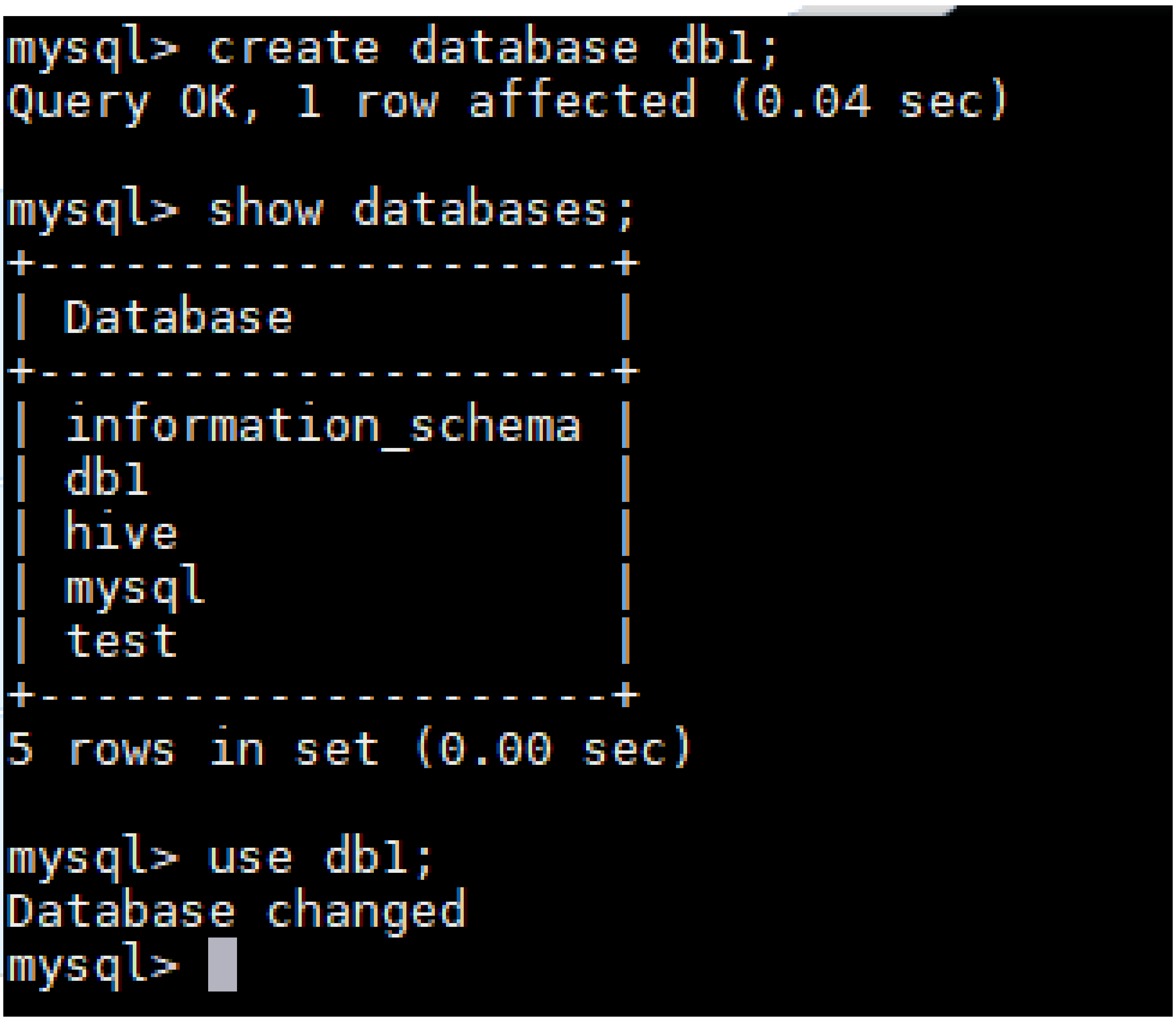
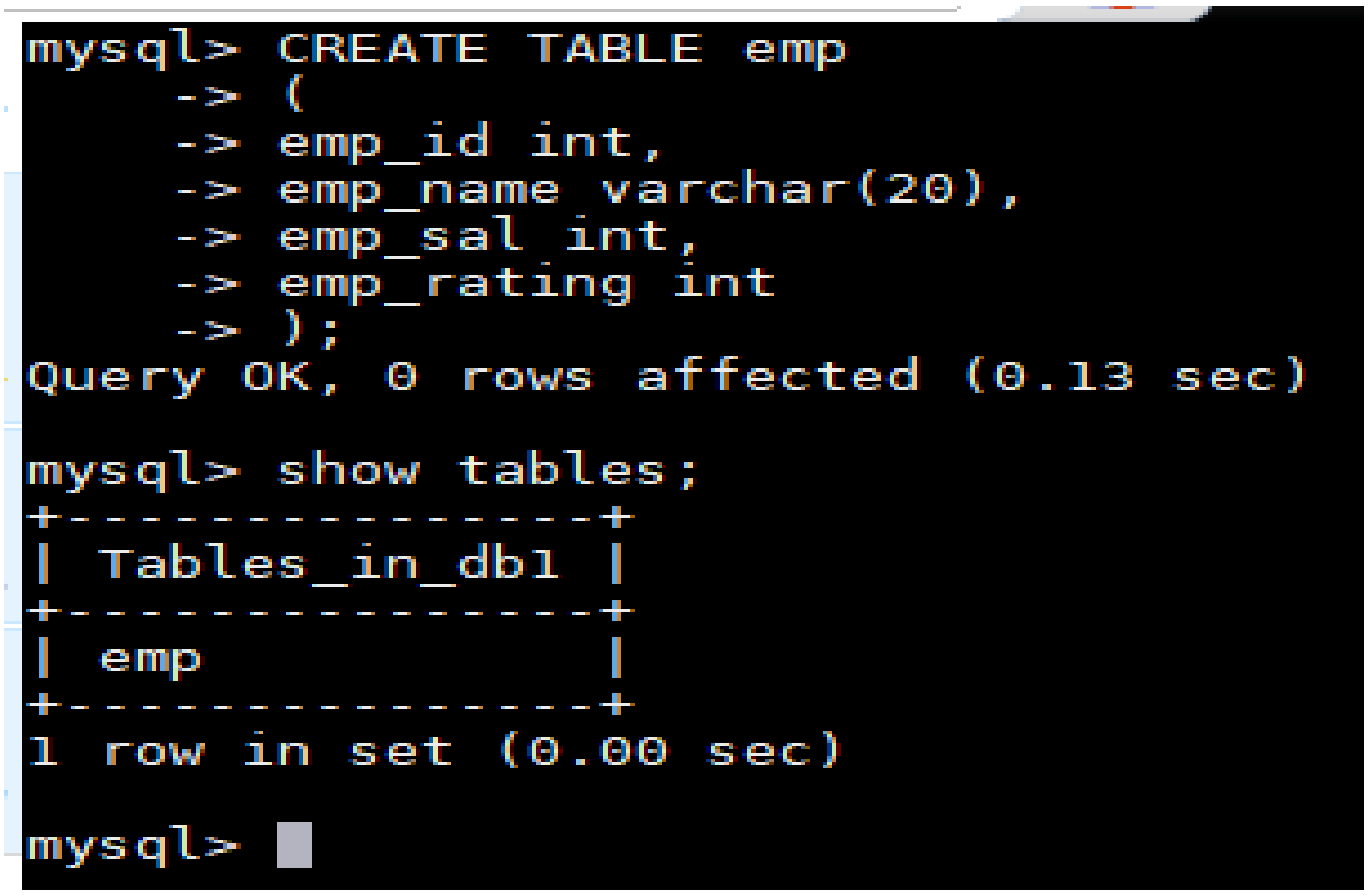
Problem Statement - Transfer data between Mysql and HDFS (Import and Export) using Sqoop.

