

Sqoop Tutorial

Step Description:

Login to the MySQL shell using username as root and password as root. Check for available databases and create a database by name wisdom_db.

Commands:

```
[jayantmohite@localhost] $ mysql -u <enter username as root> -p
```

Enter password: <enter password as root>

(you won't be able to see the password getting typed. But it is recorded by the terminal. So don't worry and simply type without any typing mistake. This will take you to the MySQL client shell.)

mysql> show databases;

(This will give you a list of all available databases in the server)

mysql> create database <enter database name as wisdom_db>;

Step Visualization:

The screenshot shows a terminal window titled 'jayantmohite@localhost:~'. The session starts with a MySQL connection command:

```
[jayantmohite@localhost ~]$ mysql -u root -p
```

It then displays the MySQL monitor welcome message:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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.
```

The user then runs the 'show databases' command:

```
mysql> show databases;
```

which lists the databases:

```
+-----+-----+
| Database |-----+
| information_schema |
| mysql |
| performance_schema |
| wisdom |
+-----+
```

Finally, the user creates a new database named 'wisdom_db':

```
4 rows in set (0.04 sec)

mysql> create database wisdom_db;
Query OK, 1 row affected (0.01 sec)

mysql> exit;
Bye
```

Step Description:

Select the target database as wisdom_db which we created in the above step and import the SQL file by name mysqlsampledatabase.sql (provided at the end of this chapter) to create the sample tables for our use.

Commands:

```
mysql> use <enter database name as wisdom_db>
```

```
mysql> source <file location>/mysqlsampledatabase.sql
```

Step Visualization:

```

Applications Places Terminal Fri 13:50
jayantmohite@localhost:~ - x
File Edit View Search Terminal Help
[jayantmohite@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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> use wisdom db;
Database changed
mysql> source /home/jayantmohite/mysqlsampledatabase.sql

```



```

Applications Places Terminal Fri 13:50
jayantmohite@localhost:~ - x
File Edit View Search Terminal Help
Query OK, 0 rows affected (0.04 sec)
Query OK, 2996 rows affected (0.24 sec)
Records: 2996 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.02 sec)
Query OK, 326 rows affected (0.06 sec)
Records: 326 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.08 sec)
Query OK, 273 rows affected (0.01 sec)
Records: 273 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.07 sec)
Query OK, 7 rows affected (0.00 sec)
Records: 7 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.02 sec)
Query OK, 110 rows affected (0.03 sec)
Records: 110 Duplicates: 0 Warnings: 0
Query OK, 0 rows affected (0.00 sec)
mysql>

```

Step Description:

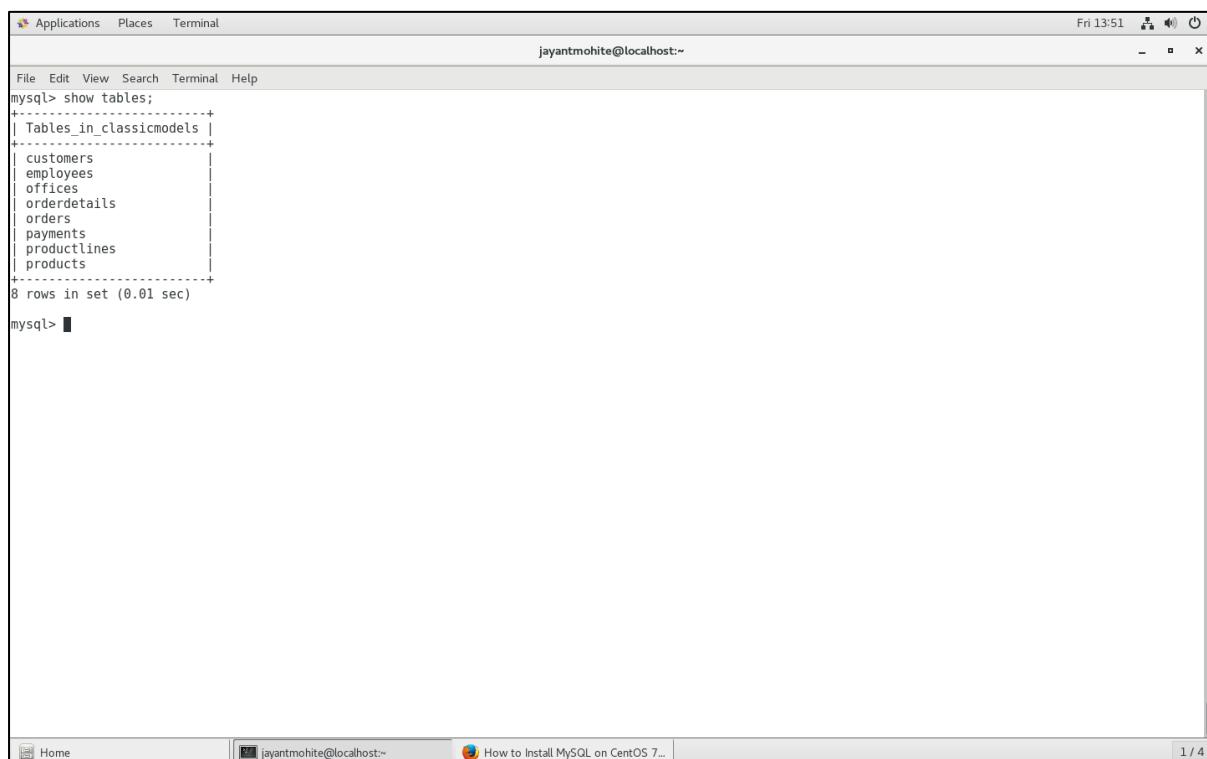
Check if the import was done successfully by listing all the tables. You should be able to see the following tables

- customers
- employees
- offices
- orderdetails
- orders
- payments
- productlines
- products

Commands:

