Query OK, 1 row affected (0.00 sec)
```

If you use all 7 of the above insertion commands, you’ll have 7 records in your table.



```
mysql> select * from test1;
+---------+------+-------------+
| name    | age  | province    |
+---------+------+-------------+
| Muneeb  | 24   | Punjab      |
| Bilal   | 24   | Punjab      |
| Nabeel  | 23   | Punjab      |
| Fawad   | 25   | KPK         |
| Ahmed   | 28   | Sindh       |
| Irfan   | 24   | Sindh       |
| Wajahat | 26   | Balochistan |
+---------+------+-------------+
7 rows in set (0.00 sec)
```

There are a few things to note before moving on to the next portion. Sqoop works with connectors and drivers. For this we use

--connect
--driver

As of right now you are on your local machine so these commands will look something as follows:

--connect jdbc:mysql://localhost/sqoop_test

--driver com.mysql.jdbc.Driver

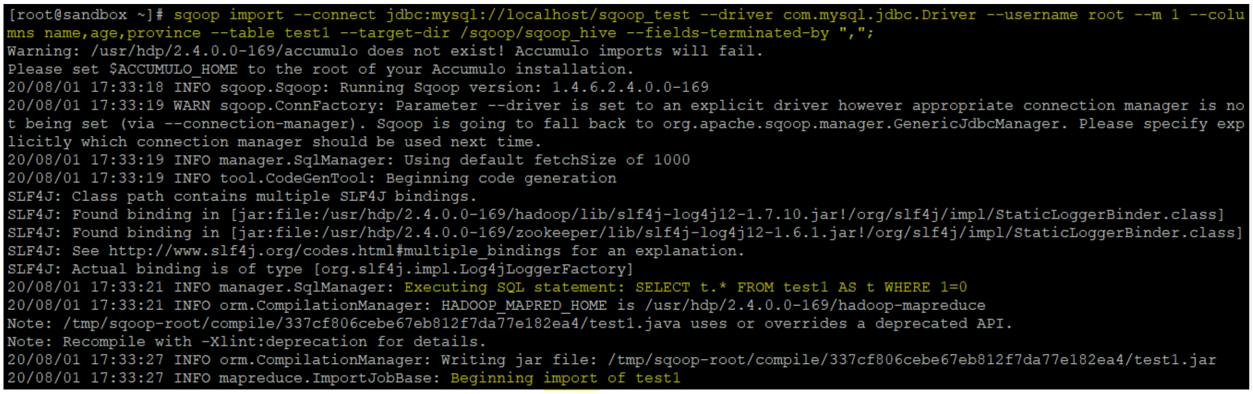
The –connect command will connect you to the sqoop_test database.

Now move to the first shell and on the root directory enter the following command:

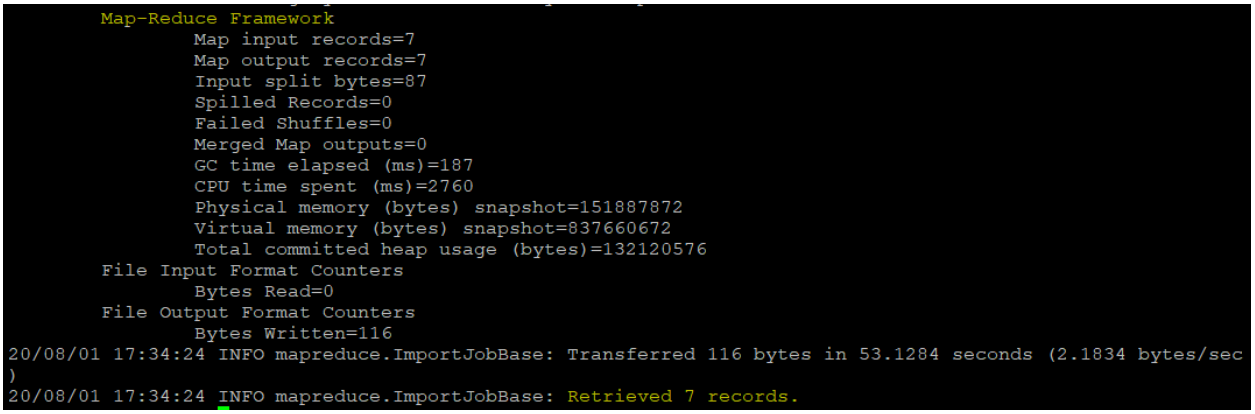
Importing a table into target directory.

```
sqoop import
--connect jdbc:mysql://localhost/sqoop_test
--driver  com.mysql.jdbc.Driver
--username root
--m 1
--columns name,age,province
--table test1
--target-dir /sqoop/sqoop_hive
--fields-terminated-by ","
```

Use Linux Shell to execute sqoop import command.



```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --m 1 --colu
mns name,age,province --table test1 --target-dir /sqoop/sqoop_hive --fields-terminated-by ",";
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/01 17:33:18 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/01 17:33:19 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/01 17:33:19 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/01 17:33:19 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/01 17:33:21 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/01 17:33:21 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.4.0.0-169/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/337cf806cebe67eb812f7da77e182ea4/test1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/08/01 17:33:27 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/337cf806cebe67eb812f7da77e182ea4/test1.jar
20/08/01 17:33:27 INFO mapreduce.ImportJobBase: Beginning import of test1
```



```
        Map-Reduce Framework
                Map input records=7
                Map output records=7
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=187
                CPU time spent (ms)=2760
                Physical memory (bytes) snapshot=151887872
                Virtual memory (bytes) snapshot=837660672
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=116
20/08/01 17:34:24 INFO mapreduce.ImportJobBase: Transferred 116 bytes in 53.1284 seconds (2.1834 bytes/sec
)
20/08/01 17:34:24 INFO mapreduce.ImportJobBase: Retrieved 7 records.
```

Now let’s explain the whole command:

--connect: you tell sqoop where your structured database is located

--driver: you tell sqoop the driver to use for the database

--username: username of the database

(If you are not the admin of the system then you’ll have to additionally have to provide password theough the –password command)

--m: the mappers that will be used for the job (Sqoop will always require for you to specify the mappers you want to use)

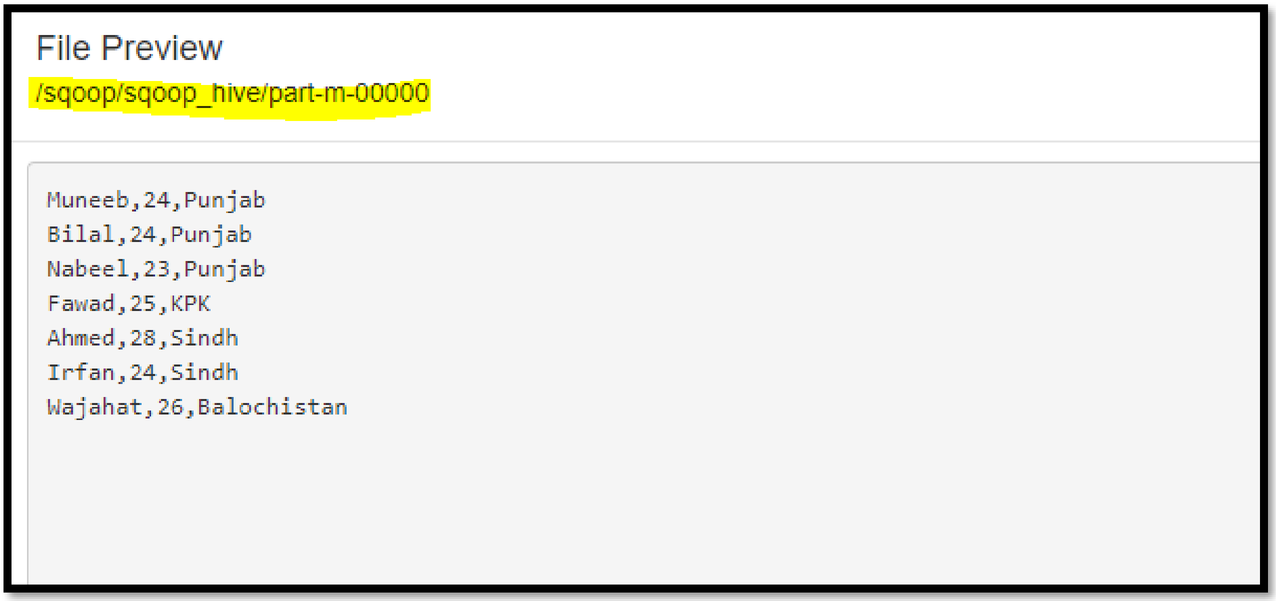
--column: names of the column you want to import

--table name of the table that you want to import from the database

--target-dir: directory in HDFS where the data will be stored

--fields-terminated-by: the basis of which data is separated by

Go to the /sqoop/sqoop_hive directory in HDFS is there any data there?



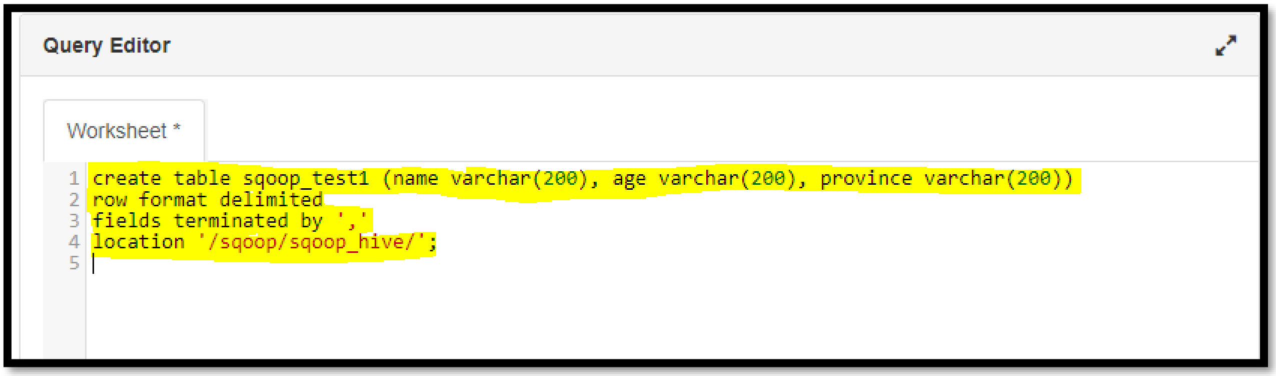
/sqoop/sqoop_hive/part-m-00000

```
Muneeb,24,Punjab
Bilal,24,Punjab
Nabeel,23,Punjab
Fawad,25,KPK
Ahmed,28,Sindh
Irfan,24,Sindh
Wajahat,26,Balochistan
```

Now the data is there but you want to view it in a presentable format.

Now go to give and run the following code on **hive terminal or hive query editor worksheet**:

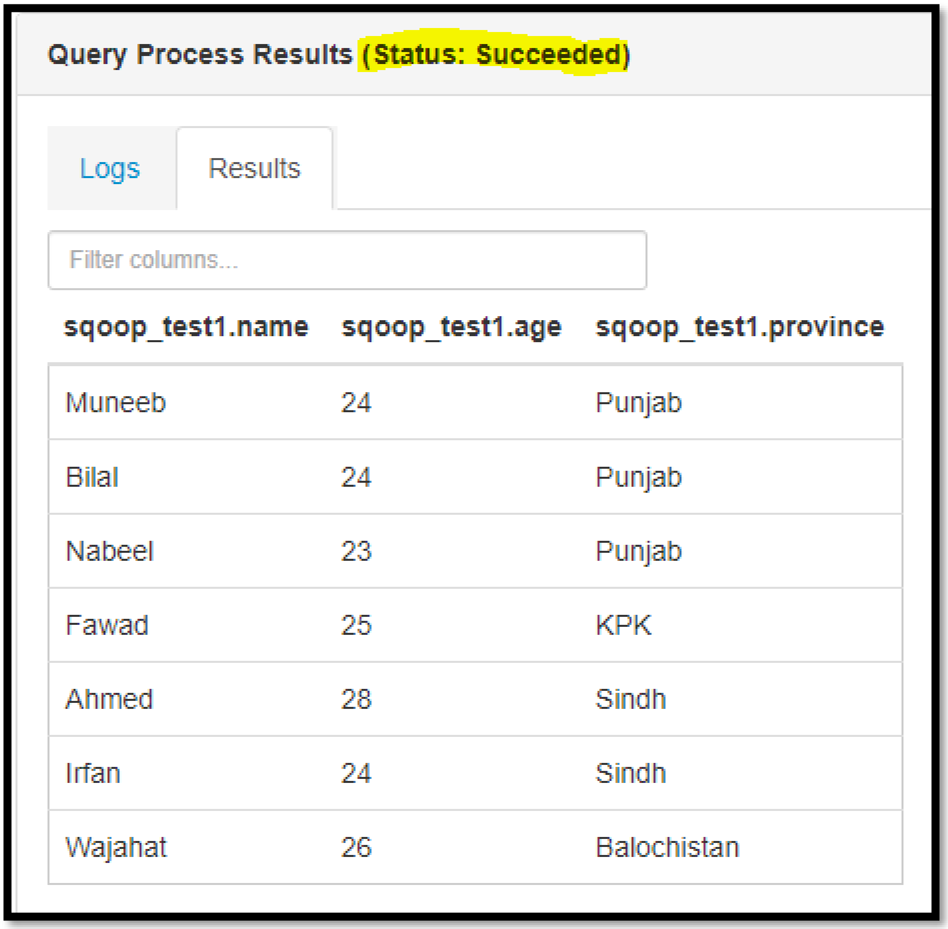
create table sqoop_test1 (name varchar(200), age varchar(200), province varchar(200))
row format delimited
fields terminated by ','
location '/sqoop/sqoop_hive/';



```
create table sqoop_test1 (name varchar(200), age varchar(200), province varchar(200))
row format delimited
fields terminated by ','
location '/sqoop/sqoop_hive/';

```

This will create a table that will be associated to the directory where you imported the data with sqoop.

SELECT * FROM sqoop_test1 LIMIT 100;



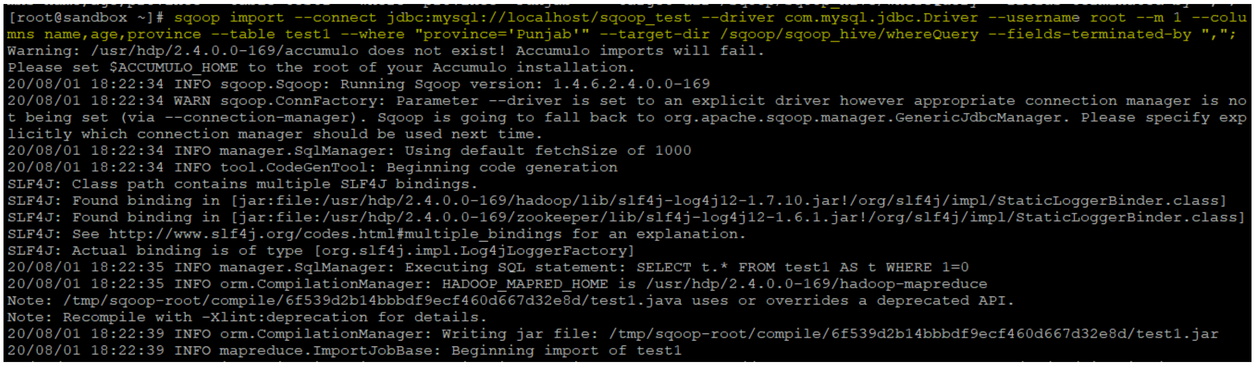
Query Process Results (Status: Succeeded)

sqoop_test1.name	sqoop_test1.age	sqoop_test1.province
Muneeb	24	Punjab
Bilal	24	Punjab
Nabeel	23	Punjab
Fawad	25	KPK
Ahmed	28	Sindh
Irfan	24	Sindh
Wajahat	26	Balochistan

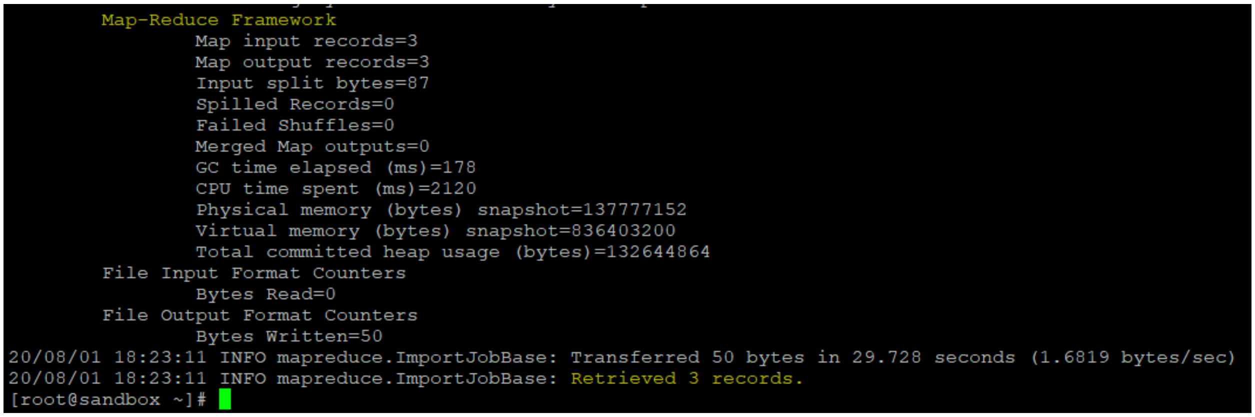
Import Subset of Table Data:

We can import a subset of a table using the 'where' clause in Sqoop import tool. It executes the corresponding SQL query in the respective database server and stores the result in a target directory in HDFS by using **where clause**.