1. **Import-** We will import first from MySQL to HDFS.
   In order to do that we will create a database db1 in MySQL first as shown below-



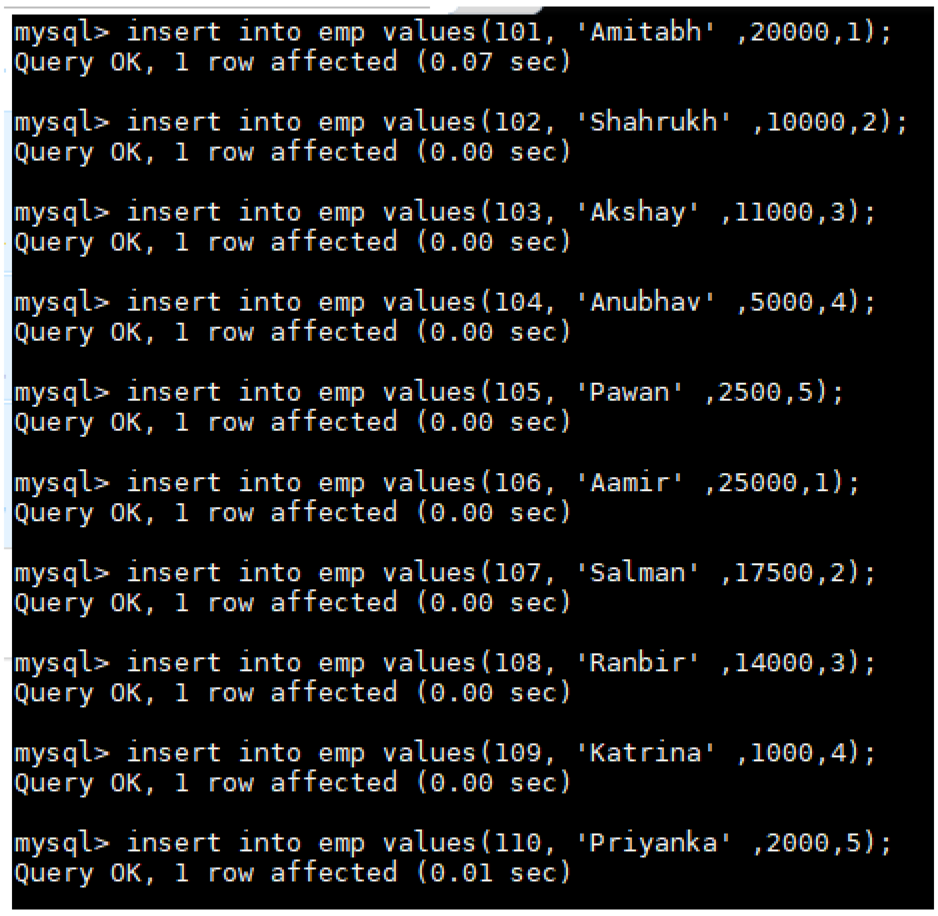
```
mysql> create database db1;
Query OK, 1 row affected (0.04 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db1                |
| hive               |
| mysql              |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql> use db1;
Database changed
mysql>
```

Now inside database db1 we will create a table emp with columns emp_id, emp_name, emp_sal and emp_rating as shown below-



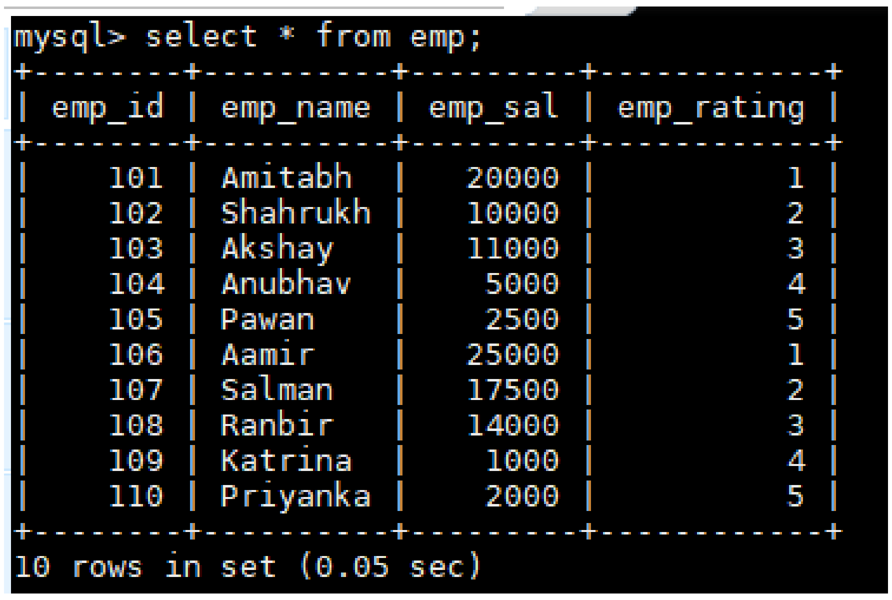
```
mysql> CREATE TABLE emp
    -> (
    -> emp_id int,
    -> emp_name varchar(20),
    -> emp_sal int,
    -> emp_rating int
    -> );
Query OK, 0 rows affected (0.13 sec)

mysql> show tables;
+---------------+
| Tables_in_db1 |
+---------------+
| emp           |
+---------------+
1 row in set (0.00 sec)

mysql>
```