```
mysql> show tables;
```

Step Visualization:



A screenshot of a terminal window titled "Terminal". The window shows the MySQL prompt "mysql>". The user has run the command "show tables;" and the output is displayed in a table format:

Tables_in_classicmodels
customers
employees
offices
orderdetails
orders
payments
productlines
products

Below the table, the message "8 rows in set (0.01 sec)" is shown. The terminal window has a standard Linux desktop interface with icons for Applications, Places, and Terminal at the top. The status bar at the bottom shows "jayantmohite@localhost:~" and "Fri 13:51".

Step Description:

You can use different tables and see how different commands in Sqoop work. For this lab session, we will be making use of the table employees. So let's try to get a glimpse of how this table looks.

Commands:

mysql> select count(*) from <enter table name as employees>;

(this will give you the count of the total records in the table)

mysql> describe <enter table name as employees>;

(this will show you the schema or structure of the table)

mysql> select * from <enter table name as employees> limit <enter integer value as 5>;

(this will give you the first 5 records from the employees table)

Step Visualization:

```

Applications Places Terminal
Fri 13:52  Applications Places Terminal Help
jayantmohite@localhost:~ - x
File Edit View Search Terminal Help
mysql> select count(*) from employees;
+-----+
| count(*) |
+-----+
|      23 |
+-----+
1 row in set (0.01 sec)

mysql> describe employees;
+-----+-----+-----+-----+-----+-----+
| Field      | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| employeeNumber | int(11) | NO   | PRI | NULL    |       |
| lastName     | varchar(50) | NO   |     | NULL    |       |
| firstName    | varchar(50) | NO   |     | NULL    |       |
| extension    | varchar(10)  | NO   |     | NULL    |       |
| email        | varchar(100) | NO   |     | NULL    |       |
| officeCode   | varchar(10)  | NO   | MUL | NULL    |       |
| reportsTo    | int(11)    | YES  | MUL | NULL    |       |
| jobTitle     | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.03 sec)