```
sqoop import
--connect jdbc:mysql://localhost/sqoop_test
--driver com.mysql.jdbc.Driver
--username root
--m 1
--columns name,age,province
--table test1
--where "province='Punjab'"
--target-dir /sqoop/sqoop_hive/whereQuery
--fields-terminated-by ",";
```

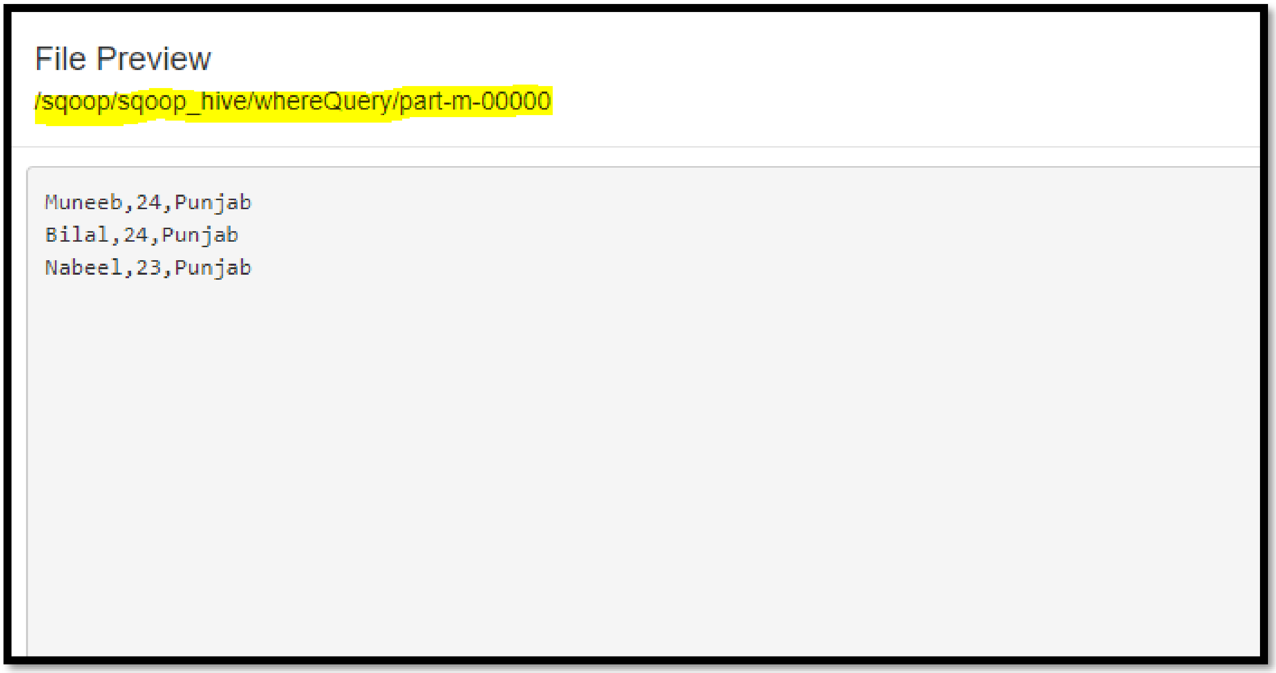


```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --m 1 --colu
mns name,age,province --table test1 --where "province='Punjab'" --target-dir /sqoop/sqoop_hive/whereQuery --fields-terminated-by ",";
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/01 18:22:34 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/01 18:22:34 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/01 18:22:34 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/01 18:22:34 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/01 18:22:35 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/01 18:22:35 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.4.0.0-169/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/6f539d2b14bbbdf9ecf460d667d32e8d/test1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/08/01 18:22:39 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/6f539d2b14bbbdf9ecf460d667d32e8d/test1.jar
20/08/01 18:22:39 INFO mapreduce.ImportJobBase: Beginning import of test1
```



```
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=178
                CPU time spent (ms)=2120
                Physical memory (bytes) snapshot=137777152
                Virtual memory (bytes) snapshot=836403200
                Total committed heap usage (bytes)=132644864
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=50
20/08/01 18:23:11 INFO mapreduce.ImportJobBase: Transferred 50 bytes in 29.728 seconds (1.6819 bytes/sec)
20/08/01 18:23:11 INFO mapreduce.ImportJobBase: Retrieved 3 records.
[root@sandbox ~]#
```

Now check the target directory



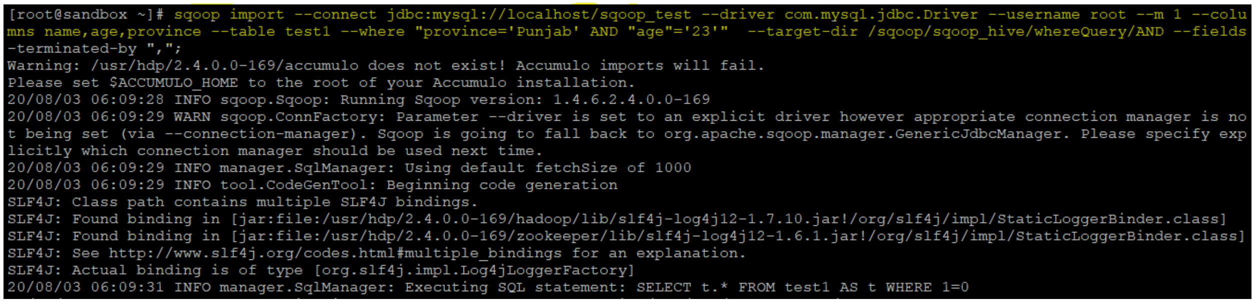
File Preview

/sqoop/sqoop_hive/whereQuery/part-m-00000

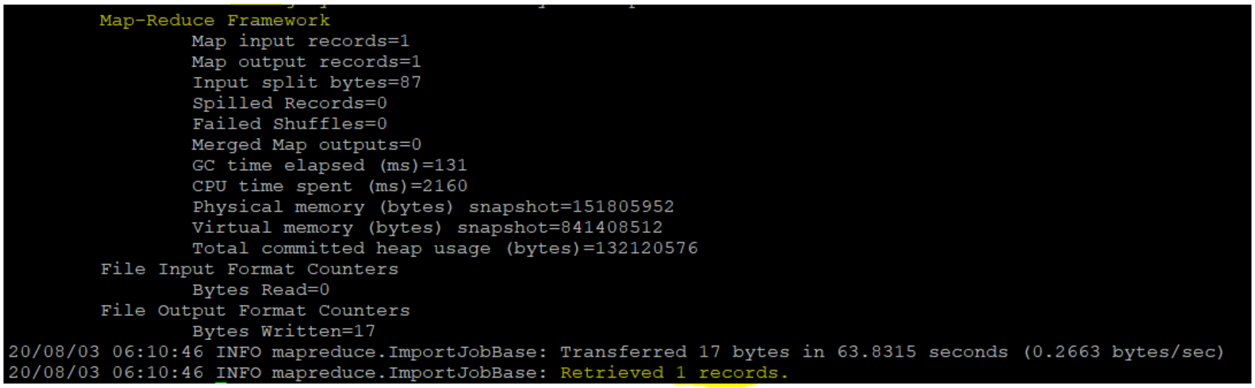
```
Muneeb,24,Punjab
Bilal,24,Punjab
Nabeel,23,Punjab
```

We can also use **AND**, **OR** & **NOT** operator in where clause.

```
sqoop import
--connect jdbc:mysql://localhost/sqoop_test
--driver com.mysql.jdbc.Driver
--username root
--m 1
--columns name,age,province
--table test1
--where "province='Punjab' AND “age”=’23’"
--target-dir /sqoop/sqoop_hive/whereQuery/AND
--fields-terminated-by ",";
```

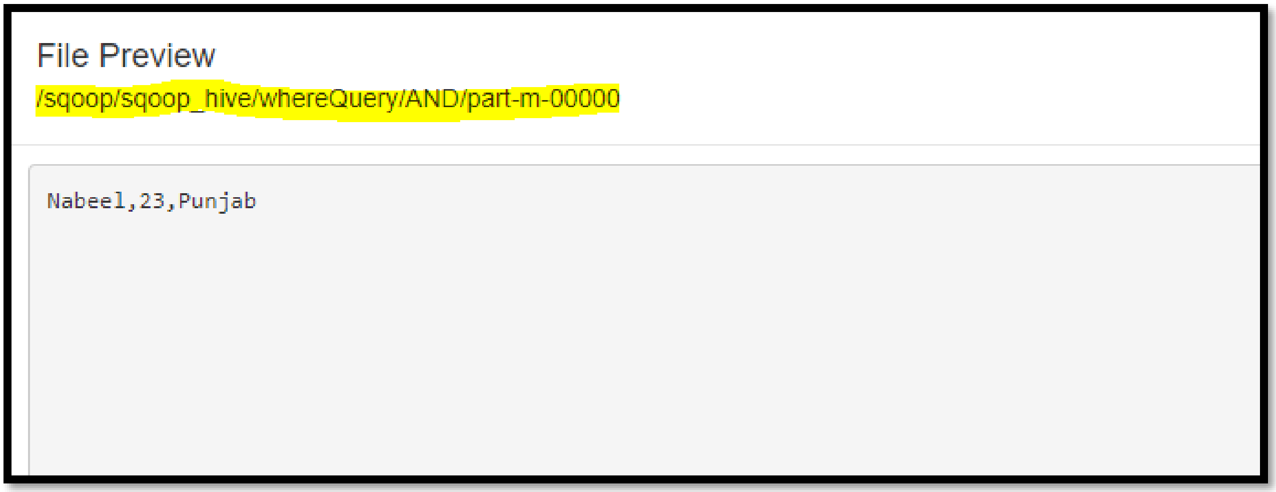


```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --m 1 --colu
mns name,age,province --table test1 --where "province='Punjab' AND "age"='23'"  --target-dir /sqoop/sqoop_hive/whereQuery/AND --fields
-terminated-by ",";
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 06:09:28 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 06:09:29 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licity which connection manager should be used next time.
20/08/03 06:09:29 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 06:09:29 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 06:09:31 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
```



```
        Map-Reduce Framework
                Map input records=1
                Map output records=1
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=131
                CPU time spent (ms)=2160
                Physical memory (bytes) snapshot=151805952
                Virtual memory (bytes) snapshot=841408512
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=17
20/08/03 06:10:46 INFO mapreduce.ImportJobBase: Transferred 17 bytes in 63.8315 seconds (0.2663 bytes/sec)
20/08/03 06:10:46 INFO mapreduce.ImportJobBase: Retrieved 1 records.
```

Only one record is retrieved, as we know that we have only one record which belongs to **Punjab** and his age was **23**. Now let’s validate.



File Preview

/sqoop/sqoop_hive/whereQuery/AND/part-m-00000

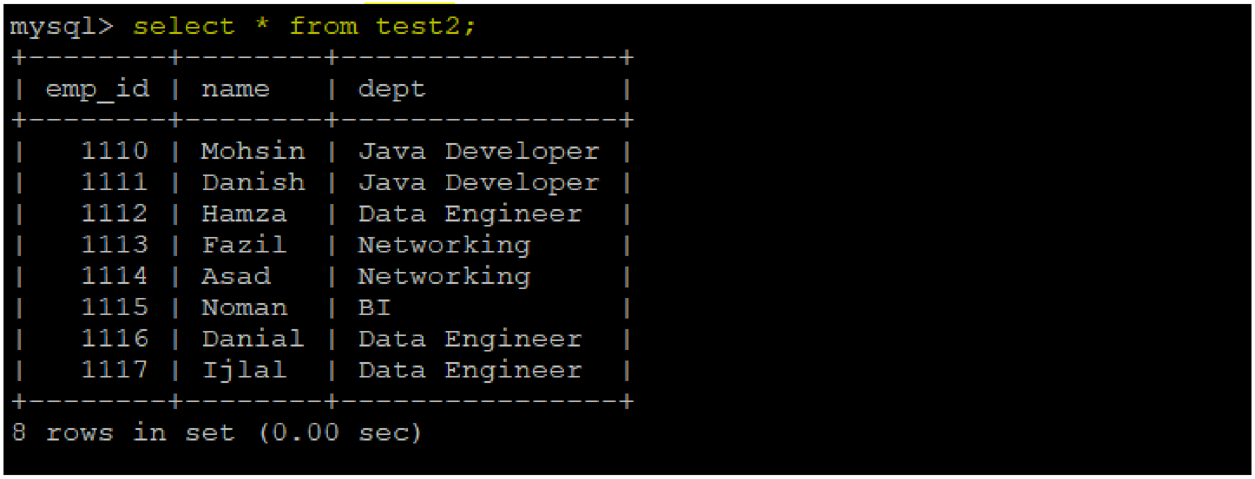
```
Nabeel,23,Punjab
```

Incremental Import:

It is a technique to import only the newly added rows in a table. It is required to add 'incremental', 'check-column', and 'last-value' options to perform the incremental import. So, for that first we have to add a new record in the table.

The incremental import is not supported for CHAR, VARCHAR data types. You can use integer or date datatypes for -check-column.

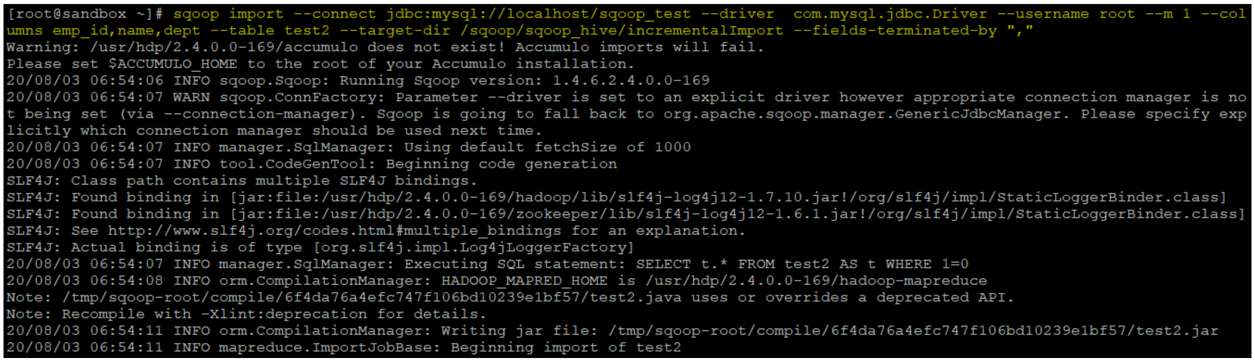
So, I created **test2** table in same database **sqoop_test** (mysql). The columns are emp_id(int), name(varchar), dept(varchar). So now insert some values in this table.



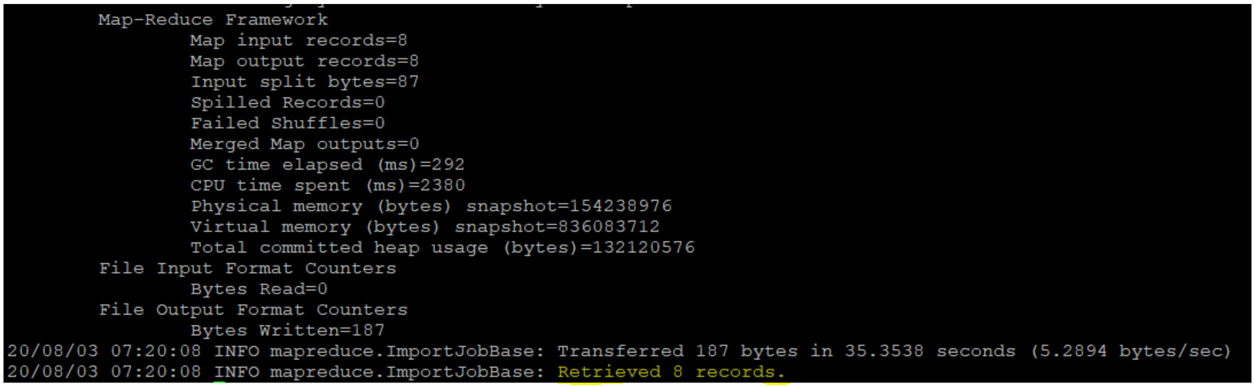
```
mysql> select * from test2;
+--------+--------+----------------+
| emp_id | name   | dept           |
+--------+--------+----------------+
|   1110 | Mohsin | Java Developer |
|   1111 | Danish | Java Developer |
|   1112 | Hamza  | Data Engineer  |
|   1113 | Fazil  | Networking     |
|   1114 | Asad   | Networking     |
|   1115 | Noman  | BI             |
|   1116 | Danial | Data Engineer  |
|   1117 | Ijlal  | Data Engineer  |
+--------+--------+----------------+
8 rows in set (0.00 sec)
```

Now import the table into HDFS directory without using incremental import technique.

sqoop import
--connect jdbc:mysql://localhost/sqoop_test
--driver com.mysql.jdbc.Driver --username root
--m 1
--columns emp_id,name,dept
--table test2
--target-dir /sqoop/sqoop_hive/incrementalImport
--fields-terminated-by ","

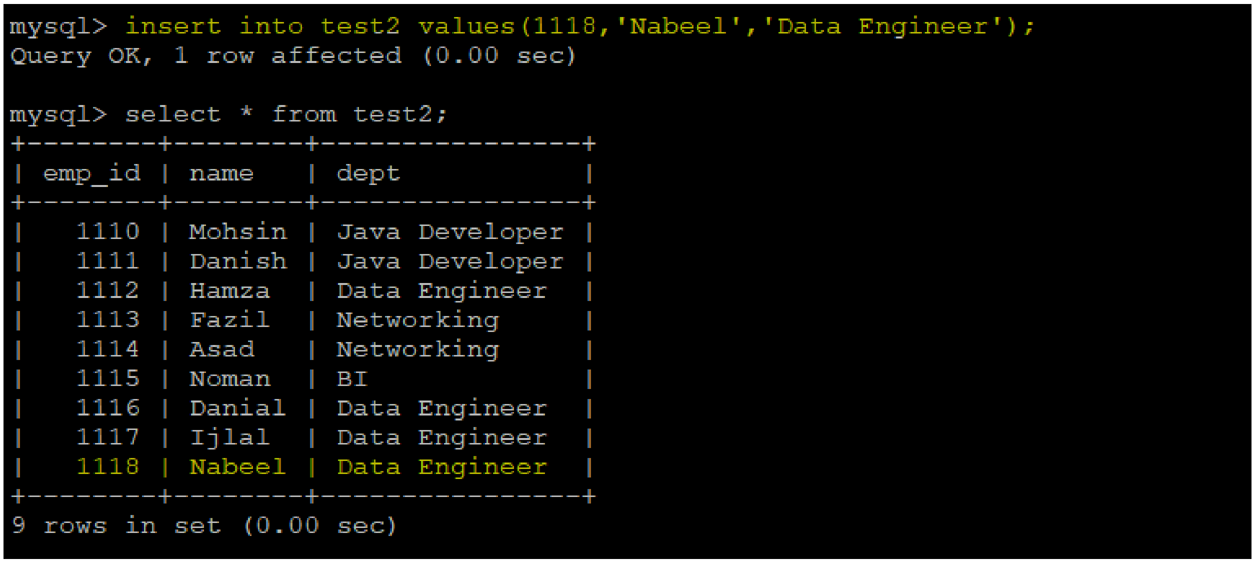