Now after creating tables we will insert some values in it as shown below-



```
mysql> insert into emp values(101, 'Amitabh' ,20000,1);
Query OK, 1 row affected (0.07 sec)

mysql> insert into emp values(102, 'Shahrukh' ,10000,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(103, 'Akshay' ,11000,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(104, 'Anubhav' ,5000,4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(105, 'Pawan' ,2500,5);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(106, 'Aamir' ,25000,1);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(107, 'Salman' ,17500,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(108, 'Ranbir' ,14000,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(109, 'Katrina' ,1000,4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(110, 'Priyanka' ,2000,5);
Query OK, 1 row affected (0.01 sec)
```

Below screenshot shows the values which we have inserted above in the emp table-



```
mysql> select * from emp;
+--------+----------+---------+------------+
| emp_id | emp_name | emp_sal | emp_rating |
+--------+----------+---------+------------+
|    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 |
|    108 | Ranbir   |   14000 |          3 |
|    109 | Katrina  |    1000 |          4 |
|    110 | Priyanka |    2000 |          5 |
+--------+----------+---------+------------+
10 rows in set (0.05 sec)
```

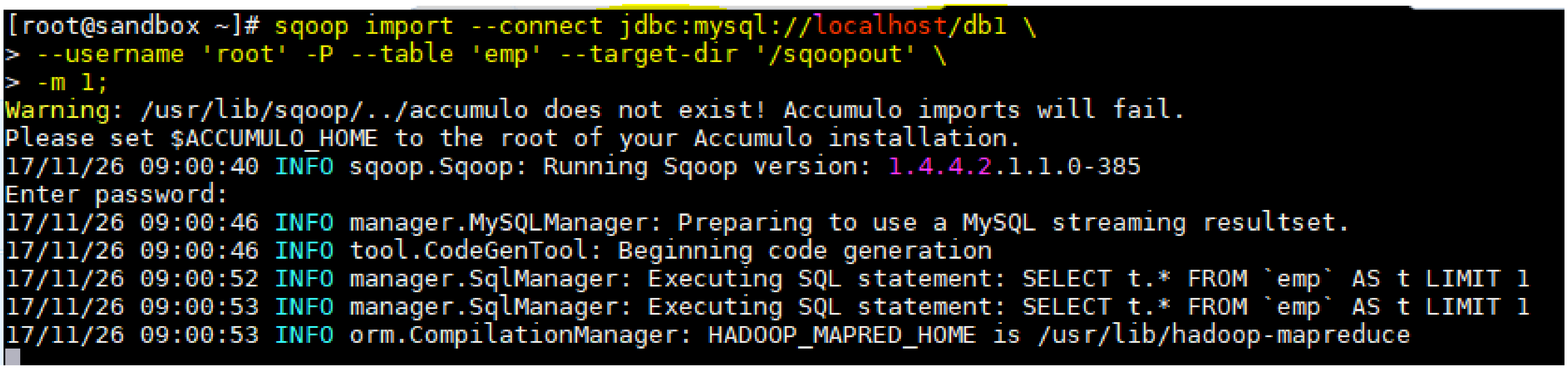
Now we will use sqoop import command to import data from above created emp table and load it into HDFS at location “sqoopout”

- **sqoop import --connect jdbc:mysql://localhost/db1 \**
- **--username 'root' -P --table 'emp' --target-dir '/sqoopout' \**
- **-m 1;**

  In above script we are first making a JDBC connection with MySQL and then specifying the username as root.

