ASSIGNMENT 11.2

Problem Statement:

Perform incremental load in Hive. Read from MySQL Table and load it in Hive table. Create hive table if it does not exist. If it exists, perform the incremental load.

Solution:

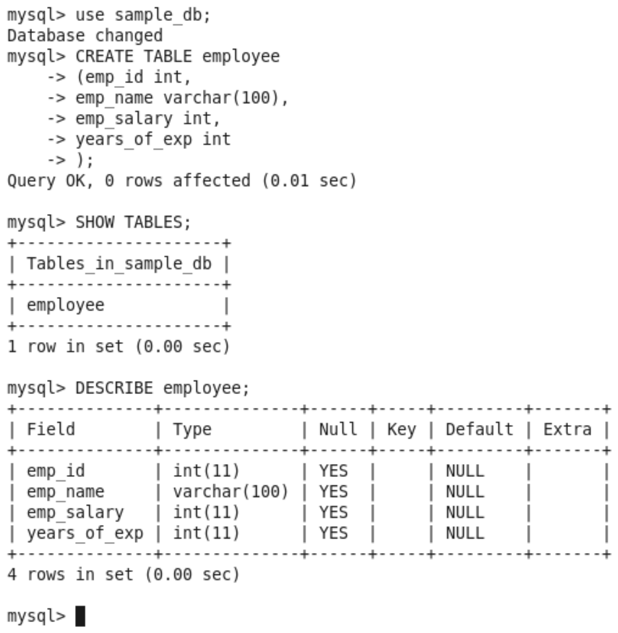
Incremental load from MySQL to Hive:

Step 1: Create a table in MySQL and insert few records into it.

We will create a table 'employee' that stores employee's id, name and years of experience. Here is the CREATE statement for the same:

```
mysql> CREATE TABLE employee
(emp_id int,
emp_name varchar(100),
emp_salary int,
years_of_exp int
);
```

We are creating this table inside a database with name 'sample_db'.



```
mysql> use sample_db;
Database changed
mysql> CREATE TABLE employee
    -> (emp_id int,
    -> emp_name varchar(100),
    -> emp_salary int,
    -> years_of_exp int
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+---------------------+
| Tables_in_sample_db |
+---------------------+
| employee            |
+---------------------+
1 row in set (0.00 sec)

mysql> DESCRIBE employee;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| emp_id       | int(11)      | YES  |     | NULL    |       |
| emp_name     | varchar(100) | YES  |     | NULL    |       |
| emp_salary   | int(11)      | YES  |     | NULL    |       |
| years_of_exp | int(11)      | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> █
```

Let’s insert some data into this table via INSERT command:

mysql> INSERT INTO employee VALUES(101,'Amitabh',20000,1);

mysql> INSERT INTO employee VALUES(102,'Shahrukh',10000,2);

mysql> INSERT INTO employee VALUES(103,'Akshay',11000,3);

mysql> INSERT INTO employee VALUES(104,'Anubhav',5000,4);

mysql> INSERT INTO employee VALUES(105,'Pawan',2500,5);

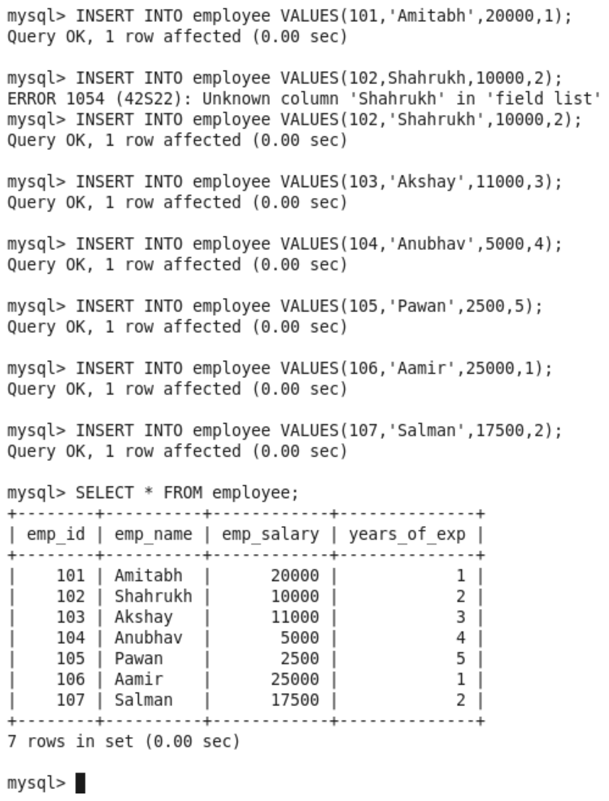
mysql> INSERT INTO employee VALUES(106,'Aamir',25000,1);