```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver  com.mysql.jdbc.Driver --username root --m 1 --col
umns emp_id,name,dept --table test2 --target-dir /sqoop/sqoop_hive/incrementalImport --fields-terminated-by ","
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 06:54:06 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 06:54:07 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 06:54:07 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 06:54:07 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 06:54:07 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test2 AS t WHERE 1=0
20/08/03 06:54:08 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.4.0.0-169/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/6f4da76a4efc747f106bd10239e1bf57/test2.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/08/03 06:54:11 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/6f4da76a4efc747f106bd10239e1bf57/test2.jar
20/08/03 06:54:11 INFO mapreduce.ImportJobBase: Beginning import of test2
```



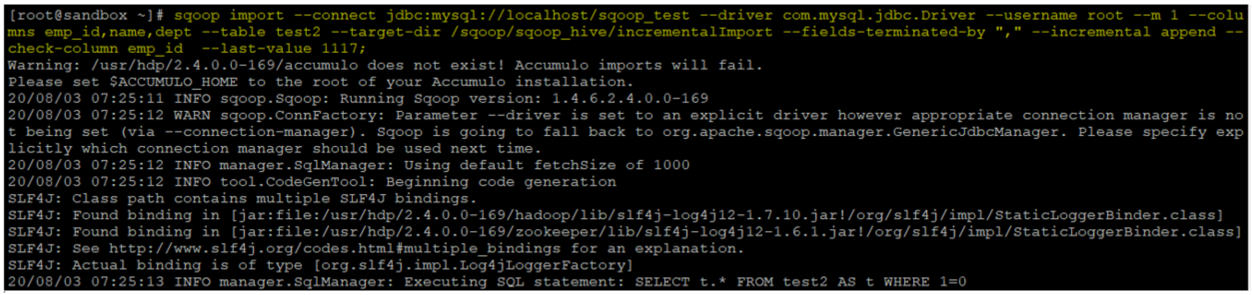
```
        Map-Reduce Framework
                Map input records=8
                Map output records=8
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=292
                CPU time spent (ms)=2380
                Physical memory (bytes) snapshot=154238976
                Virtual memory (bytes) snapshot=836083712
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=187
20/08/03 07:20:08 INFO mapreduce.ImportJobBase: Transferred 187 bytes in 35.3538 seconds (5.2894 bytes/sec)
20/08/03 07:20:08 INFO mapreduce.ImportJobBase: Retrieved 8 records.
```

Now add a new row to test2 table and use incremental import approach.

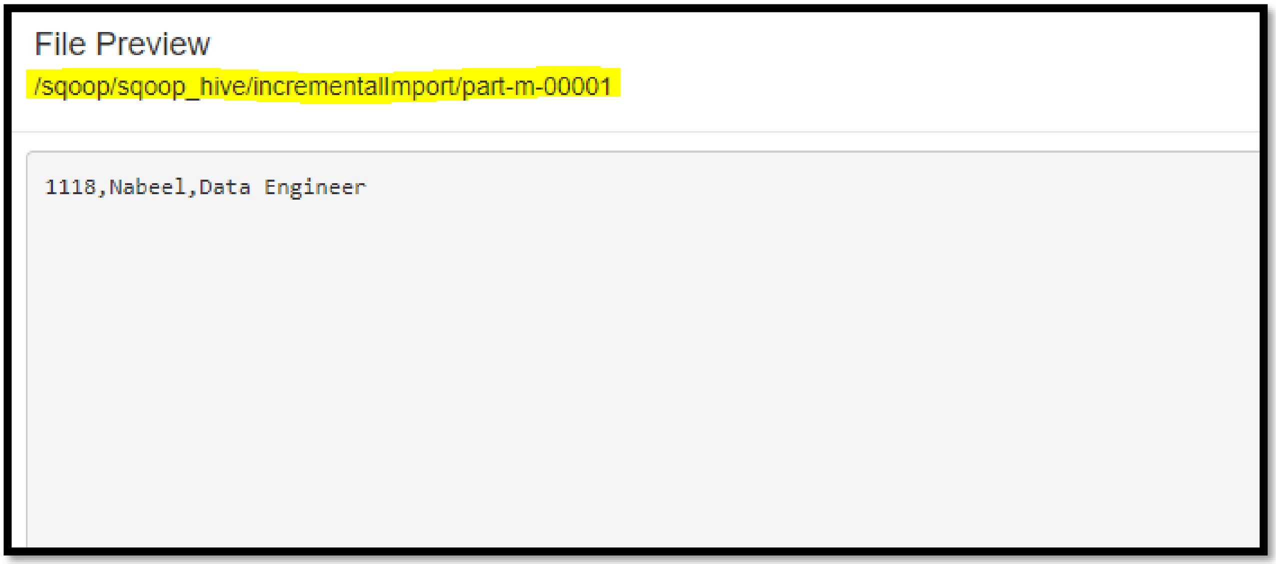


```
mysql> insert into test2 values(1118,'Nabeel','Data Engineer');
Query OK, 1 row affected (0.00 sec)

mysql> select * from test2;
+--------+--------+----------------+
| emp_id | name   | dept           |
+--------+--------+----------------+
|   1110 | Mohsin | Java Developer |
|   1111 | Danish | Java Developer |
|   1112 | Hamza  | Data Engineer  |
|   1113 | Fazil  | Networking     |
|   1114 | Asad   | Networking     |
|   1115 | Noman  | BI             |
|   1116 | Danial | Data Engineer  |
|   1117 | Ijlal  | Data Engineer  |
|   1118 | Nabeel | Data Engineer  |
+--------+--------+----------------+
9 rows in set (0.00 sec)
```

sqoop import
--connect jdbc:mysql://localhost/sqoop_test
--driver com.mysql.jdbc.Driver
--username root
--m 1
--columns emp_id,name,dept
--table test2
--target-dir /sqoop/sqoop_hive/incrementalImport
--fields-terminated-by ","
--incremental append
--check-column emp_id
--last-value 1117;



```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --m 1 --colu
mns emp_id,name,dept --table test2 --target-dir /sqoop/sqoop_hive/incrementalImport --fields-terminated-by "," --incremental append --
check-column emp_id  --last-value 1117;
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 07:25:11 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 07:25:12 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 07:25:12 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 07:25:12 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 07:25:13 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test2 AS t WHERE 1=0
```



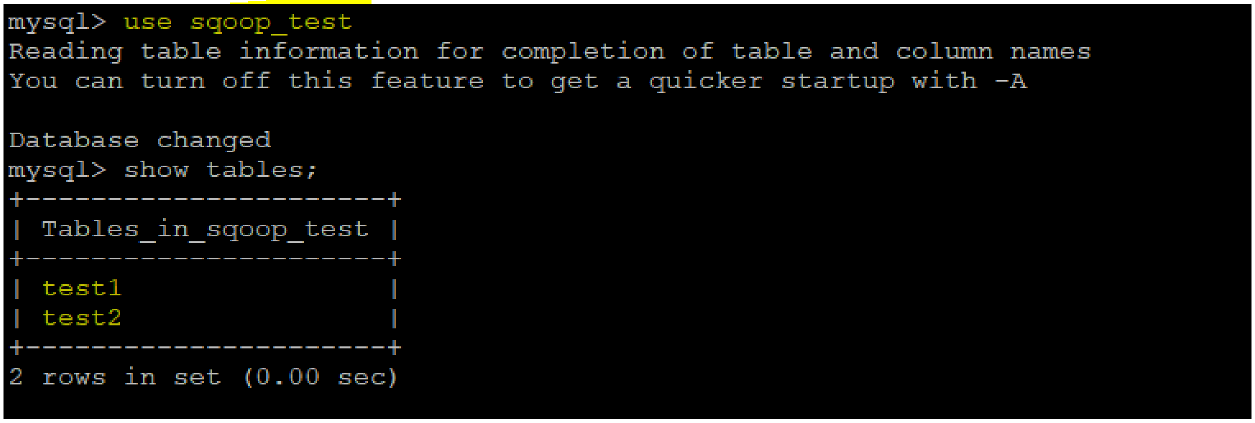
File Preview

/sqoop/sqoop_hive/incrementalImport/part-m-00001

```
1118,Nabeel,Data Engineer
```

Import all tables:

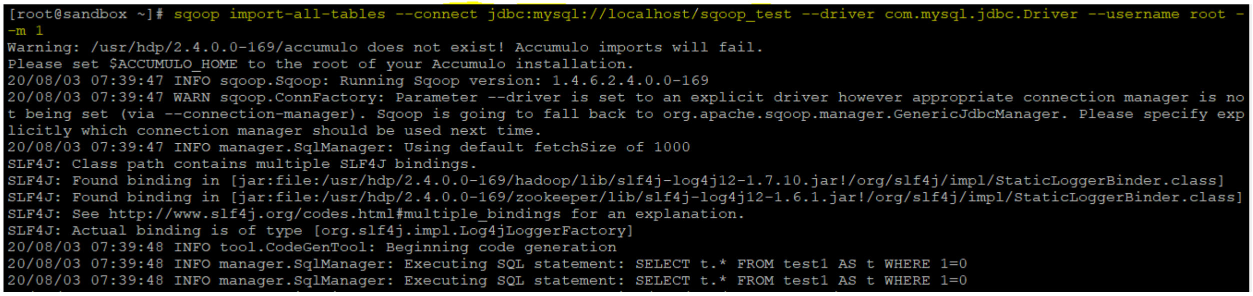
If we want to import all table from a database(mysql) then we use sqoop import all table command. Each table will be imported as a directory in HDFS lets see an example.



```
mysql> use sqoop_test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------------+
| Tables_in_sqoop_test |
+----------------------+
| test1                |
| test2                |
+----------------------+
2 rows in set (0.00 sec)
```

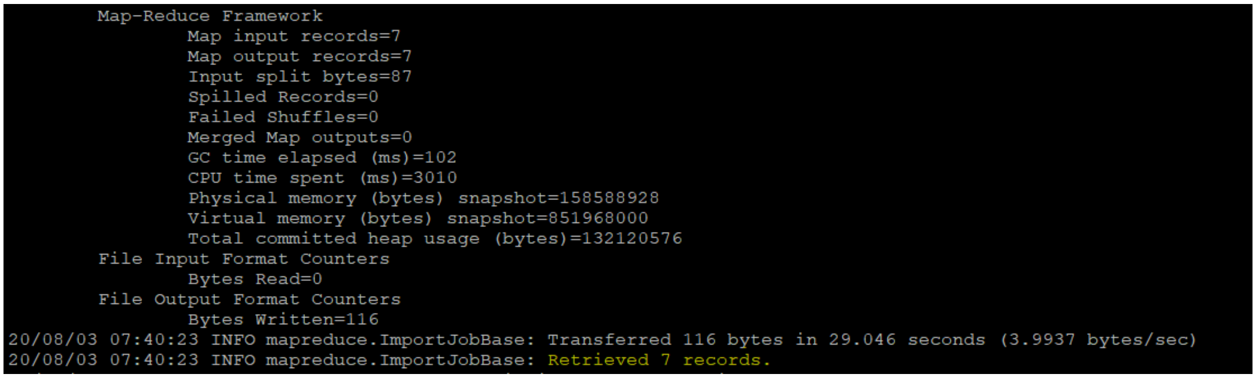
We have only 2 tables in sqoop_test database. Lets import it to HDFS