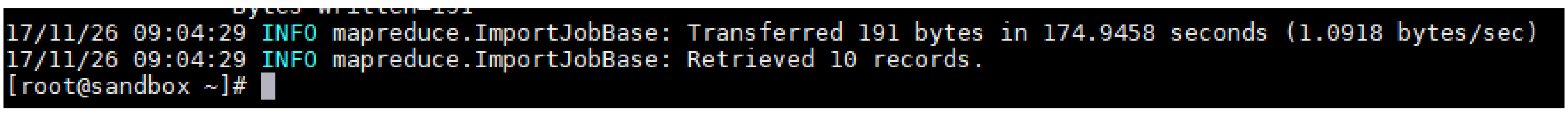
  -P specifies that we will be prompted for password. Then we are specifying table name as ‘emp’ and target as sqoopout in HDFS.

  “-m 1” specifies that this operation will use only 1 mapper.



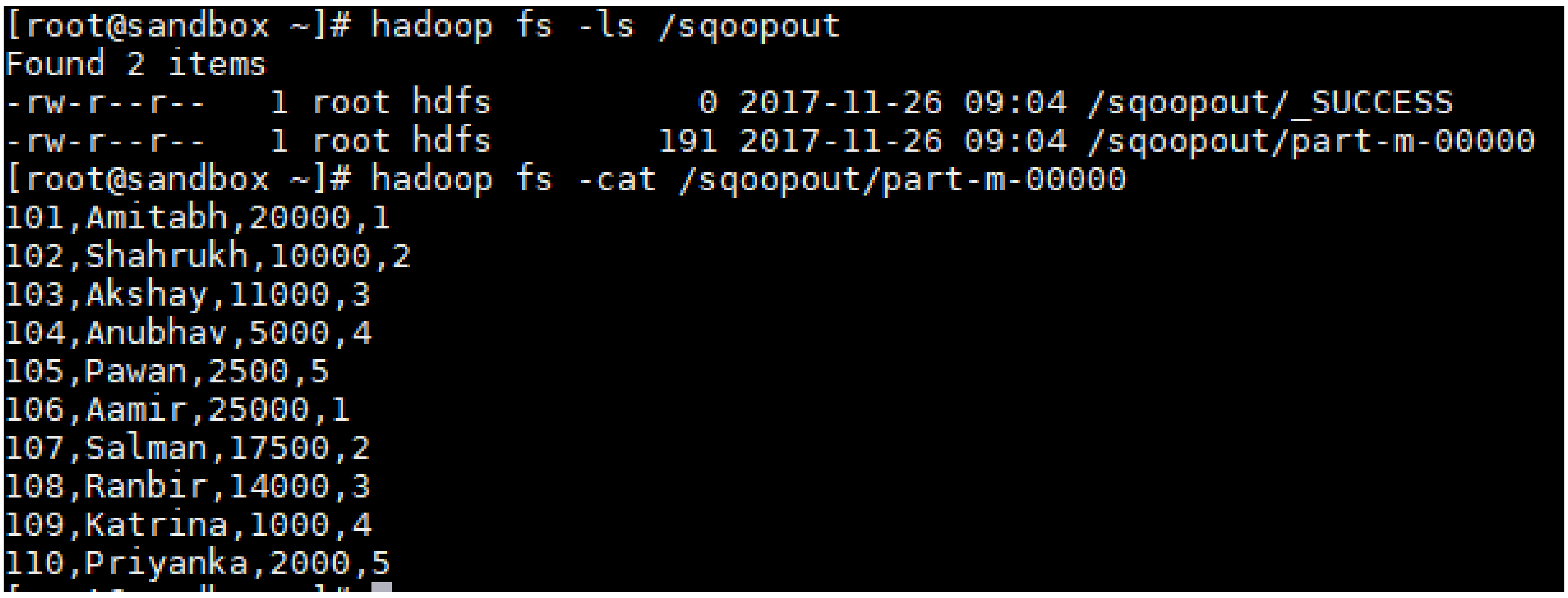
```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/db1 \
> --username 'root' -P --table 'emp' --target-dir '/sqoopout' \
> -m 1;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
17/11/26 09:00:40 INFO sqoop.Sqoop: Running Sqoop version: 1.4.4.2.1.1.0-385
Enter password:
17/11/26 09:00:46 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/11/26 09:00:46 INFO tool.CodeGenTool: Beginning code generation
17/11/26 09:00:52 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp` AS t LIMIT 1
17/11/26 09:00:53 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp` AS t LIMIT 1
17/11/26 09:00:53 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
```