mysql> select * from employees limit 5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| employeeNumber | lastName | firstName | extension | email           | officeCode | reportsTo | jobTitle |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      1002 | Murphy   | Diane    | x5800    | dmurphy@classicmodelcars.com | 1          | NULL     | President |
|      1056 | Patterson | Mary     | x4611    | mpatters@classicmodelcars.com | 1          | 1002     | VP Sales  |
|      1076 | Firrelli  | Jeff     | x9273    | jfirrelli@classicmodelcars.com | 1          | 1002     | VP Marketing |
|      1088 | Patterson | William  | x4871    | wpatterson@classicmodelcars.com | 6          | 1056     | Sales Manager (APAC) |
|      1102 | Bondur   | Gerard   | x5408    | gbondur@classicmodelcars.com | 4          | 1056     | Sale Manager (EMEA) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> 

```

Step Description:

Browse the HDFS Web Browser. Here you can see that there is no directory by name employees. Rather what we have is the two directories that we created while installing Hadoop in the previous chapters.

This will help you to understand the changes that will be brought to effect by executing the sqoop commands.

In case you have created any more directories, you can always delete them by using the following command:

```
[jayantmohite@localhost] $ hadoop fs -rmr /user/jayantmohite/*
```

This command will delete all directories in the HDFS user location including the directories listed in the visualization below.

Step Visualization:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	jayantmohite	supergroup	0 B	Fri Mar 02 04:22:34 -0800 2018	0	0 B	input
drwxr-xr-x	jayantmohite	supergroup	0 B	Fri Mar 02 04:27:20 -0800 2018	0	0 B	output23

Step Description:

In this step we will be import the table employees from the MySQL database server installed in our client machine. This is called as Sqoop Default Import command in Sqoop terminology. We will use the following parameters:

Database Connector: jdbc (Java DataBase Connector)

Database Server: mysql

Installed on Server (hostname): localhost

Database: classicmodels

Table (to import data from): employees

Username (credentials for MySQL server validation): root

Password (credentials for MySQL server validation): root

Commands:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --username root --password root
```