mysql> INSERT INTO employee VALUES(107,'Salman',17500,2);

We have inserted seven records with employee id in sequential manner starting from value ‘101’.

We can see the result by using SELECT command on this table:

SELECT * FROM employee;

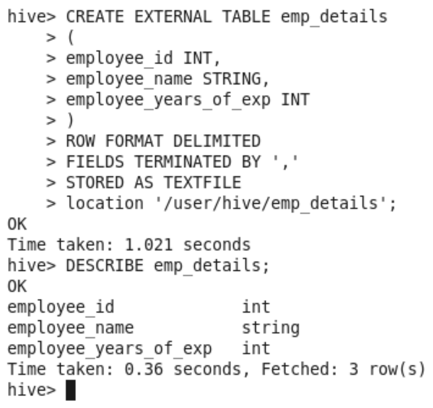


```
mysql> INSERT INTO employee VALUES(101,'Amitabh',20000,1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(102,Shahrukh,10000,2);
ERROR 1054 (42S22): Unknown column 'Shahrukh' in 'field list'
mysql> INSERT INTO employee VALUES(102,'Shahrukh',10000,2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(103,'Akshay',11000,3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(104,'Anubhav',5000,4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(105,'Pawan',2500,5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(106,'Aamir',25000,1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(107,'Salman',17500,2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM employee;
+--------+----------+------------+--------------+
| emp_id | emp_name | emp_salary | years_of_exp |
+--------+----------+------------+--------------+
|    101 | Amitabh  |      20000 |            1 |
|    102 | Shahrukh |      10000 |            2 |
|    103 | Akshay   |      11000 |            3 |
|    104 | Anubhav  |       5000 |            4 |
|    105 | Pawan    |       2500 |            5 |
|    106 | Aamir    |      25000 |            1 |
|    107 | Salman   |      17500 |            2 |
+--------+----------+------------+--------------+
7 rows in set (0.00 sec)

mysql> █
```

Step 2: Create an external table in Hive with same column structure as the one in MySQL.

Here is the CREATE statement used to create table in Hive:

```
mysql> CREATE EXTERNAL TABLE emp_details
(
employee_id INT,
employee_name STRING,
employee_years_of_exp INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/hive/emp_details';
```