```
sqoop import-all-tables
--connect jdbc:mysql://localhost/sqoop_test
--driver com.mysql.jdbc.Driver
--username root
--m 1
```



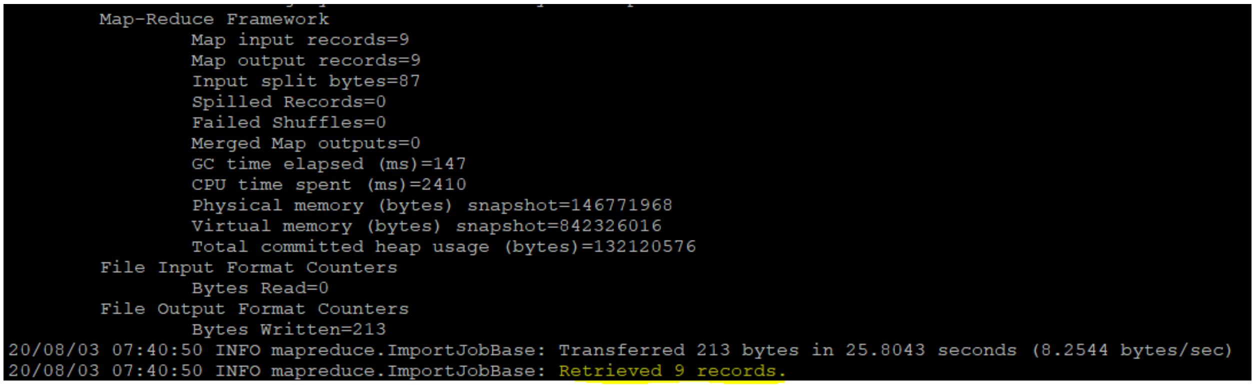
```
[root@sandbox ~]# sqoop import-all-tables --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root -
-m 1
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 07:39:47 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 07:39:47 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 07:39:47 INFO manager.SqlManager: Using default fetchSize of 1000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 07:39:48 INFO tool.CodeGenTool: Beginning code generation
20/08/03 07:39:48 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/03 07:39:48 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
```

As we have two tables so there will be two jobs for importing which run sequentially.



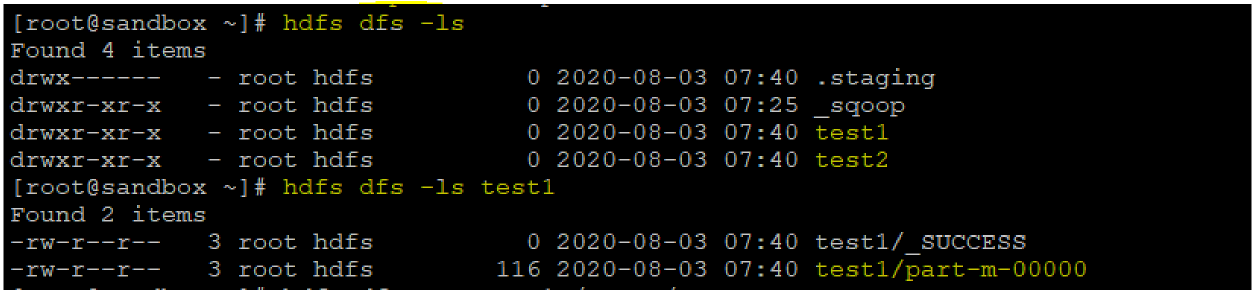
```
        Map-Reduce Framework
                Map input records=7
                Map output records=7
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=102
                CPU time spent (ms)=3010
                Physical memory (bytes) snapshot=158588928
                Virtual memory (bytes) snapshot=851968000
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=116
20/08/03 07:40:23 INFO mapreduce.ImportJobBase: Transferred 116 bytes in 29.046 seconds (3.9937 bytes/sec)
20/08/03 07:40:23 INFO mapreduce.ImportJobBase: Retrieved 7 records.
```

And here is the import of second table:



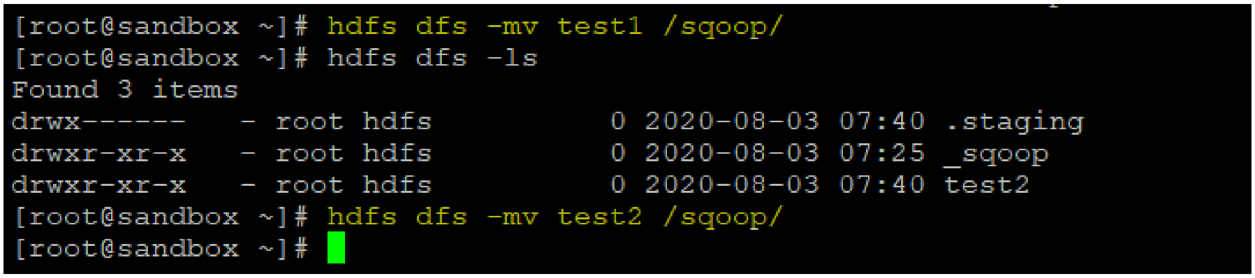
```
        Map-Reduce Framework
                Map input records=9
                Map output records=9
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=147
                CPU time spent (ms)=2410
                Physical memory (bytes) snapshot=146771968
                Virtual memory (bytes) snapshot=842326016
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=213
20/08/03 07:40:50 INFO mapreduce.ImportJobBase: Transferred 213 bytes in 25.8043 seconds (8.2544 bytes/sec)
20/08/03 07:40:50 INFO mapreduce.ImportJobBase: Retrieved 9 records.
```

Now lets validate what we have done. So for that we have to use the ls command so it will be

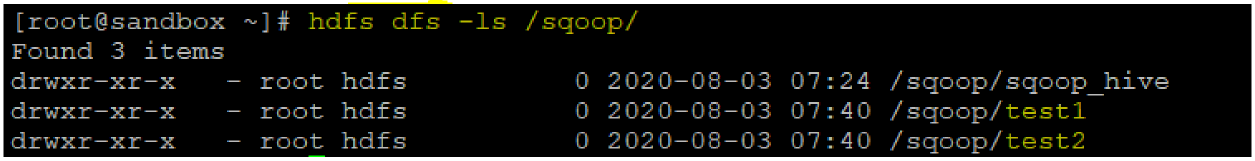


```
[root@sandbox ~]# hdfs dfs -ls
Found 4 items
drwx------   - root hdfs          0 2020-08-03 07:40 .staging
drwxr-xr-x   - root hdfs          0 2020-08-03 07:25 _sqoop
drwxr-xr-x   - root hdfs          0 2020-08-03 07:40 test1
drwxr-xr-x   - root hdfs          0 2020-08-03 07:40 test2
[root@sandbox ~]# hdfs dfs -ls test1
Found 2 items
-rw-r--r--   3 root hdfs          0 2020-08-03 07:40 test1/_SUCCESS
-rw-r--r--   3 root hdfs        116 2020-08-03 07:40 test1/part-m-00000
```

If we want to move them to desired directory then the command will be



```
[root@sandbox ~]# hdfs dfs -mv test1 /sqoop/
[root@sandbox ~]# hdfs dfs -ls
Found 3 items
drwx------   - root hdfs          0 2020-08-03 07:40 .staging
drwxr-xr-x   - root hdfs          0 2020-08-03 07:25 _sqoop
drwxr-xr-x   - root hdfs          0 2020-08-03 07:40 test2
[root@sandbox ~]# hdfs dfs -mv test2 /sqoop/
[root@sandbox ~]# 
```

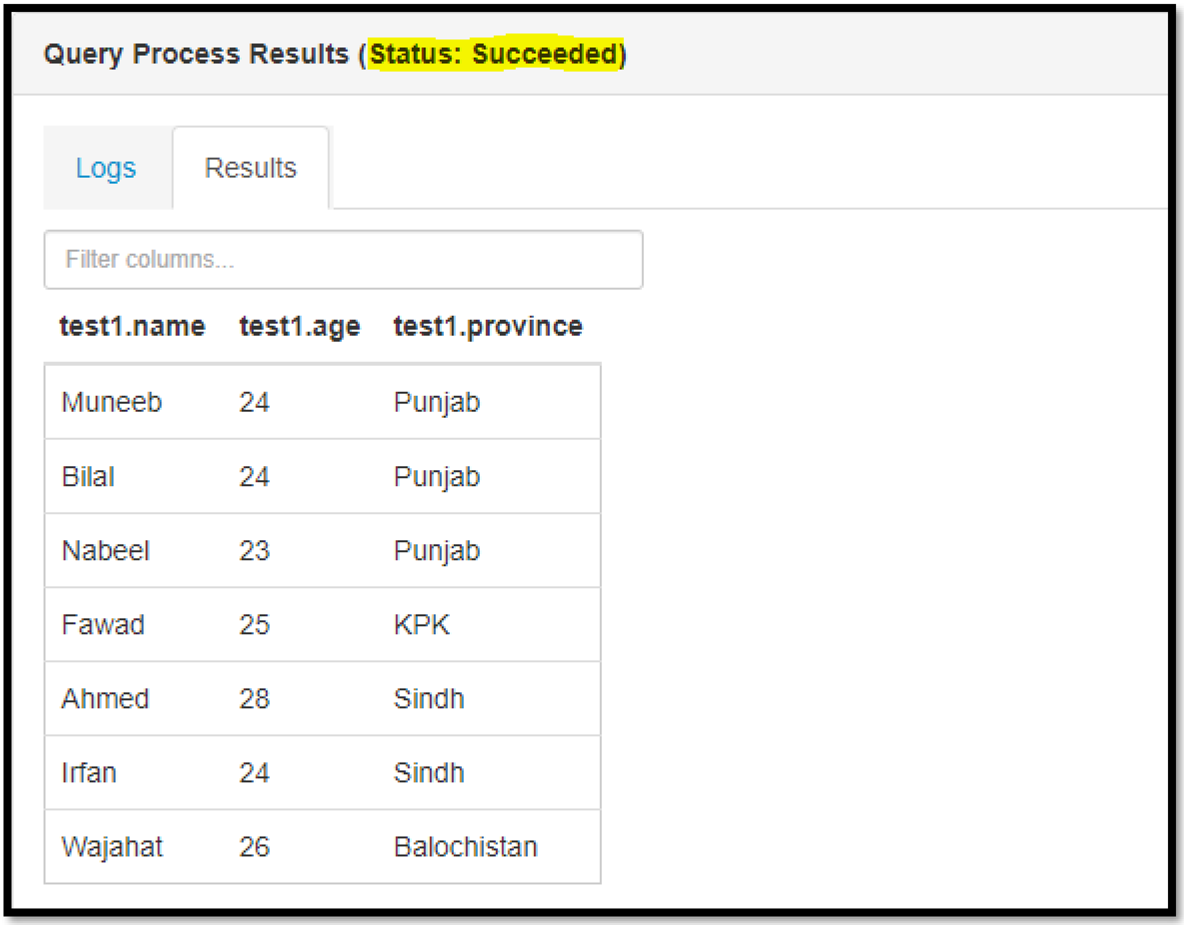


```
[root@sandbox ~]# hdfs dfs -ls /sqoop/
Found 3 items
drwxr-xr-x   - root hdfs          0 2020-08-03 07:24 /sqoop/sqoop_hive
drwxr-xr-x   - root hdfs          0 2020-08-03 07:40 /sqoop/test1
drwxr-xr-x   - root hdfs          0 2020-08-03 07:40 /sqoop/test2
```

Now if we want to load them into a table so we have to create a table structure so it will be look like this

create table test1 (name varchar(200), age int, province varchar(200)) row format delimited fields terminated by ',' location '/sqoop/test1/';

Now try Select * and the output will be.



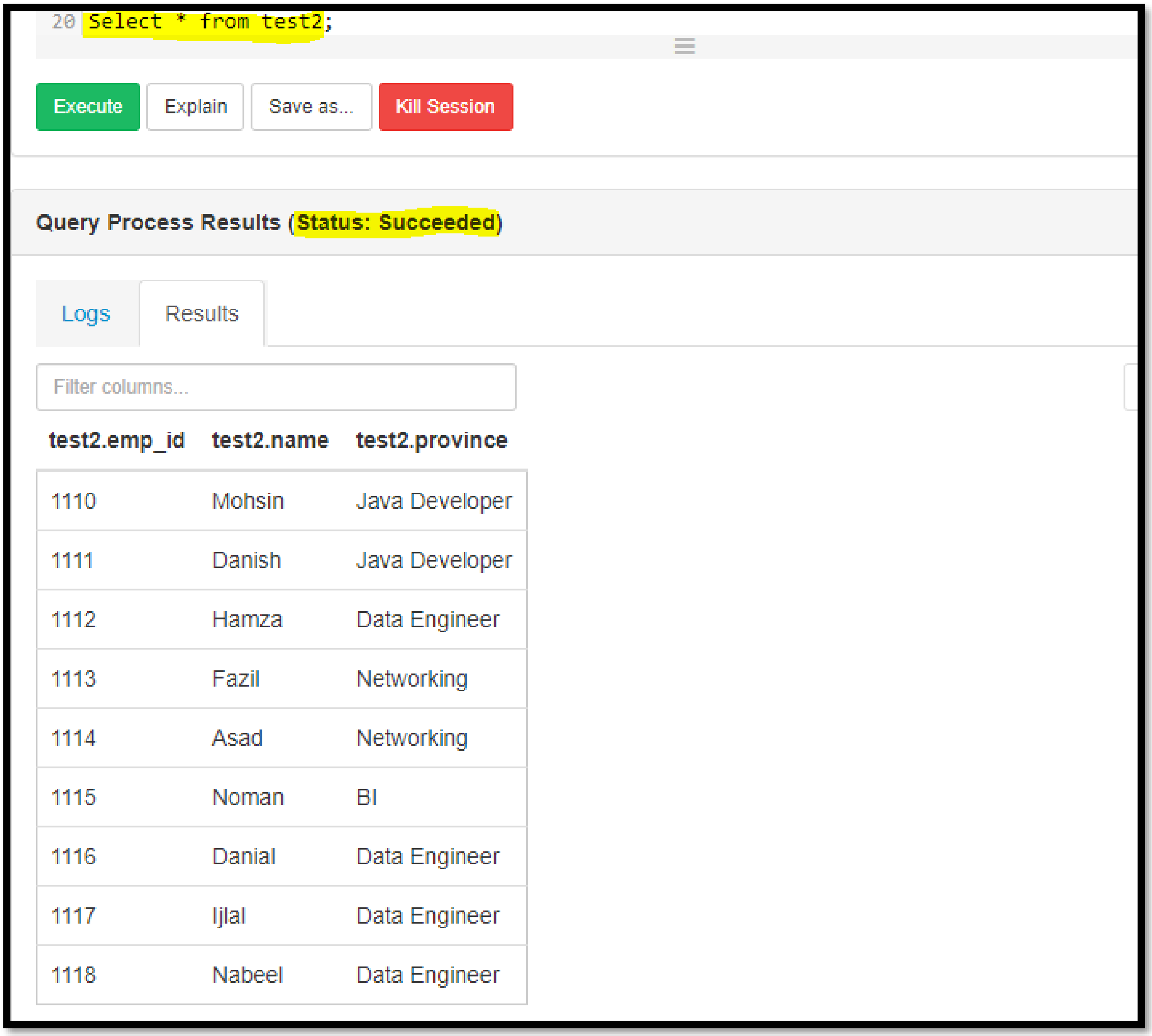
Query Process Results (Status: Succeeded)

test1.name	test1.age	test1.province
Muneeb	24	Punjab
Bilal	24	Punjab
Nabeel	23	Punjab
Fawad	25	KPK
Ahmed	28	Sindh
Irfan	24	Sindh
Wajahat	26	Balochistan

And now create DDL for test2 table

create table test2 (emp_id int, name string, province string) row format delimited fields terminated by ',' location '/sqoop/test2/';

Now use select clause to retrieve all the records



```
Select * from test2;
```

Query Process Results (Status: Succeeded)

test2.emp_id	test2.name	test2.province
1110	Mohsin	Java Developer
1111	Danish	Java Developer
1112	Hamza	Data Engineer
1113	Fazil	Networking
1114	Asad	Networking
1115	Noman	BI
1116	Danial	Data Engineer
1117	Ijlal	Data Engineer
1118	Nabeel	Data Engineer

2- Import data from MySQL to Hive

Now you can skip the hive table creation process. You can create a table from the sqoop as well. Run the following command:

sqoop import

--connect jdbc:mysql://localhost/sqoop_test

--driver com.mysql.jdbc.Driver

--username root

--m 1

--columns name,age,province

--table test1

--target-dir /usr/sqoop/hive2

--fields-terminated-by ","

--hive-import

--create-hive-table

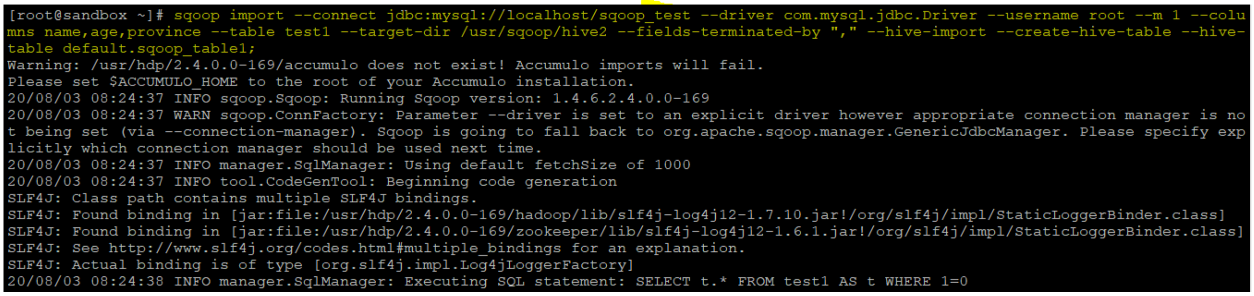
--hive-table default.sqoop_test_table1

There are three new identifiers here:

--hive-import (This creates a small hive instance in the sqoop where you can use other DDL functions)

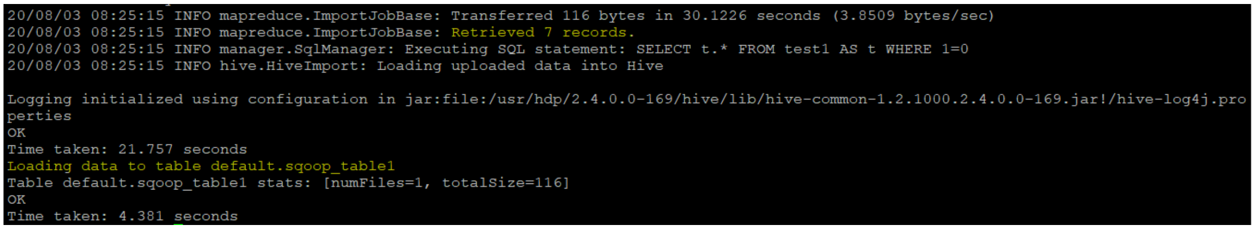
--create-hive-table (This is to create the database in hive)

--hive-table default.sqoop_test2 (specifying table name and database in which table will be created)

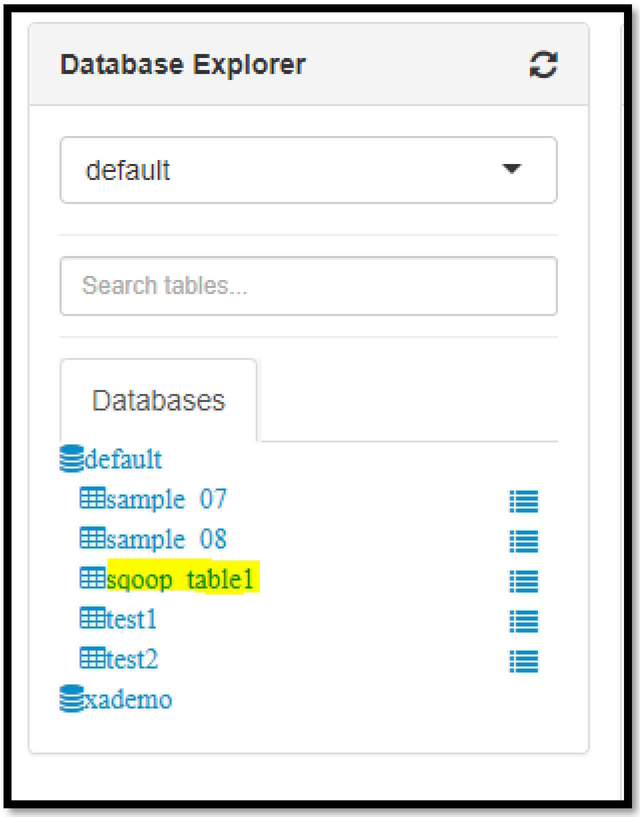