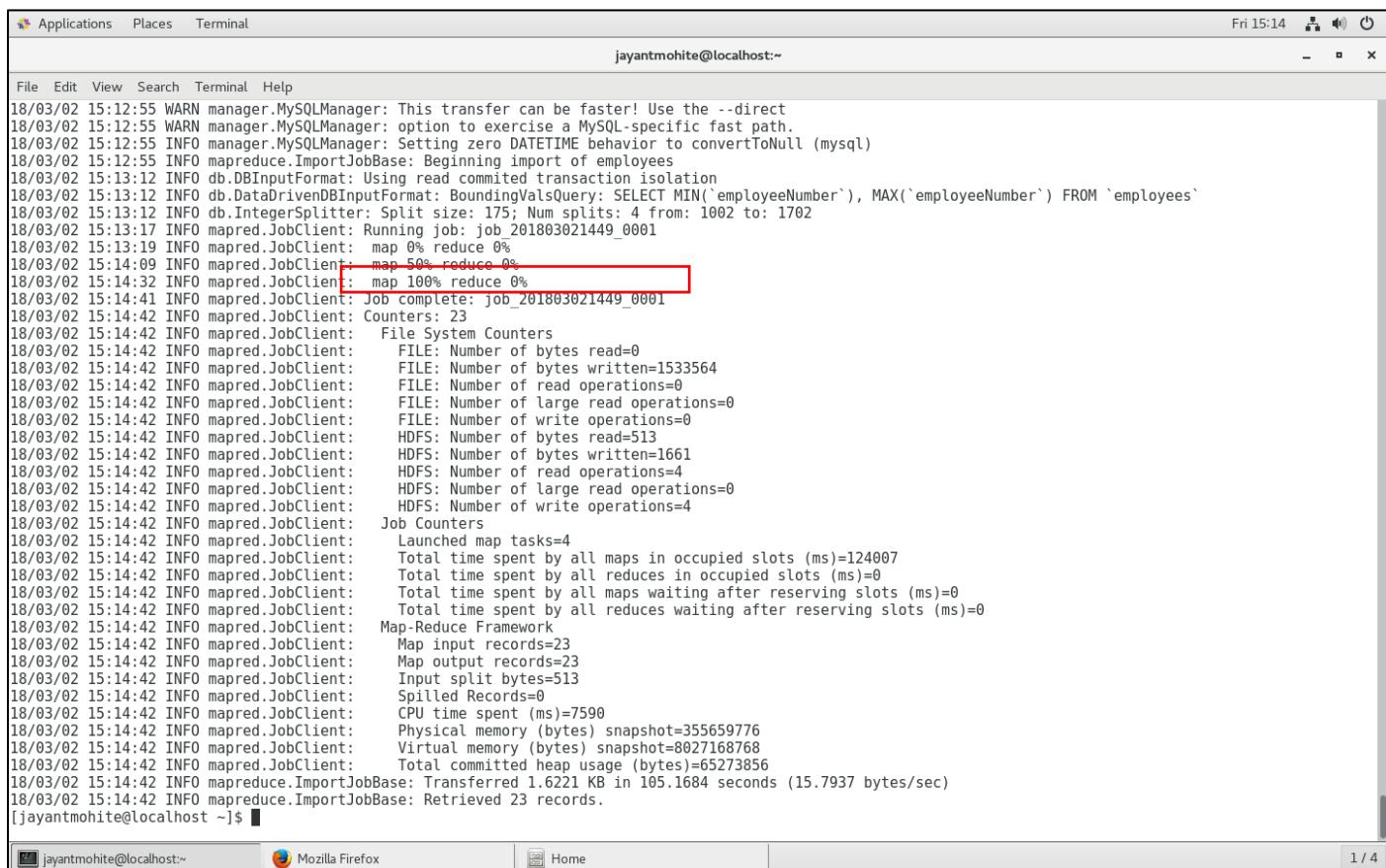
Step Visualization 1:

```
File Edit View Search Terminal Help
[jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --username root --password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:12:43 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:12:43 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:12:43 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:12:43 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:12:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:12:45 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:12:45 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/052cae84fb2ab6e726540d693b9cd0f/employees.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:12:55 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/052cae84fb2ab6e726540d693b9cd0f/employees.jar
18/03/02 15:12:55 WARN manager.MySQLManager: It looks like you are importing from mysql.
18/03/02 15:12:55 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/03/02 15:12:55 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/03/02 15:12:55 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/03/02 15:12:55 INFO mapreduce.ImportJobBase: Beginning import of employees
18/03/02 15:13:12 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:13:12 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`employeeNumber`), MAX(`employeeNumber`) FROM `employees`
18/03/02 15:13:12 INFO db.IntegerSplitter: Split size: 175; Num splits: 4 from: 1002 to: 1702
18/03/02 15:13:17 INFO mapred.JobClient: Running job: job_201803021449_0001
18/03/02 15:13:19 INFO mapred.JobClient: map 0% reduce 0%
```

Step Observations List 1:

- The given job is split into tasks based on the value of the primary key.
- Backend code of the sqoop operation triggers a Map Reduce Program
- Number of tasks in which the job is split is 4.
- This means 4 mappers will be used in this job.
- For Example if the maximum value of the primary key field is 12, then mapper 1 will fetch records having the value of primary key from 1 to 3; the 2nd mapper will fetch values 4 to 6; the 3rd will fetch the values from 7 to 9 and the 4th will fetch values from 10 to 12.
- This also means that there is a possibility that the size of output files produced by each mapper may differ. For example there are no records in the range 4 to 6; In this case mapper 2 will produce an empty file.