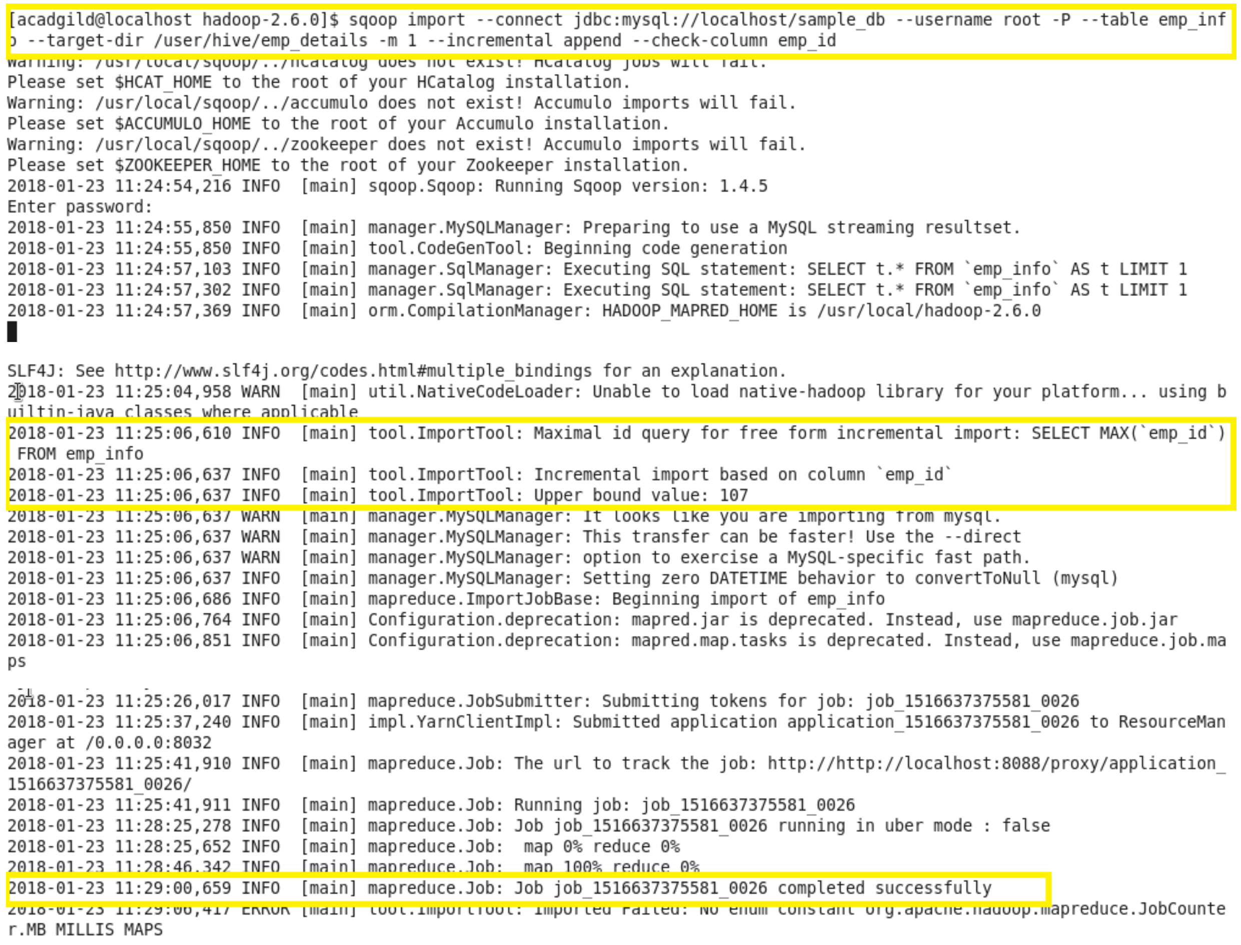
```
hive> CREATE EXTERNAL TABLE emp_details
    > (
    > employee_id INT,
    > employee_name STRING,
    > employee_years_of_exp INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > location '/user/hive/emp_details';
OK
Time taken: 1.021 seconds
hive> DESCRIBE emp_details;
OK
employee_id             int
employee_name           string
employee_years_of_exp   int
Time taken: 0.36 seconds, Fetched: 3 row(s)
hive> 
```

The reason behind specifying the location in the above command is that the table will be ingested with data files put in this path. So while importing data from any other data sources we just need to specify this location and table in Hive will automatically get data from files stored in that path.