As we can see in below screen shot that the operation was successful and 10 records were retrieved-



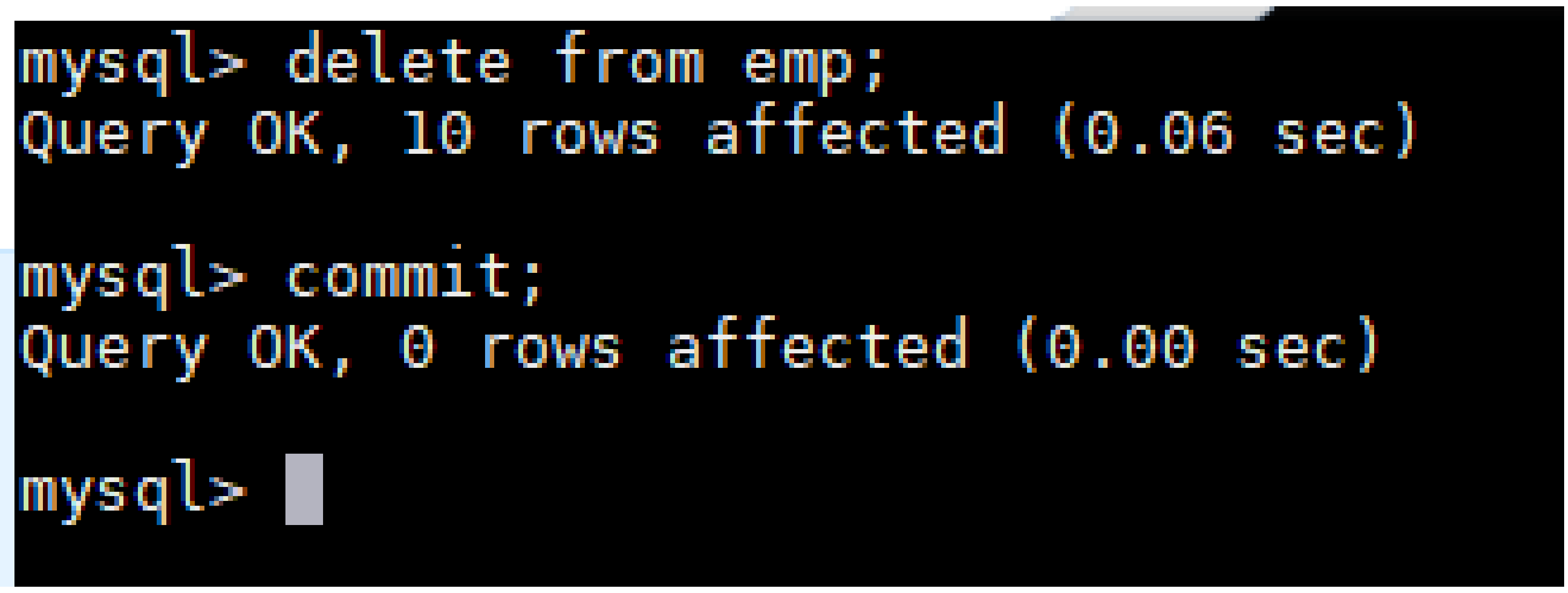
```
                Bytes Written=191
17/11/26 09:04:29 INFO mapreduce.ImportJobBase: Transferred 191 bytes in 174.9458 seconds (1.0918 bytes/sec)
17/11/26 09:04:29 INFO mapreduce.ImportJobBase: Retrieved 10 records.
[root@sandbox ~]#
```