```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --m 1 --colu
mns name,age,province --table test1 --target-dir /usr/sqoop/hive2 --fields-terminated-by "," --hive-import --create-hive-table --hive-
table default.sqoop_table1;
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 08:24:37 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 08:24:37 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 08:24:37 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 08:24:37 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 08:24:38 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
```

The above will create a table in the default database in hive and also associate the table with the directory mentioned as well.



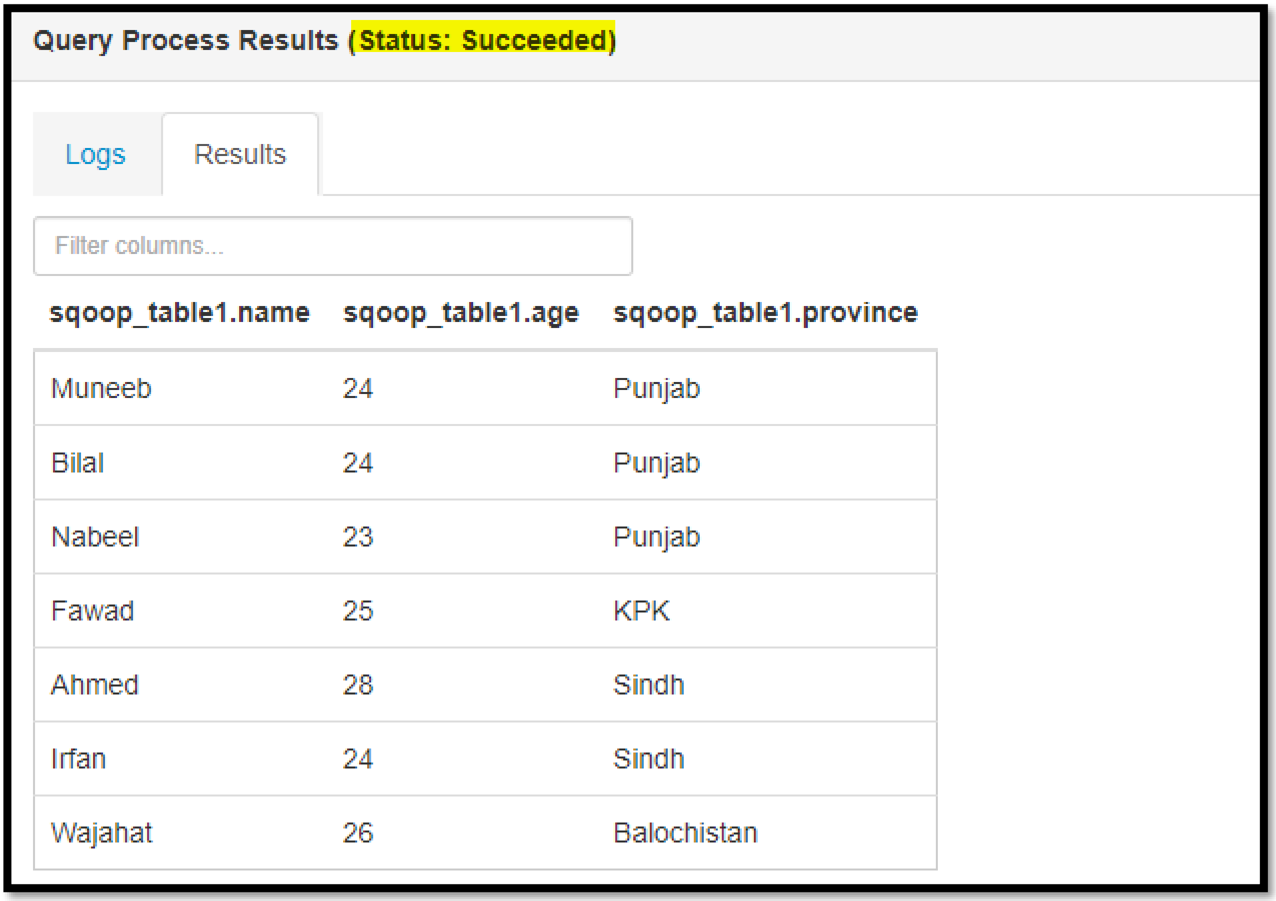
```
20/08/03 08:25:15 INFO mapreduce.ImportJobBase: Transferred 116 bytes in 30.1226 seconds (3.8509 bytes/sec)
20/08/03 08:25:15 INFO mapreduce.ImportJobBase: Retrieved 7 records.
20/08/03 08:25:15 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/03 08:25:15 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/hdp/2.4.0.0-169/hive/lib/hive-common-1.2.1000.2.4.0.0-169.jar!/hive-log4j.pro
perties
OK
Time taken: 21.757 seconds
Loading data to table default.sqoop_table1
Table default.sqoop_table1 stats: [numFiles=1, totalSize=116]
OK
Time taken: 4.381 seconds
```



- default
  - sample_07
  - sample_08
  - sqoop_table1
  - test1
  - test2
- xademo

Go to the directory /usr/sqoop/hive2/ and see if there’s anything there?

Now check if the table was actually created in hive and is the data correct?

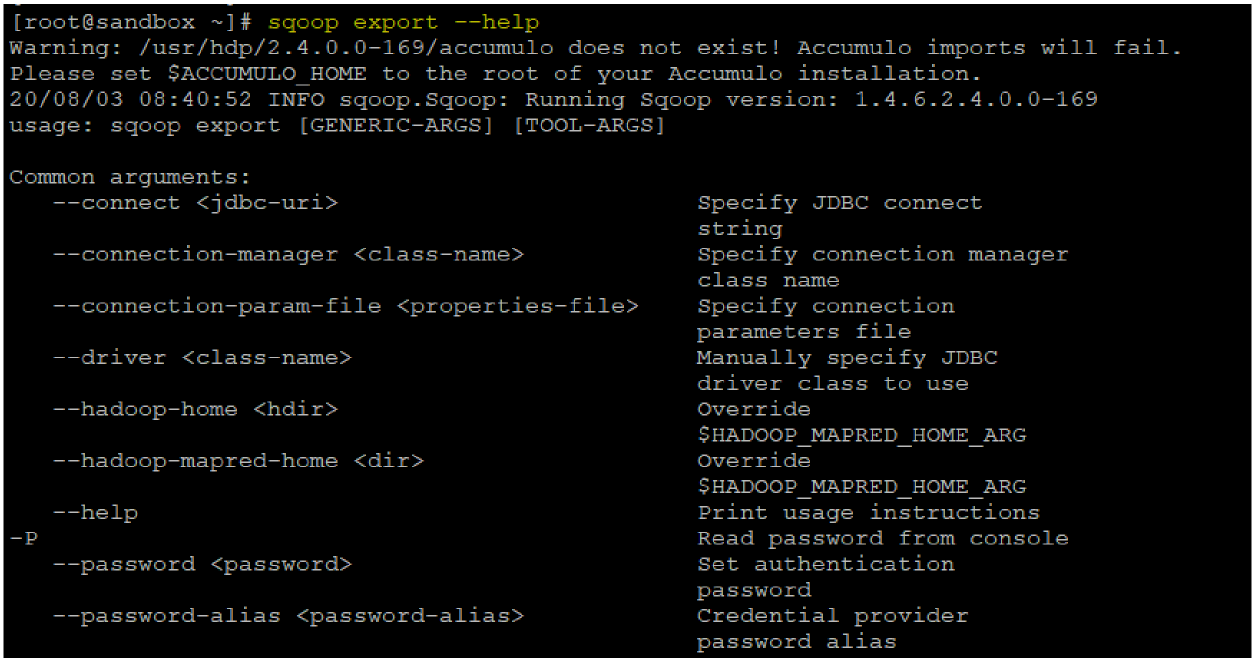


sqoop_table1.name	sqoop_table1.age	sqoop_table1.province
Muneeb	24	Punjab
Bilal	24	Punjab
Nabeel	23	Punjab
Fawad	25	KPK
Ahmed	28	Sindh
Irfan	24	Sindh
Wajahat	26	Balochistan

Sqoop Export:

As discussed above, we can import data from RDBMS to HDFS or RDBMS to hive using Sqoop. Similarly, data can be exported to RDBMS from either HDFS or Hive using ***Sqoop Export***. However, target table must exist in the target database.

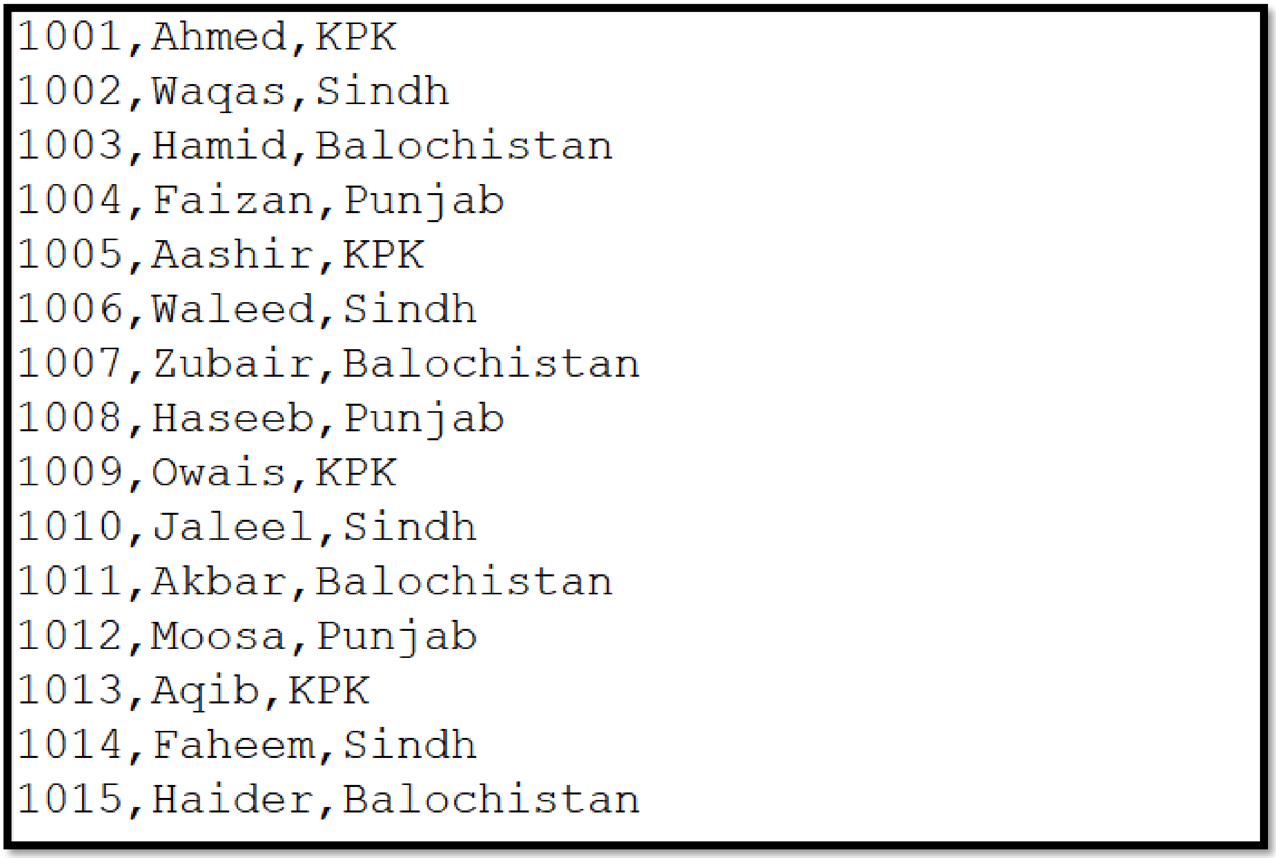
You can see ***export*** command arguments by typing ***sqoop export - -help*** in shell.



```
[root@sandbox ~]# sqoop export --help
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 08:40:52 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
usage: sqoop export [GENERIC-ARGS] [TOOL-ARGS]

Common arguments:
   --connect <jdbc-uri>                         Specify JDBC connect
                                                string
   --connection-manager <class-name>            Specify connection manager
                                                class name
   --connection-param-file <properties-file>    Specify connection
                                                parameters file
   --driver <class-name>                        Manually specify JDBC
                                                driver class to use
   --hadoop-home <hdir>                         Override
                                                $HADOOP_MAPRED_HOME_ARG
   --hadoop-mapred-home <dir>                   Override
                                                $HADOOP_MAPRED_HOME_ARG
   --help                                       Print usage instructions
-P                                              Read password from console
   --password <password>                        Set authentication
                                                password
   --password-alias <password-alias>            Credential provider
                                                password alias
```

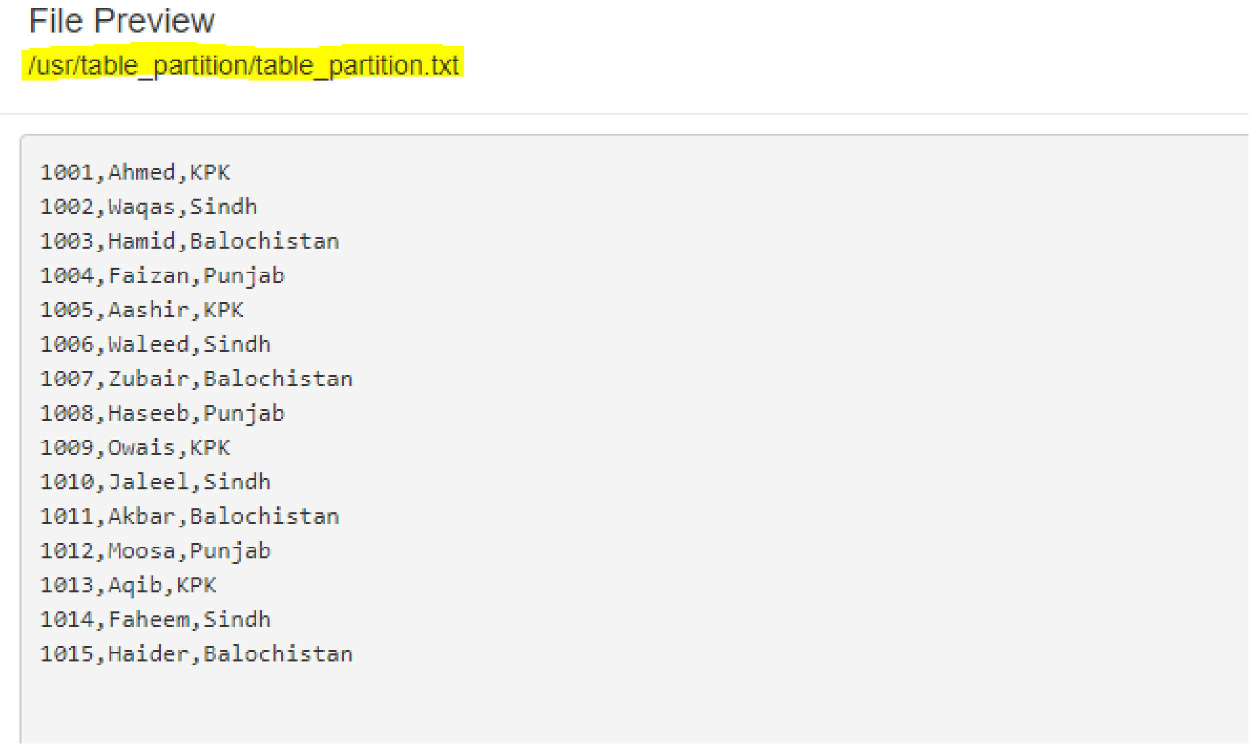
3- Exporting from HDFS:

First, create a .csv file from below given data and upload in a directory in HDFS.