Step 3: Use Sqoop import command to perform initial data load from MySQL table into Hive. Here is the command to achieve this:

```
$ sqoop import --connect jdbc:mysql://localhost/sample_db --username root -P --table emp_info
--target-dir /user/hive/emp_details -m 1 --incremental append --check-column emp_id
```

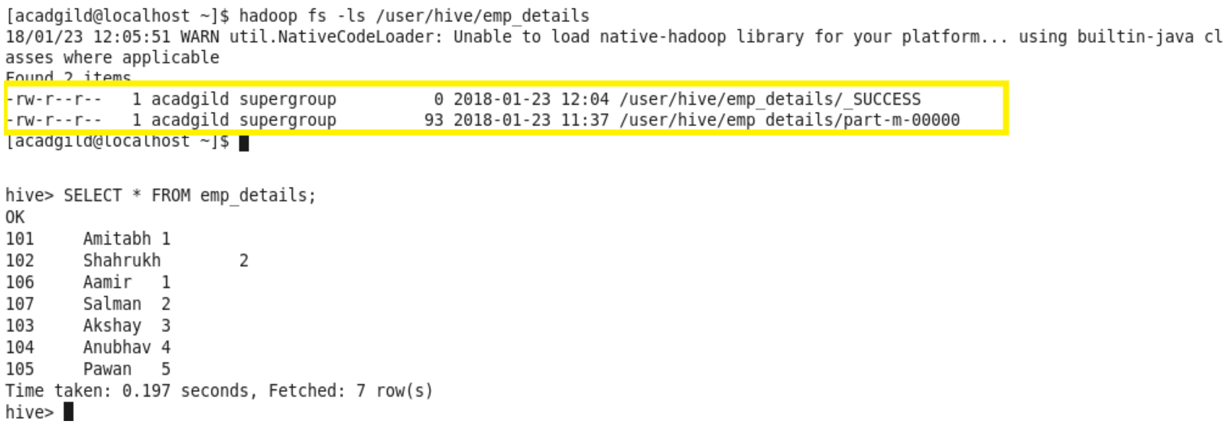
In the above command the target directory we specified is same as the one we used while creating Hive table. There is another important parameter, **incremental append** to instruct Sqoop that we would like to get data in Hive table as and when new data gets inserted into MySQL table. This depends on the values of the column we specify for **check-column** parameter, in this case we are using employee id which will be given in incremental fashion.



```
[acadgild@localhost hadoop-2.6.0]$ sqoop import --connect jdbc:mysql://localhost/sample_db --username root -P --table emp_inf
o --target-dir /user/hive/emp_details -m 1 --incremental append --check-column emp_id
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 11:24:54,216 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 11:24:55,850 INFO  [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 11:24:55,850 INFO  [main] tool.CodeGenTool: Beginning code generation
2018-01-23 11:24:57,103 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 11:24:57,302 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 11:24:57,369 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
█

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2018-01-23 11:25:04,958 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
2018-01-23 11:25:06,610 INFO  [main] tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`emp_id`)
 FROM emp_info
2018-01-23 11:25:06,637 INFO  [main] tool.ImportTool: Incremental import based on column `emp_id`
2018-01-23 11:25:06,637 INFO  [main] tool.ImportTool: Upper bound value: 107
2018-01-23 11:25:06,637 WARN  [main] manager.MySQLManager: It looks like you are importing from mysql.
2018-01-23 11:25:06,637 WARN  [main] manager.MySQLManager: This transfer can be faster! Use the --direct
2018-01-23 11:25:06,637 WARN  [main] manager.MySQLManager: option to exercise a MySQL-specific fast path.
2018-01-23 11:25:06,637 INFO  [main] manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
2018-01-23 11:25:06,686 INFO  [main] mapreduce.ImportJobBase: Beginning import of emp_info
2018-01-23 11:25:06,764 INFO  [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2018-01-23 11:25:06,851 INFO  [main] Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.ma
ps

2018-01-23 11:25:26,017 INFO  [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1516637375581_0026
2018-01-23 11:25:37,240 INFO  [main] impl.YarnClientImpl: Submitted application application_1516637375581_0026 to ResourceMan
ager at /0.0.0.0:8032
2018-01-23 11:25:41,910 INFO  [main] mapreduce.Job: The url to track the job: http://http://localhost:8088/proxy/application_
1516637375581_0026/
2018-01-23 11:25:41,911 INFO  [main] mapreduce.Job: Running job: job_1516637375581_0026
2018-01-23 11:28:25,278 INFO  [main] mapreduce.Job: Job job_1516637375581_0026 running in uber mode : false
2018-01-23 11:28:25,652 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2018-01-23 11:28:46,342 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2018-01-23 11:29:00,659 INFO  [main] mapreduce.Job: Job job_1516637375581_0026 completed successfully
2018-01-23 11:29:06,417 ERROR [main] tool.ImportTool: Imported Failed: No enum constant org.apache.hadoop.mapreduce.JobCounte
r.MB MILLIS MAPS
```