Step Visualization 2:



```

Applications Places Terminal Fri 15:14
jayantmohite@localhost:~ - x

File Edit View Search Terminal Help
18/03/02 15:12:55 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/03/02 15:12:55 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/03/02 15:12:55 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/03/02 15:12:55 INFO mapreduce.ImportJobBase: Beginning import of employees
18/03/02 15:13:12 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:13:12 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`employeeNumber`), MAX(`employeeNumber`) FROM `employees`
18/03/02 15:13:12 INFO db.IntegerSplitter: Split size: 175; Num splits: 4 from: 1002 to: 1702
18/03/02 15:13:17 INFO mapred.JobClient: Running job: job_201803021449_0001
18/03/02 15:13:19 INFO mapred.JobClient: map 0% reduce 0%
18/03/02 15:14:09 INFO mapred.JobClient: map 50% reduce 0%
18/03/02 15:14:32 INFO mapred.JobClient: map 100% reduce 0%
18/03/02 15:14:41 INFO mapred.JobClient: Job complete: job_201803021449_0001
18/03/02 15:14:42 INFO mapred.JobClient: Counters: 23
18/03/02 15:14:42 INFO mapred.JobClient: File System Counters
18/03/02 15:14:42 INFO mapred.JobClient: FILE: Number of bytes read=0
18/03/02 15:14:42 INFO mapred.JobClient: FILE: Number of bytes written=1533564
18/03/02 15:14:42 INFO mapred.JobClient: FILE: Number of read operations=0
18/03/02 15:14:42 INFO mapred.JobClient: FILE: Number of large read operations=0
18/03/02 15:14:42 INFO mapred.JobClient: FILE: Number of write operations=0
18/03/02 15:14:42 INFO mapred.JobClient: HDFS: Number of bytes read=513
18/03/02 15:14:42 INFO mapred.JobClient: HDFS: Number of bytes written=1661
18/03/02 15:14:42 INFO mapred.JobClient: HDFS: Number of read operations=4
18/03/02 15:14:42 INFO mapred.JobClient: HDFS: Number of large read operations=0
18/03/02 15:14:42 INFO mapred.JobClient: HDFS: Number of write operations=4
18/03/02 15:14:42 INFO mapred.JobClient: Job Counters
18/03/02 15:14:42 INFO mapred.JobClient: Launched map tasks=4
18/03/02 15:14:42 INFO mapred.JobClient: Total time spent by all maps in occupied slots (ms)=124007
18/03/02 15:14:42 INFO mapred.JobClient: Total time spent by all reduces in occupied slots (ms)=0
18/03/02 15:14:42 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
18/03/02 15:14:42 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
18/03/02 15:14:42 INFO mapred.JobClient: Map-Reduce Framework
18/03/02 15:14:42 INFO mapred.JobClient: Map input records=23
18/03/02 15:14:42 INFO mapred.JobClient: Map output records=23
18/03/02 15:14:42 INFO mapred.JobClient: Input split bytes=513
18/03/02 15:14:42 INFO mapred.JobClient: Spilled Records=0
18/03/02 15:14:42 INFO mapred.JobClient: CPU time spent (ms)=7590
18/03/02 15:14:42 INFO mapred.JobClient: Physical memory (bytes) snapshot=355659776
18/03/02 15:14:42 INFO mapred.JobClient: Virtual memory (bytes) snapshot=8027168768
18/03/02 15:14:42 INFO mapred.JobClient: Total committed heap usage (bytes)=65273856
18/03/02 15:14:42 INFO mapreduce.ImportJobBase: Transferred 1.6221 KB in 105.1684 seconds (15.7937 bytes/sec)
18/03/02 15:14:42 INFO mapreduce.ImportJobBase: Retrieved 23 records.
[jayantmohite@localhost ~]$ 

```

Step Observations List 2:

- No Reducer is used in this operation. As it's a simple copy paste job and no aggregation is involved.

Author: Jayant Mohite

Step Visualization 3:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	jayantmohite	supergroup	0 B	Fri Mar 02 15:14:40 -0800 2018	0	0 B	employees

Hadoop, 2017.