```
1001,Ahmed,KPK
1002,Waqas,Sindh
1003,Hamid,Balochistan
1004,Faizan,Punjab
1005,Aashir,KPK
1006,Waleed,Sindh
1007,Zubair,Balochistan
1008,Haseeb,Punjab
1009,Owais,KPK
1010,Jaleel,Sindh
1011,Akbar,Balochistan
1012,Moosa,Punjab
1013,Aqib,KPK
1014,Faheem,Sindh
1015,Haider,Balochistan
```

e.g. I have uploaded the .csv file in /usr/table_partition/ directory.



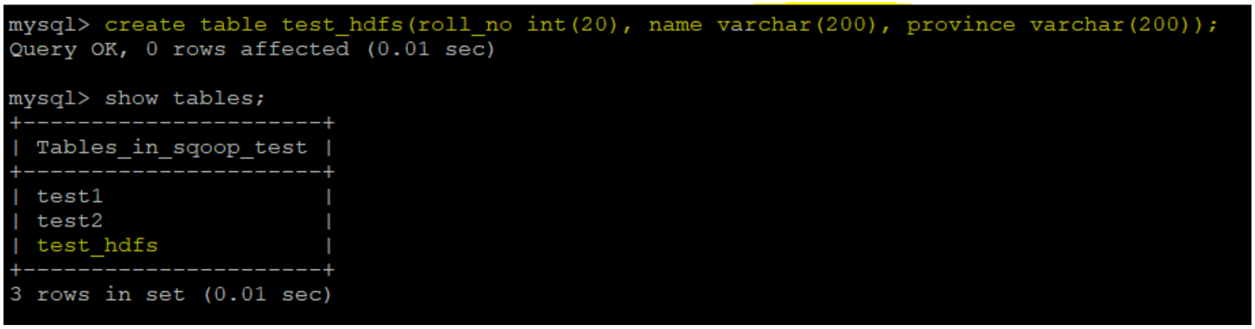
File Preview

/usr/table_partition/table_partition.txt

```
1001,Ahmed,KPK
1002,Waqas,Sindh
1003,Hamid,Balochistan
1004,Faizan,Punjab
1005,Aashir,KPK
1006,Waleed,Sindh
1007,Zubair,Balochistan
1008,Haseeb,Punjab
1009,Owais,KPK
1010,Jaleel,Sindh
1011,Akbar,Balochistan
1012,Moosa,Punjab
1013,Aqib,KPK
1014,Faheem,Sindh
1015,Haider,Balochistan
```

Now, create an empty table in mysql which will be receiving data from HDFS.

create table test_hdfs(roll_no int(20), name varchar(200), province varchar(200))



```
mysql> create table test_hdfs(roll_no int(20), name varchar(200), province varchar(200));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+----------------------+
| Tables_in_sqoop_test |
+----------------------+
| test1                |
| test2                |
| test_hdfs            |
+----------------------+
3 rows in set (0.01 sec)
```

In order to export, run the ***sqoop export*** command in shell as follows:

**sqoop export **

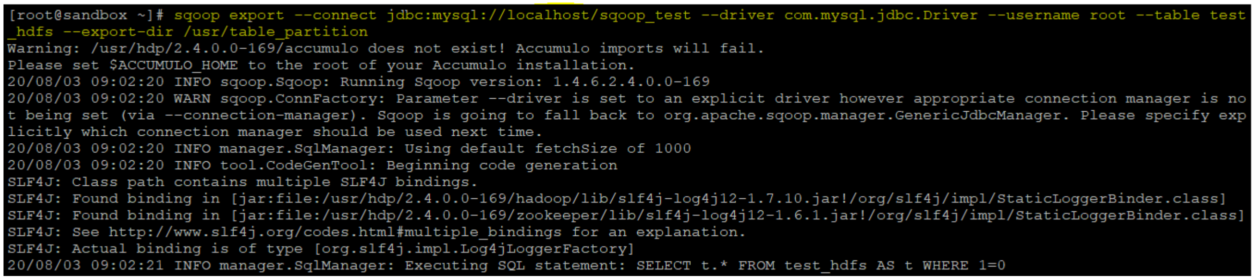
**--connect jdbc:mysql://localhost/test **

**--driver com.mysql.jdbc.Driver **

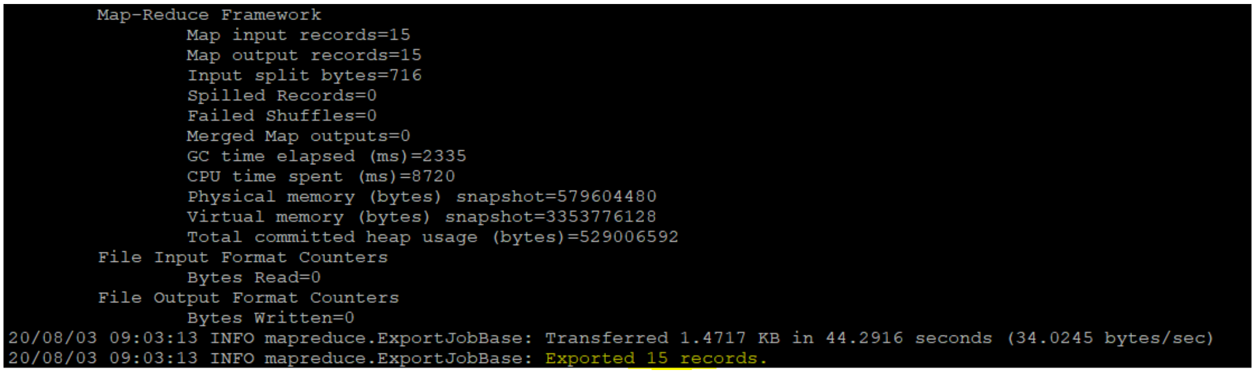
**--username root **

**--table test_hdfs **

--export-dir /usr/table_partition



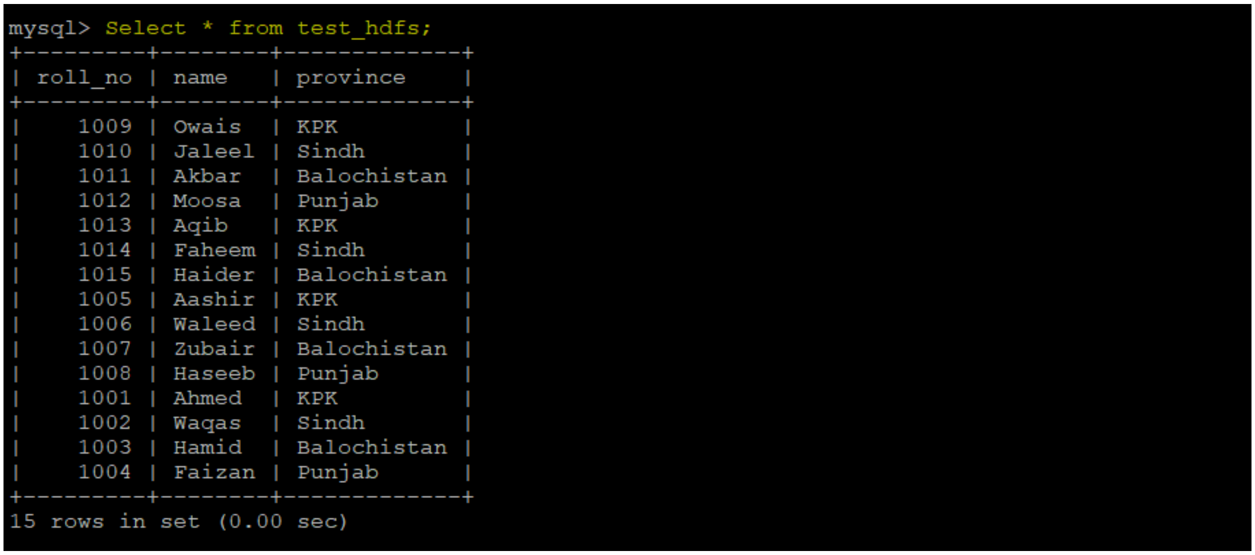
```
[root@sandbox ~]# sqoop export --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --table test
_hdfs --export-dir /usr/table_partition
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 09:02:20 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 09:02:20 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 09:02:20 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 09:02:20 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 09:02:21 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test_hdfs AS t WHERE 1=0
```



```
        Map-Reduce Framework
                Map input records=15
                Map output records=15
                Input split bytes=716
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=2335
                CPU time spent (ms)=8720
                Physical memory (bytes) snapshot=579604480
                Virtual memory (bytes) snapshot=3353776128
                Total committed heap usage (bytes)=529006592
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
20/08/03 09:03:13 INFO mapreduce.ExportJobBase: Transferred 1.4717 KB in 44.2916 seconds (34.0245 bytes/sec)
20/08/03 09:03:13 INFO mapreduce.ExportJobBase: Exported 15 records.
```

In case of successful export, check the **mysql** table to verify the data using a ***SELECT*** command.

SELECT * FROM test_hdfs;



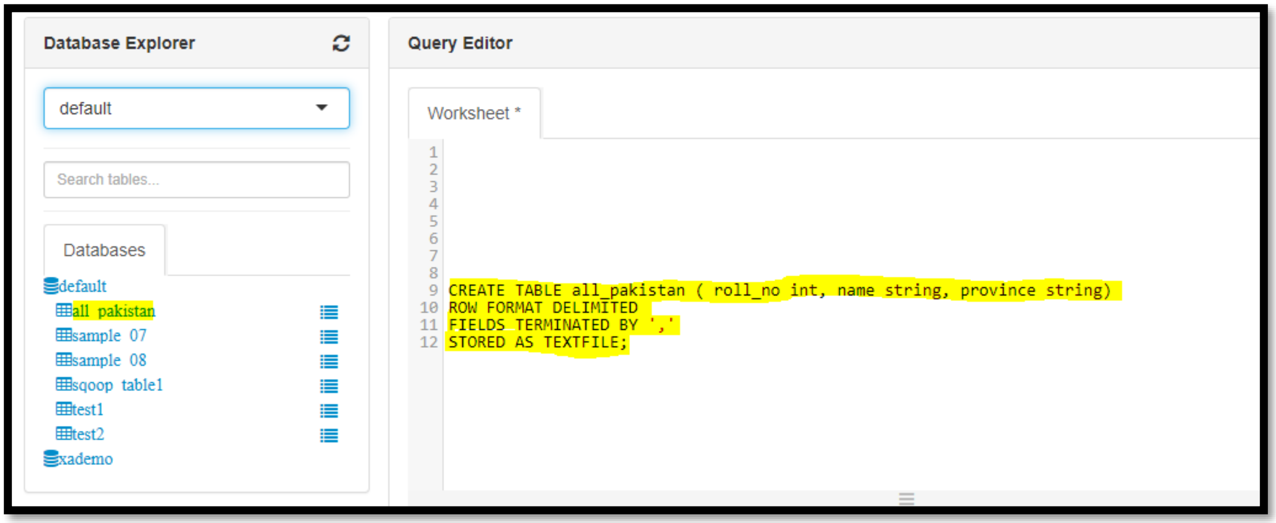
```
mysql> Select * from test_hdfs;
+---------+--------+-------------+
| roll_no | name   | province    |
+---------+--------+-------------+
|    1009 | Owais  | KPK         |
|    1010 | Jaleel | Sindh       |
|    1011 | Akbar  | Balochistan |
|    1012 | Moosa  | Punjab      |
|    1013 | Aqib   | KPK         |
|    1014 | Faheem | Sindh       |
|    1015 | Haider | Balochistan |
|    1005 | Aashir | KPK         |
|    1006 | Waleed | Sindh       |
|    1007 | Zubair | Balochistan |
|    1008 | Haseeb | Punjab      |
|    1001 | Ahmed  | KPK         |
|    1002 | Waqas  | Sindh       |
|    1003 | Hamid  | Balochistan |
|    1004 | Faizan | Punjab      |
+---------+--------+-------------+
15 rows in set (0.00 sec)
```

4- Exporting from Hive:

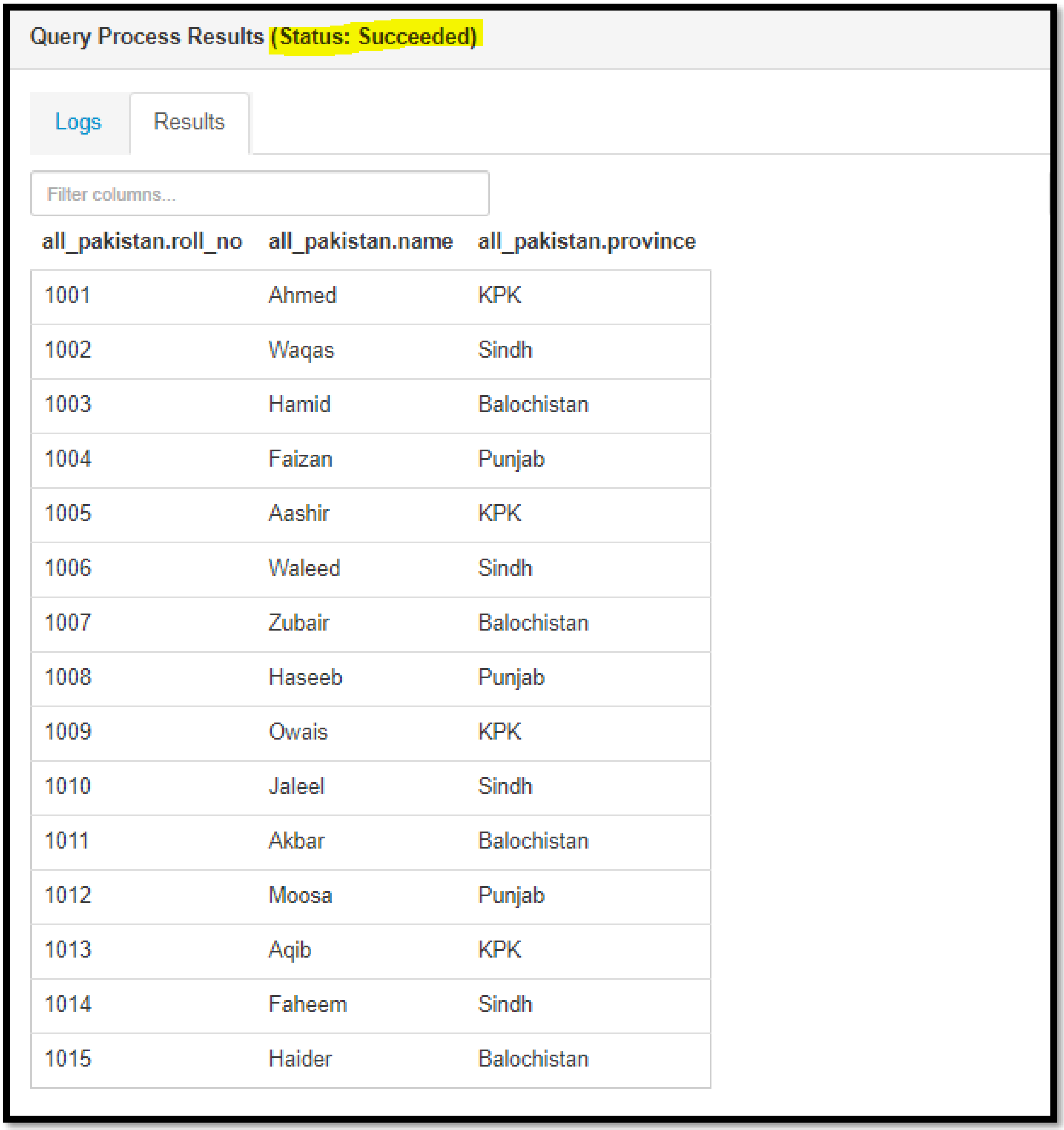
Exporting from hive table is same as exporting from HDFS as hive tables exist on top of directories. However, if you want to export just by listing hive table name, - -***hcatalog*** argument should be invoked.

First, create a new table in hive and upload the .csv given as an example above.

CREATE TABLE all_pakistan (roll_no int, name string, province string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;



```
CREATE TABLE all_pakistan ( roll_no int, name string, province string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```



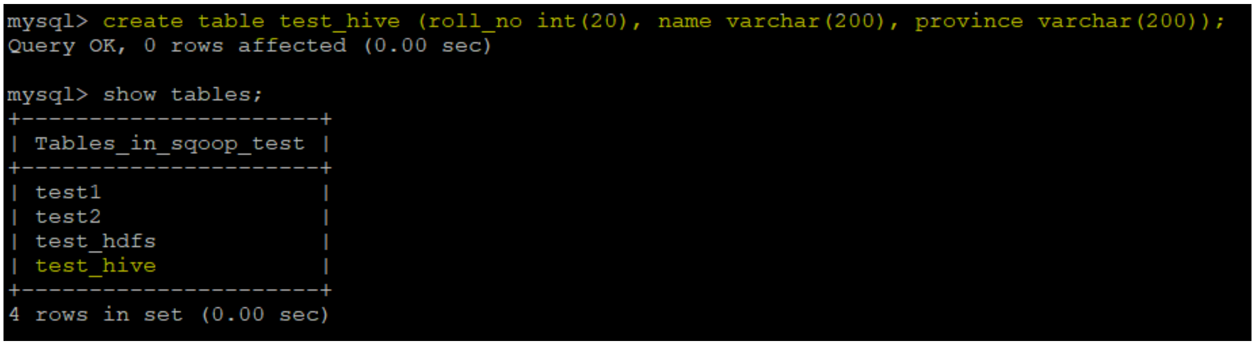
Query Process Results (Status: Succeeded)

all_pakistan.roll_no	all_pakistan.name	all_pakistan.province
1001	Ahmed	KPK
1002	Waqas	Sindh
1003	Hamid	Balochistan
1004	Faizan	Punjab
1005	Aashir	KPK
1006	Waleed	Sindh
1007	Zubair	Balochistan
1008	Haseeb	Punjab
1009	Owais	KPK
1010	Jaleel	Sindh
1011	Akbar	Balochistan
1012	Moosa	Punjab
1013	Aqib	KPK
1014	Faheem	Sindh
1015	Haider	Balochistan

In order to export to mysql, create a similar table in mysql using following command.