The ‘emp_info’ table in MySQL has seven records with employee id ranging from 101 to 107. We can the query applied internally highlighted in the above screenshot: **SELECT MAX(`emp_id`) FROM emp_info**. The upper bound value is has got is 107. These records will be inserted into a file and stored the target directory location given in the import command.

Let’s check the contents of target directory and then the corresponding table in Hive.



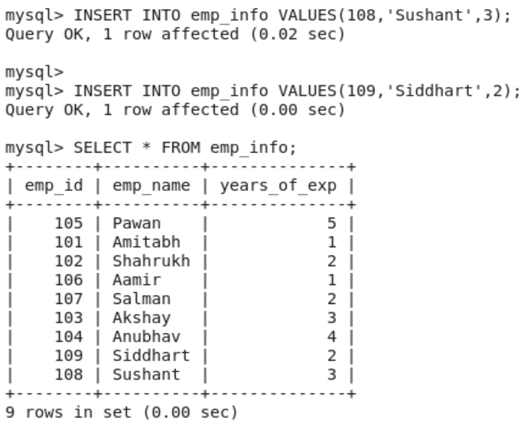
```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/emp_details
18/01/23 12:05:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-01-23 12:04 /user/hive/emp_details/_SUCCESS
-rw-r--r--   1 acadgild supergroup         93 2018-01-23 11:37 /user/hive/emp_details/part-m-00000
[acadgild@localhost ~]$

hive> SELECT * FROM emp_details;
OK
101     Amitabh 1
102     Shahrukh        2
106     Aamir   1
107     Salman  2
103     Akshay  3
104     Anubhav 4
105     Pawan   5
Time taken: 0.197 seconds, Fetched: 7 row(s)
hive>
```

Step 4: Insert few more records into the MySQL table

Let's insert a couple of new records into the table in MySQL 'emp_info' via INSERT command:

INSERT INTO emp_info VALUES(108,'Sushant',3);

INSERT INTO emp_info VALUES(109,'Siddhart',2);