Now if we go inside that HDFS location we can see the contents of table emp loaded in that location separated with ‘,’-



```
[root@sandbox ~]# hadoop fs -ls /sqoopout
Found 2 items
-rw-r--r--   1 root hdfs          0 2017-11-26 09:04 /sqoopout/_SUCCESS
-rw-r--r--   1 root hdfs        191 2017-11-26 09:04 /sqoopout/part-m-00000
[root@sandbox ~]# hadoop fs -cat /sqoopout/part-m-00000
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
108,Ranbir,14000,3
109,Katrina,1000,4
110,Priyanka,2000,5
```

2. **Export-** Now after importing we will delete the data from emp table and will try to load it again from HDFS using Sqoop-

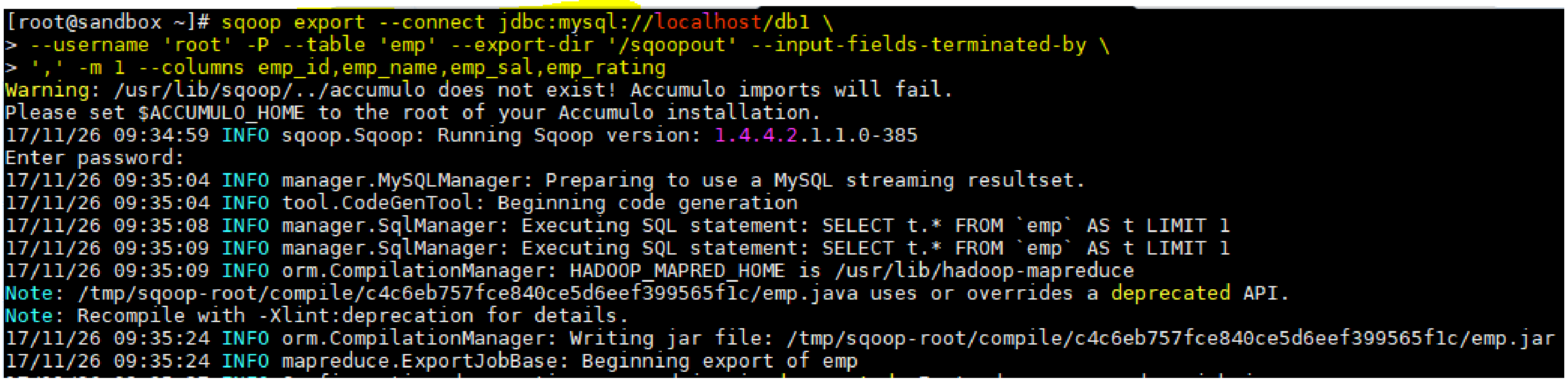


```
mysql> delete from emp;
Query OK, 10 rows affected (0.06 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Below is the sqoop script we have used to export from HDFS to MySQL-

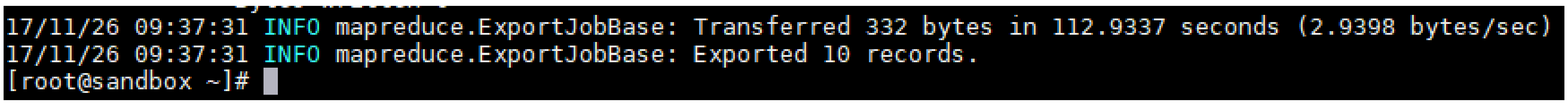
- **sqoop export --connect jdbc:mysql://localhost/db1 **
- **--username 'root' -P --table 'emp' --export-dir '/sqoopout' **
- **--input-fields-terminated-by ',' **
- **-m 1 --columns emp_id,emp_name,emp_sal,emp_rating**

  In above script we are first making a JDBC connection with db1 MySQL database. Specifying username as ‘root’ and –P specifies that it should ask for password. The table name we have specified as table.
  The export directory (sqoopout) is same where we imported the data from MySQL. We are also specifying the delimiter as ‘,’. Here also –m 1 determines that we are using only 1 mapper.



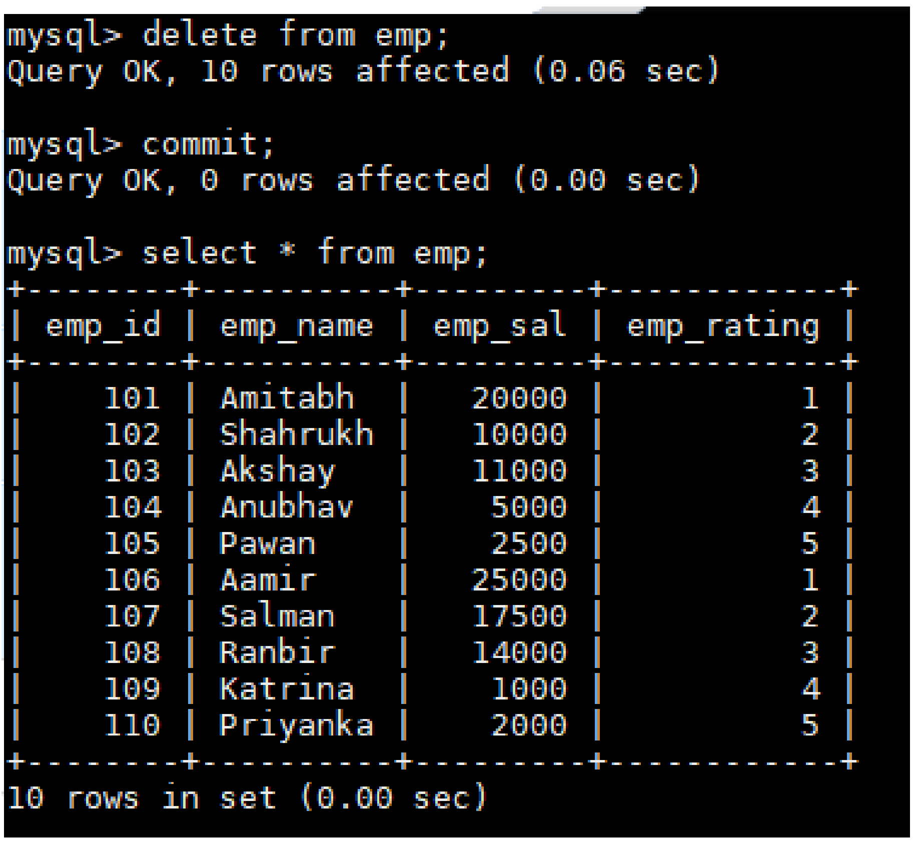
```
[root@sandbox ~]# sqoop export --connect jdbc:mysql://localhost/db1 \
> --username 'root' -P --table 'emp' --export-dir '/sqoopout' --input-fields-terminated-by \
> ',' -m 1 --columns emp_id,emp_name,emp_sal,emp_rating