- **create table hive_test (roll_no int(20), name varchar(200), province varchar(200));**

Note: Column names should be same in both tables:



```
mysql> create table test_hive (roll_no int(20), name varchar(200), province varchar(200));
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+----------------------+
| Tables_in_sqoop_test |
+----------------------+
| test1                |
| test2                |
| test_hdfs            |
| test_hive            |
+----------------------+
4 rows in set (0.00 sec)
```

A hive table in ***default*** database can be exported to mysql by following command.

Note: Use ***- -hcatalog-database <database name>*** if desired table is in any other database.

sqoop export \

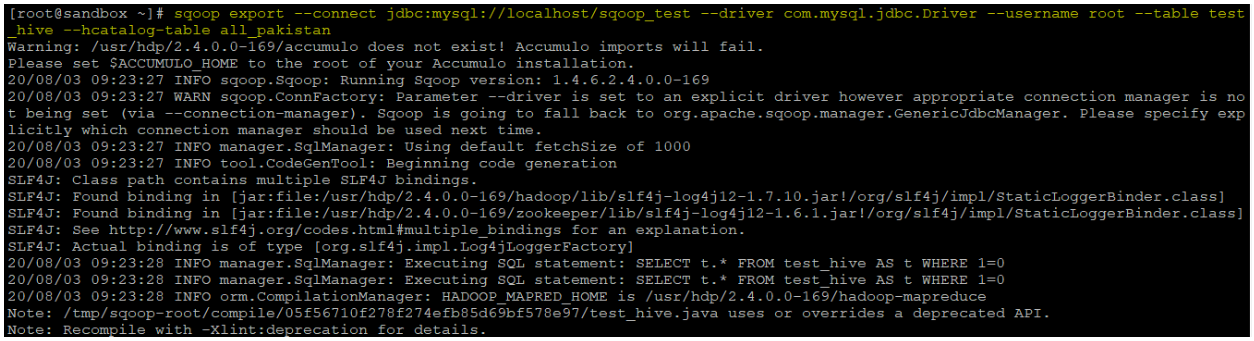
--connect jdbc:mysql://localhost/test \

--driver com.mysql.jdbc.Driver \

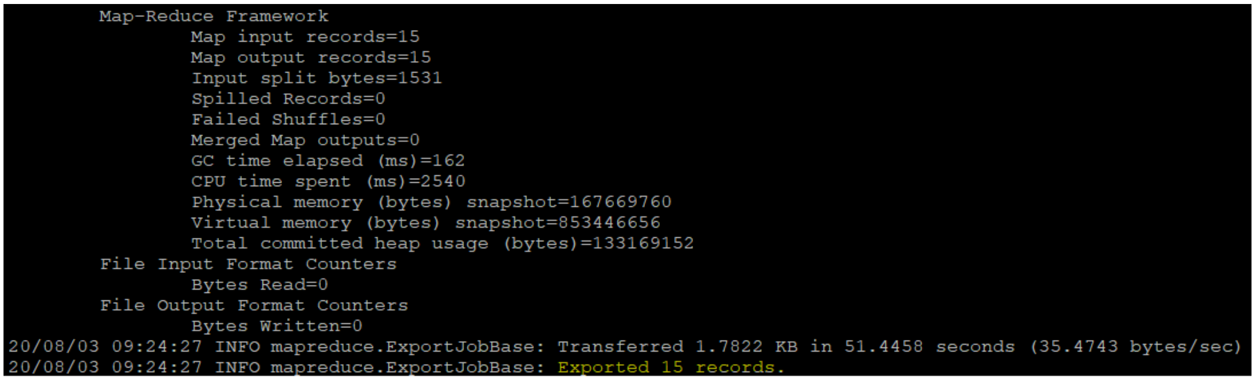
--username root \

--table hive_test \

--hcatalog-table all_pakistan



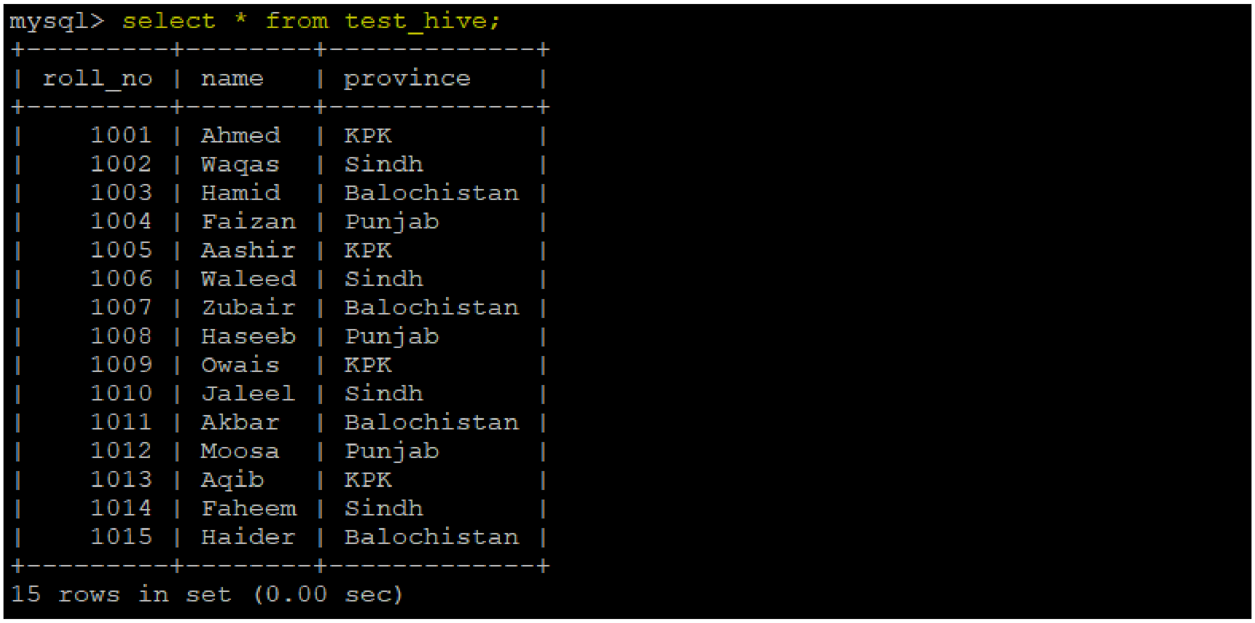
```
[root@sandbox ~]# sqoop export --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --table test
_hive --hcatalog-table all_pakistan
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 09:23:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 09:23:27 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 09:23:27 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 09:23:27 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 09:23:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test_hive AS t WHERE 1=0
20/08/03 09:23:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test_hive AS t WHERE 1=0
20/08/03 09:23:28 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.4.0.0-169/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/05f56710f278f274efb85d69bf578e97/test_hive.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```



```
        Map-Reduce Framework
                Map input records=15
                Map output records=15
                Input split bytes=1531
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=162
                CPU time spent (ms)=2540
                Physical memory (bytes) snapshot=167669760
                Virtual memory (bytes) snapshot=853446656
                Total committed heap usage (bytes)=133169152
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
20/08/03 09:24:27 INFO mapreduce.ExportJobBase: Transferred 1.7822 KB in 51.4458 seconds (35.4743 bytes/sec)
20/08/03 09:24:27 INFO mapreduce.ExportJobBase: Exported 15 records.
```

After successful export, check the mysql table to verify the data using a ***SELECT*** command.

SELECT * FROM test_hive;

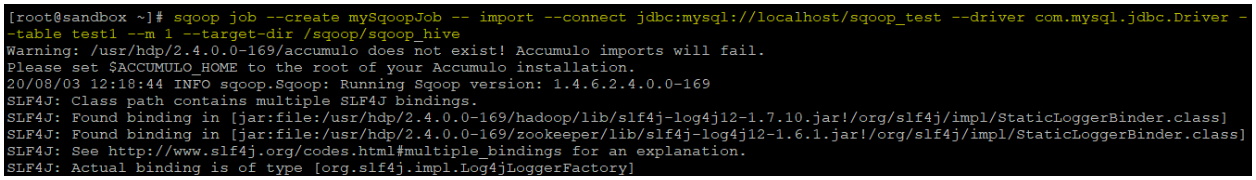


```
mysql> select * from test_hive;
+---------+--------+-------------+
| roll_no | name   | province    |
+---------+--------+-------------+
|    1001 | Ahmed  | KPK         |
|    1002 | Waqas  | Sindh       |
|    1003 | Hamid  | Balochistan |
|    1004 | Faizan | Punjab      |
|    1005 | Aashir | KPK         |
|    1006 | Waleed | Sindh       |
|    1007 | Zubair | Balochistan |
|    1008 | Haseeb | Punjab      |
|    1009 | Owais  | KPK         |
|    1010 | Jaleel | Sindh       |
|    1011 | Akbar  | Balochistan |
|    1012 | Moosa  | Punjab      |
|    1013 | Aqib   | KPK         |
|    1014 | Faheem | Sindh       |
|    1015 | Haider | Balochistan |
+---------+--------+-------------+
15 rows in set (0.00 sec)
```

Sqoop Jobs:

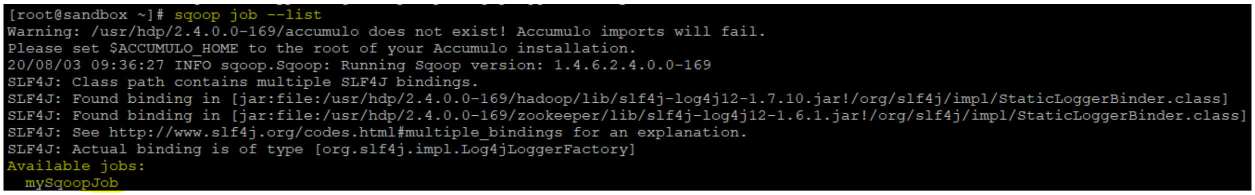
We can also create and maintain jobs. Sqoop job creates and saves the import and export commands. It specifies parameters to identify and recall the saved job. This re-calling or re-executing is used in the incremental import, which can import the updated rows from RDBMS table to HDFS. For that we can use **–create clause.**

```
sqoop job \
--create mySqoopJob \
-- import \
--connect jdbc:mysql://localhost/db  \
--driver com.mysql.jdbc.Driver \
--table test1 \
 --m 1 \
--target-dir /sqoop/sqoop_hive/  \
```



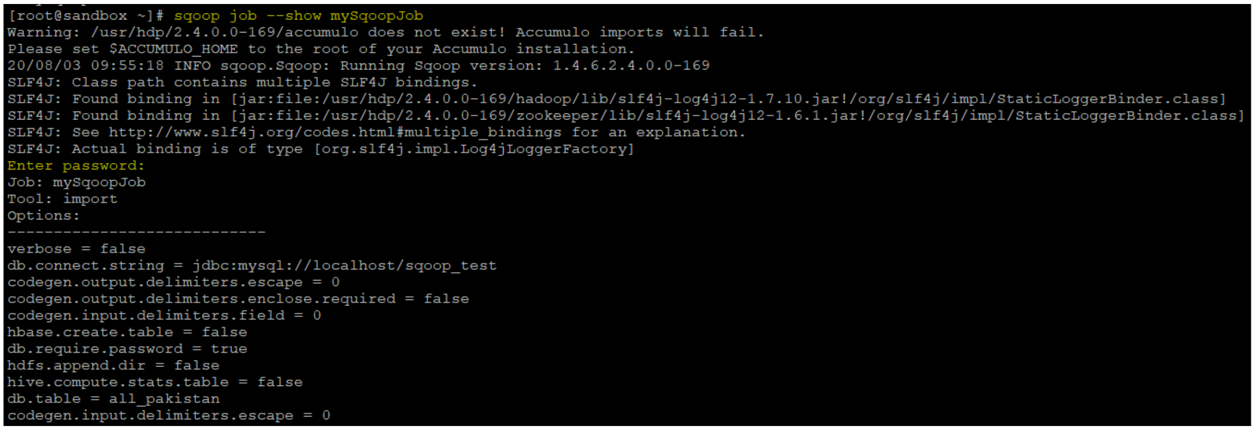
```
[root@sandbox ~]# sqoop job --create mySqoopJob -- import --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver -
-table test1 --m 1 --target-dir /sqoop/sqoop_hive
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:18:44 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
```

We can also list down the sqoop jobs by using **–list clause.**
--delete is used to delete a sqoop job



```
[root@sandbox ~]# sqoop job --list
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 09:36:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Available jobs:
  mySqoopJob
```

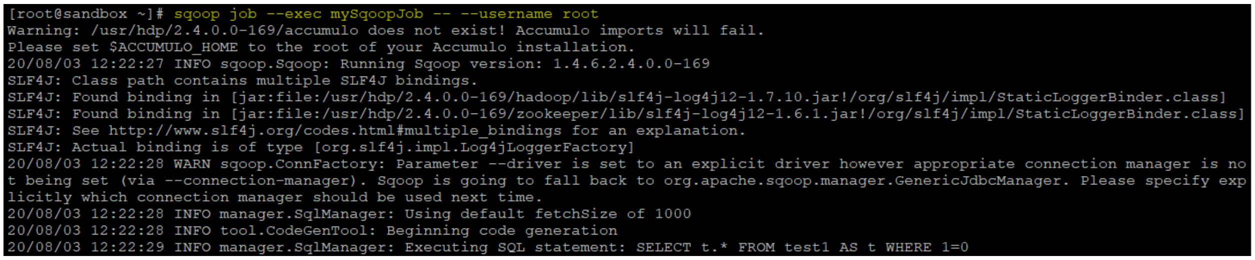
--show is used to view details of a specific job



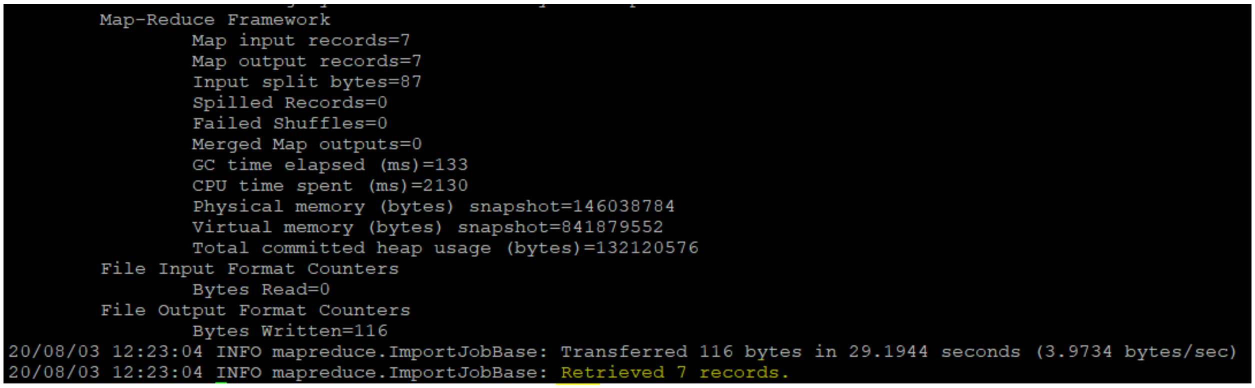
```
[root@sandbox ~]# sqoop job --show mySqoopJob
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 09:55:18 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Enter password:
Job: mySqoopJob
Tool: import
Options:
----------------------------
verbose = false
db.connect.string = jdbc:mysql://localhost/sqoop_test
codegen.output.delimiters.escape = 0
codegen.output.delimiters.enclose.required = false
codegen.input.delimiters.field = 0
hbase.create.table = false
db.require.password = true
hdfs.append.dir = false
hive.compute.stats.table = false
db.table = all_pakistan
codegen.input.delimiters.escape = 0
```

--exec is used to execute a particular job

- **Sqoop job –exec mySqoopJob -- --username root**

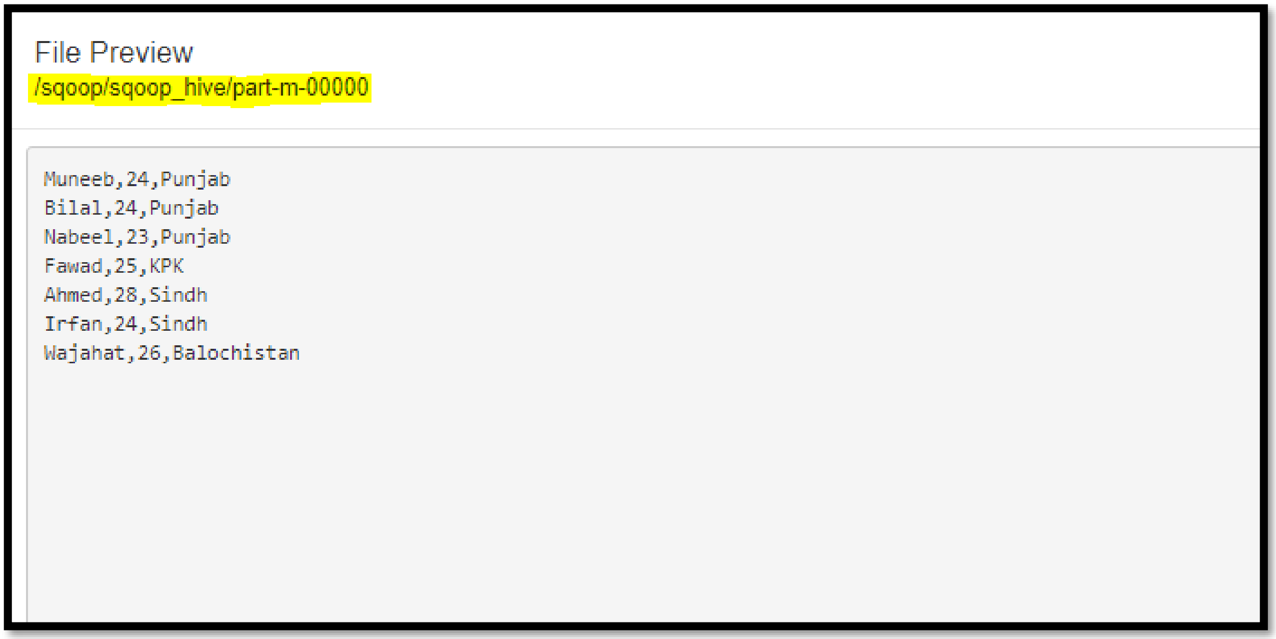