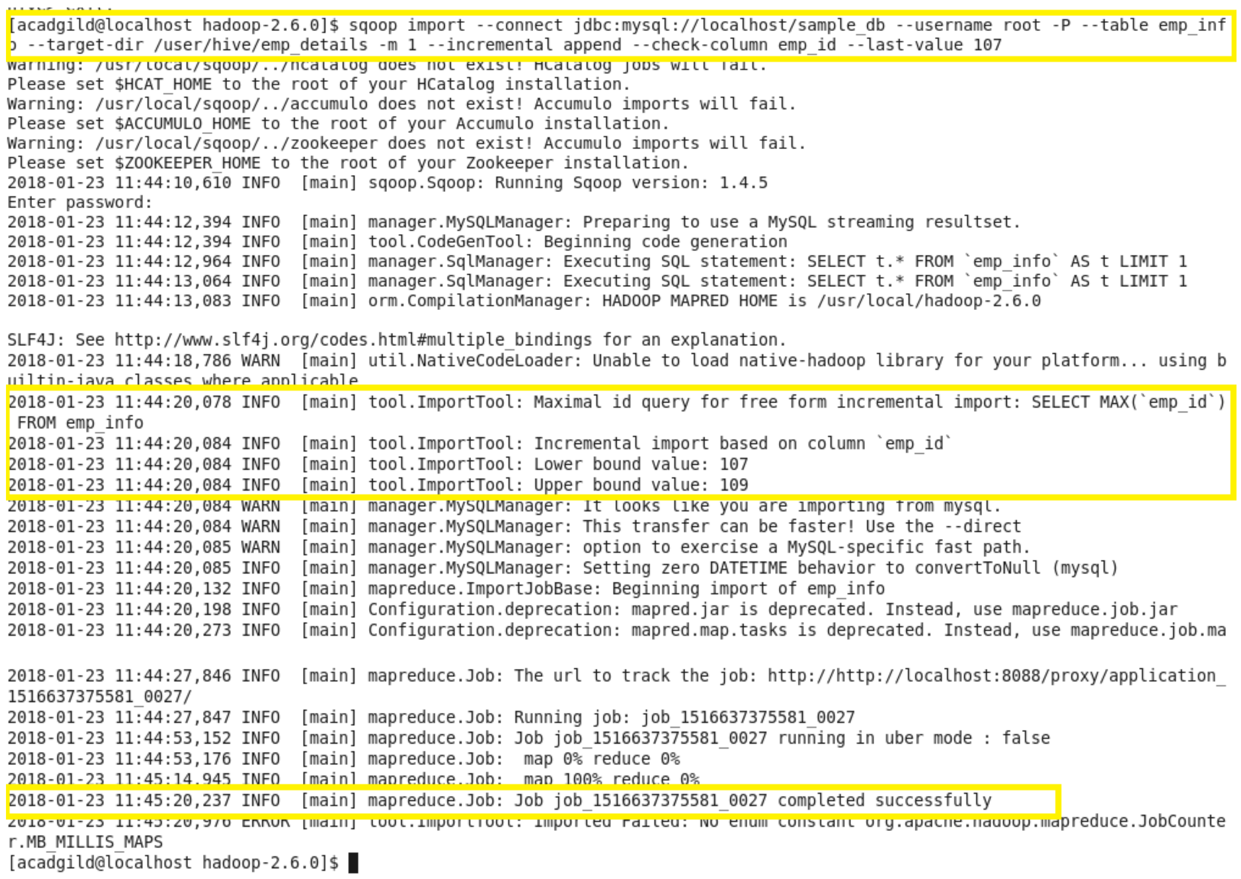
```
mysql> INSERT INTO emp_info VALUES(108,'Sushant',3);
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> INSERT INTO emp_info VALUES(109,'Siddhart',2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM emp_info;
+--------+----------+--------------+
| emp_id | emp_name | years_of_exp |
+--------+----------+--------------+
|    105 | Pawan    |            5 |
|    101 | Amitabh  |            1 |
|    102 | Shahrukh |            2 |
|    106 | Aamir    |            1 |
|    107 | Salman   |            2 |
|    103 | Akshay   |            3 |
|    104 | Anubhav  |            4 |
|    109 | Siddhart |            2 |
|    108 | Sushant  |            3 |
+--------+----------+--------------+
9 rows in set (0.00 sec)
```

Step 5: Use Sqoop incremental append command to add these records in the Hive table.

Here is the command we can use to do the same:

$ sqoop import --connect jdbc:mysql://localhost/sample_db --username root -P --table emp_info --target-dir /user/hive/emp_details -m 1 --incremental append --check-column emp_id --last-value 107

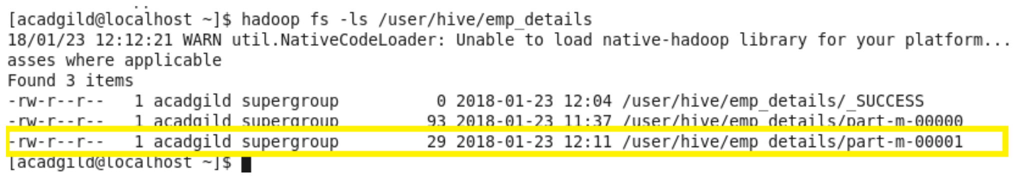
Here the only parameter which is added in comparison with the import command in step 3 is **last-value.** We need to mention the last value for the column in check-column parameter so that Sqoop starts inserting records appearing after the last value of emp_id column.