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
17/11/26 09:34:59 INFO sqoop.Sqoop: Running Sqoop version: 1.4.4.2.1.1.0-385
Enter password:
17/11/26 09:35:04 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/11/26 09:35:04 INFO tool.CodeGenTool: Beginning code generation
17/11/26 09:35:08 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp` AS t LIMIT 1
17/11/26 09:35:09 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp` AS t LIMIT 1
17/11/26 09:35:09 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/c4c6eb757fce840ce5d6eef399565f1c/emp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
17/11/26 09:35:24 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/c4c6eb757fce840ce5d6eef399565f1c/emp.jar
17/11/26 09:35:24 INFO mapreduce.ExportJobBase: Beginning export of emp
```

Below screenshot shows that our export operation is successful and 10 records were exported-



```
17/11/26 09:37:31 INFO mapreduce.ExportJobBase: Transferred 332 bytes in 112.9337 seconds (2.9398 bytes/sec)
17/11/26 09:37:31 INFO mapreduce.ExportJobBase: Exported 10 records.
[root@sandbox ~]#
```

Meanwhile we can see in below screenshot that the table has been loaded and it contains the same 10 records which were present in HDFS location sqoopout.



```
mysql> delete from emp;
Query OK, 10 rows affected (0.06 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from emp;
+--------+----------+---------+------------+
| emp_id | emp_name | emp_sal | emp_rating |
+--------+----------+---------+------------+
|    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 |
|    108 | Ranbir   |   14000 |          3 |
|    109 | Katrina  |    1000 |          4 |
|    110 | Priyanka |    2000 |          5 |
+--------+----------+---------+------------+
10 rows in set (0.00 sec)
```