```
[root@sandbox ~]# sqoop job --exec mySqoopJob -- --username root
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:22:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 12:22:28 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 12:22:28 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 12:22:28 INFO tool.CodeGenTool: Beginning code generation
20/08/03 12:22:29 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
```



```
        Map-Reduce Framework
                Map input records=7
                Map output records=7
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=133
                CPU time spent (ms)=2130
                Physical memory (bytes) snapshot=146038784
                Virtual memory (bytes) snapshot=841879552
                Total committed heap usage (bytes)=132120576
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=116
20/08/03 12:23:04 INFO mapreduce.ImportJobBase: Transferred 116 bytes in 29.1944 seconds (3.9734 bytes/sec)
20/08/03 12:23:04 INFO mapreduce.ImportJobBase: Retrieved 7 records.
```

Now let's validate the target directory:



File Preview

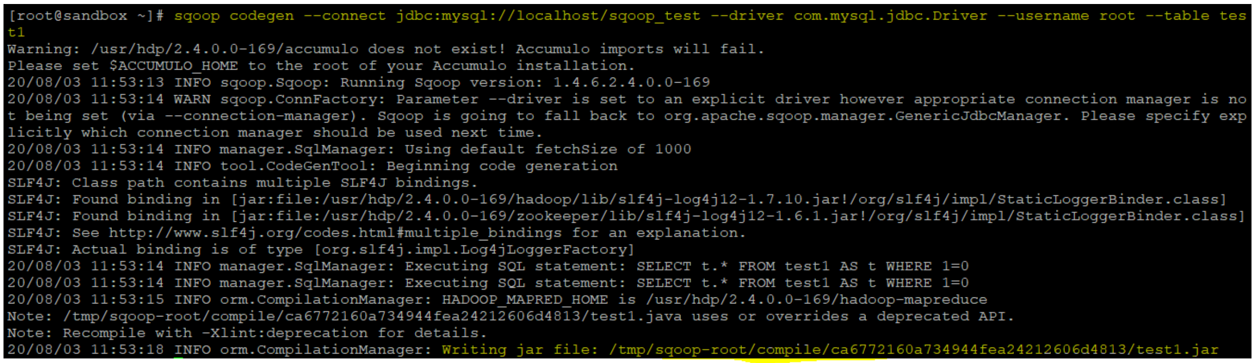
/sqoop/sqoop_hive/part-m-00000

```
Muneeb,24,Punjab
Bilal,24,Punjab
Nabeel,23,Punjab
Fawad,25,KPK
Ahmed,28,Sindh
Irfan,24,Sindh
Wajahat,26,Balochistan
```

Sqoop Code-gen:

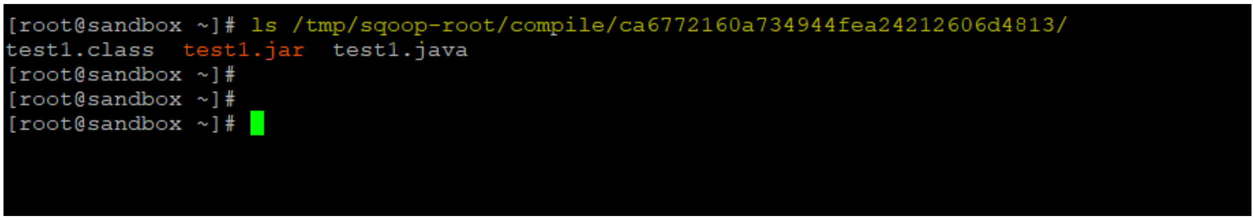
The codegen tool generates Java classes which encapsulate and interpret imported records. The Java definition of a record is instantiated as part of the import process, but can also be performed separately. For example, if Java source is lost, it can be recreated. New versions of a class can be created which use different delimiters between fields, and so on.

```
sqoop codegen \
--connect jdbc:mysql://localhost/sqoop_test \
--driver com.mysql.jdbc.Driver \
--username root \
--table test1 \
```



```
[root@sandbox ~]# sqoop codegen --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --table tes
t1
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 11:53:13 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 11:53:14 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 11:53:14 INFO manager.SqlManager: Using default fetchSize of 1000
20/08/03 11:53:14 INFO tool.CodeGenTool: Beginning code generation
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 11:53:14 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/03 11:53:14 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM test1 AS t WHERE 1=0
20/08/03 11:53:15 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.4.0.0-169/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/ca6772160a734944fea24212606d4813/test1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/08/03 11:53:18 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/ca6772160a734944fea24212606d4813/test1.jar
```

Let validate the files at mentioned location

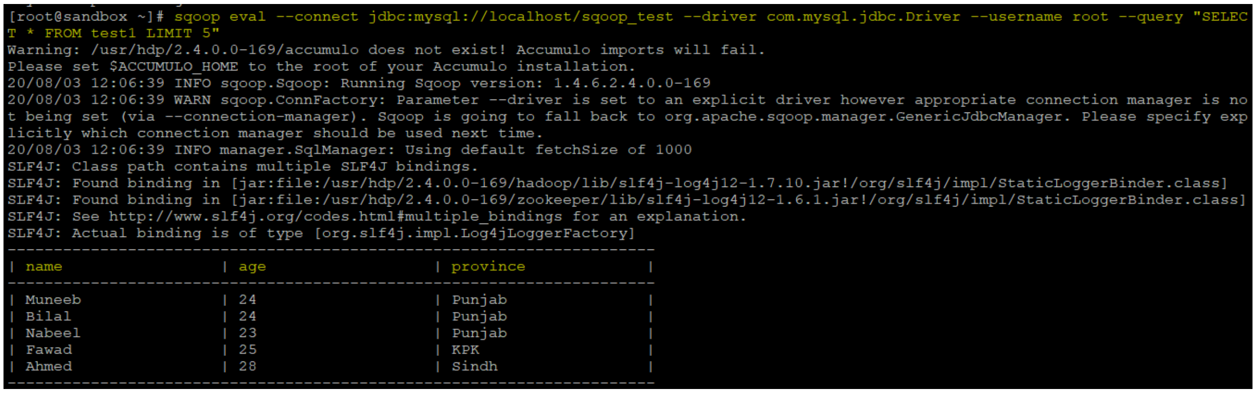


```
[root@sandbox ~]# ls /tmp/sqoop-root/compile/ca6772160a734944fea24212606d4813/
test1.class  test1.jar  test1.java
[root@sandbox ~]#
[root@sandbox ~]#
[root@sandbox ~]#
```

Sqoop Eval:

It allows users to execute user-defined queries against respective database servers and preview the result in the console. So, the user can expect the resultant table data to import. Using eval, we can evaluate any type of SQL query that can be either DDL or DML statement.

sqoop eval \
--connect jdbc:mysql://localhost/sqoop_test \
--driver com.mysql.jdbc.Driver \
--username root \
--query “SELECT * FROM test1 LIMIT 5” \



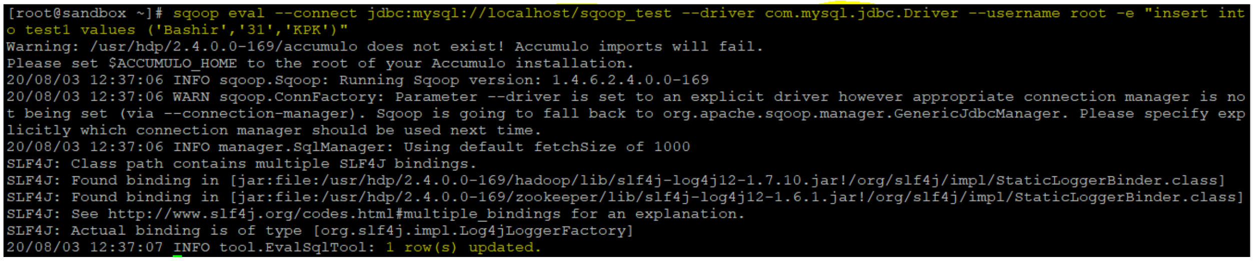
```
[root@sandbox ~]# sqoop eval --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root --query "SELEC
T * FROM test1 LIMIT 5"
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:06:39 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 12:06:39 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 12:06:39 INFO manager.SqlManager: Using default fetchSize of 1000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
-----------------------------------------------------------------------
| name                 | age                  | province             |
-----------------------------------------------------------------------
| Muneeb               | 24                   | Punjab               |
| Bilal                | 24                   | Punjab               |
| Nabeel               | 23                   | Punjab               |
| Fawad                | 25                   | KPK                  |
| Ahmed                | 28                   | Sindh                |
-----------------------------------------------------------------------
```

Warning

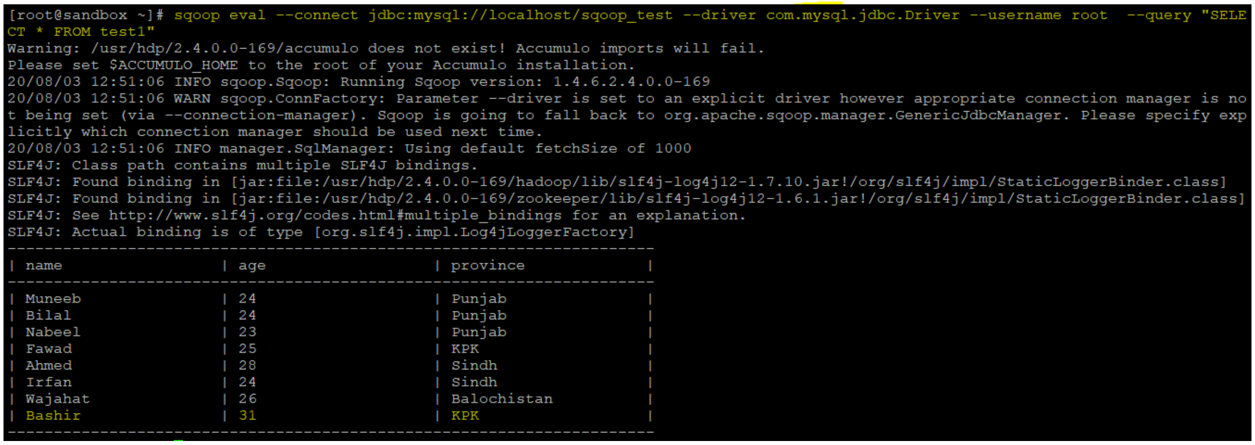
The eval tool is provided for evaluation purpose only. You can use it to verify database connection from within the Sqoop or to test simple queries. It’s not supposed to be used in production workflows.

We can also insert data from **EVAL** using **-e** keyword

sqoop eval \
--connect jdbc:mysql://localhost/sqoop_test \
--driver com.mysql.jdbc.Driver \
--username root \
-e "insert into test1 values ('Bashir','31','KPK')" \

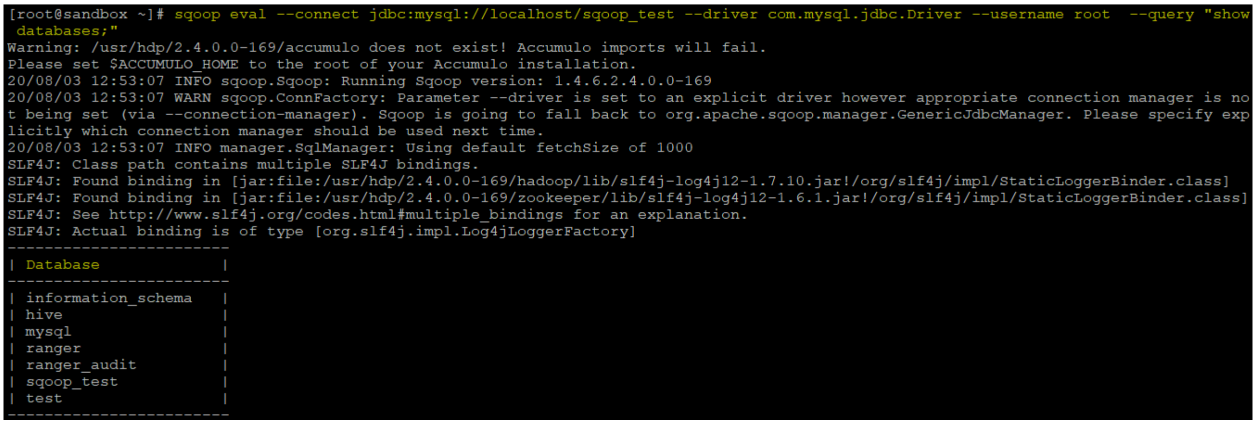


```
[root@sandbox ~]# sqoop eval --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root -e "insert int
o test1 values ('Bashir','31','KPK')"
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:37:06 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 12:37:06 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 12:37:06 INFO manager.SqlManager: Using default fetchSize of 1000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/08/03 12:37:07 INFO tool.EvalSqlTool: 1 row(s) updated.
```



```
[root@sandbox ~]# sqoop eval --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root  --query "SELE
CT * FROM test1"
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:51:06 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 12:51:06 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 12:51:06 INFO manager.SqlManager: Using default fetchSize of 1000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
-----------------------------------------------------------------------
| name                 | age                  | province               |
-----------------------------------------------------------------------
| Muneeb               | 24                   | Punjab                 |
| Bilal                | 24                   | Punjab                 |
| Nabeel               | 23                   | Punjab                 |
| Fawad                | 25                   | KPK                    |
| Ahmed                | 28                   | Sindh                  |
| Irfan                | 24                   | Sindh                  |
| Wajahat              | 26                   | Balochistan            |
| Bashir               | 31                   | KPK                    |
-----------------------------------------------------------------------
```

- Show databases using query eval



```
[root@sandbox ~]# sqoop eval --connect jdbc:mysql://localhost/sqoop_test --driver com.mysql.jdbc.Driver --username root  --query "show
 databases;"
Warning: /usr/hdp/2.4.0.0-169/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/08/03 12:53:07 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.4.0.0-169
20/08/03 12:53:07 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is no
t being set (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify exp
licitly which connection manager should be used next time.
20/08/03 12:53:07 INFO manager.SqlManager: Using default fetchSize of 1000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/2.4.0.0-169/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
------------------------
| Database             |
------------------------
| information_schema   |
| hive                 |
| mysql                |
| ranger               |
| ranger_audit         |
| sqoop_test           |
| test                 |
------------------------
```