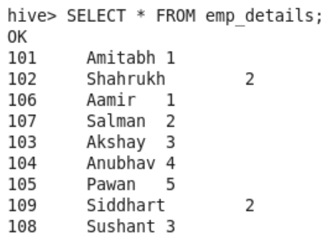
```
[acadgild@localhost hadoop-2.6.0]$ sqoop import --connect jdbc:mysql://localhost/sample_db --username root -P --table emp_inf
o --target-dir /user/hive/emp_details -m 1 --incremental append --check-column emp_id --last-value 107
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 11:44:10,610 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 11:44:12,394 INFO  [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 11:44:12,394 INFO  [main] tool.CodeGenTool: Beginning code generation
2018-01-23 11:44:12,964 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 11:44:13,064 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 11:44:13,083 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2018-01-23 11:44:18,786 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
2018-01-23 11:44:20,078 INFO  [main] tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`emp_id`)
 FROM emp_info
2018-01-23 11:44:20,084 INFO  [main] tool.ImportTool: Incremental import based on column `emp_id`
2018-01-23 11:44:20,084 INFO  [main] tool.ImportTool: Lower bound value: 107
2018-01-23 11:44:20,084 INFO  [main] tool.ImportTool: Upper bound value: 109
2018-01-23 11:44:20,084 WARN  [main] manager.MySQLManager: It looks like you are importing from mysql.
2018-01-23 11:44:20,084 WARN  [main] manager.MySQLManager: This transfer can be faster! Use the --direct
2018-01-23 11:44:20,085 WARN  [main] manager.MySQLManager: option to exercise a MySQL-specific fast path.
2018-01-23 11:44:20,085 INFO  [main] manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
2018-01-23 11:44:20,132 INFO  [main] mapreduce.ImportJobBase: Beginning import of emp_info
2018-01-23 11:44:20,198 INFO  [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2018-01-23 11:44:20,273 INFO  [main] Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.ma

2018-01-23 11:44:27,846 INFO  [main] mapreduce.Job: The url to track the job: http://http://localhost:8088/proxy/application_
1516637375581_0027/
2018-01-23 11:44:27,847 INFO  [main] mapreduce.Job: Running job: job_1516637375581_0027
2018-01-23 11:44:53,152 INFO  [main] mapreduce.Job: Job job_1516637375581_0027 running in uber mode : false
2018-01-23 11:44:53,176 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2018-01-23 11:45:14,945 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2018-01-23 11:45:20,237 INFO  [main] mapreduce.Job: Job job_1516637375581_0027 completed successfully
2018-01-23 11:45:20,976 ERROR [main] tool.ImportTool: Imported Failed: No enum constant org.apache.hadoop.mapreduce.JobCounte
r.MB_MILLIS_MAPS
[acadgild@localhost hadoop-2.6.0]$
```

In our case, the last value of emp_id is 107 and we have inserted two more records after the initial load with emp_id values: 108 and 109. So Sqoop will decide the lower and upper bounds by checking previous load statistics. We can see in the screenshot that it has come up with lower bound value as 107 and upper bound value as 109.

Let’s check the contents of target directory and then the corresponding table in Hive.



```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/emp_details
18/01/23 12:12:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 3 items
-rw-r--r--   1 acadgild supergroup          0 2018-01-23 12:04 /user/hive/emp_details/_SUCCESS
-rw-r--r--   1 acadgild supergroup         93 2018-01-23 11:37 /user/hive/emp_details/part-m-00000
-rw-r--r--   1 acadgild supergroup         29 2018-01-23 12:11 /user/hive/emp_details/part-m-00001
[acadgild@localhost ~]$
```



```
hive> SELECT * FROM emp_details;
OK
101     Amitabh 1
102     Shahrukh        2
106     Aamir   1
107     Salman  2
103     Akshay  3
104     Anubhav 4
105     Pawan   5
109     Siddhart        2
108     Sushant 3
```

As we can see here, the new records have been inserted into the Hive table successfully with the use of **incremental load feature provided by Sqoop**.