Step Observation List 3 (combines observations from visualization 4 and 5):

- By default this sqoop commands create the output directory with the same name as that of the table.
- The processing jar file is created default by sqoop with the name same as the name of the table.
- The structure of the output directory is same as it used to be in case of the Map Reduce programs.
- There are 4 output files (as the number of mappers used were 4 and numbers of reducers used was 0)
- Size of all output files varies (reason is explained in previous observations).

Step Visualization 4:

The screenshot shows the Hadoop Map/Reduce Administration interface in Mozilla Firefox. The page displays the following information:

- Cluster Summary (Heap Size is 15.56 MB/966.69 MB)**
- Scheduling Information**: A table showing one queue named "default" in state "running" with "N/A" scheduling information.
- Running Jobs**: A table showing one job named "job_201803021449_0001" in "NORMAL" priority, user "jayantmohite", and name "employees.jar". The map percentage is 100.00%, and both map and reduce tasks are at 100.00% completion.
- Completed Jobs**: No completed jobs listed.
- Retired Jobs**: No retired jobs listed.
- Local Logs**: A log entry from "jayantmohite@localhost" dated "Fri Mar 02 15:14:40 -0800 2018" indicating a successful job run.

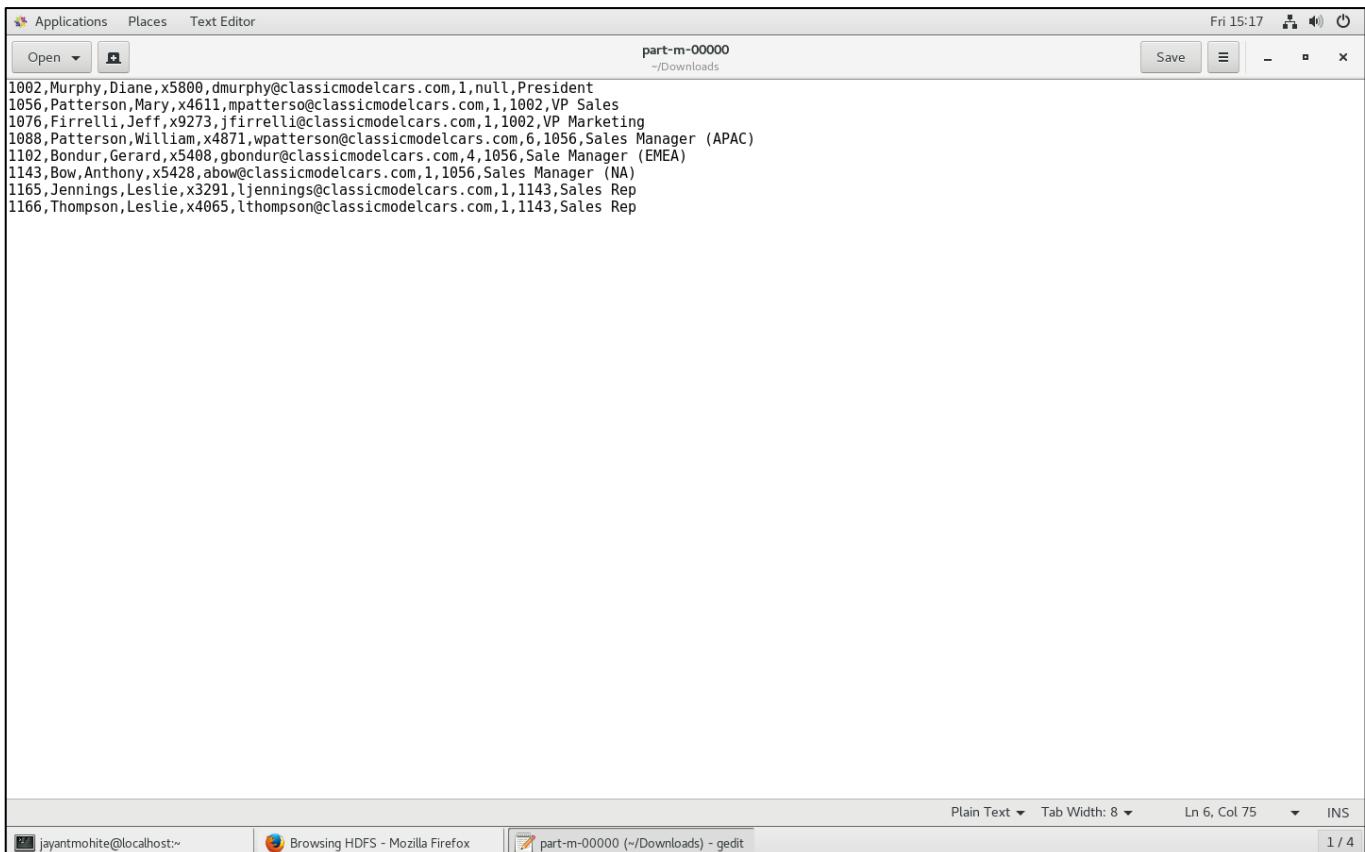
Step Visualization 5:

The screenshot shows the HDFS browser interface in Mozilla Firefox. The page displays the following information:

- Browsing HDFS**: A table listing files in the directory "/user/jayantmohite/employees". The table includes columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name.
- Browsing HDFS - Mozilla Firefox**: A status message at the bottom stating "Hadoop, 2017."

Author: Jayant Mohite

Step Visualization 6:



```
1002,Murphy,Diane,x5800,dmurphy@classicmodelcars.com,1,null,President
1056,Patterson,Mary,x4611,mpatterso@classicmodelcars.com,1,1002,VP Sales
1076,Firrelli,Jeff,x9273,jfirrelli@classicmodelcars.com,1,1002,VP Marketing
1088,Patterson,William,x4871,wpatterson@classicmodelcars.com,6,1056,Sales Manager (APAC)
1102,Bondur,Gerard,x5408,gbondur@classicmodelcars.com,4,1056,Sale Manager (EMEA)
1143,Bow,Anthony,x5428,abow@classicmodelcars.com,1,1056,Sales Manager (NA)
1165,Jennings,Leslie,x3291,ljennings@classicmodelcars.com,1,1143,Sales Rep
1166,Thompson,Leslie,x4065,lthompson@classicmodelcars.com,1,1143,Sales Rep
```

Step Visualization List 4:

If you download either of the output files, you will be able to observe the following

- Every output contains some number of records based on certain range
- The default delimiter used by which every field is separated from the other is comma.

Step Description:

In this step we will override the default behavior of the sqoop command and will specify our own target directory for dumping the output files.

Commands:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop1 --username root -password root
```

New Parameter Used:

--target-directory

(the value of this parameter will become the output directory)

Step Visualization:

The screenshot shows a Linux desktop environment with a terminal window and a web browser window.

Terminal Window:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop1 --username root -password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:18:31 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:18:31 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:18:32 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:18:32 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:18:33 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:18:33 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:18:33 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/0a4f975a0613e0dbcba6724c1f4ca0/employees.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:18:43 INFO compilation.CompilationManager: Writing job jar file:/tmp/sqoop-jayantmohite/compile/0a4f975a0613e0dbcba6724c1f4ca0/employees.jar
18/03/02 15:18:44 [jayantmohite@localhost:~]$
```

Firefox Browser Window:

The browser is displaying the HDFS file browser interface at localhost:50070/explorer.html#/user/jayantmohite. The page shows a list of files in the directory `/user/jayantmohite`.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	jayantmohite	supergroup	0 B	Fri Mar 02 15:14:40 -0800 2018	0	0 B	employees
drwxr-xr-x	jayantmohite	supergroup	0 B	Fri Mar 02 15:20:52 -0800 2018	0	0 B	sqoop1

The file `sqoop1` is highlighted with a red box.

Step Description:

In this step we will change the default delimiter used and specify a delimiter of our choice. In this case we are specifying the delimiter as tab.

Commands:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop2 --fields-terminated-by '\t' --username root -password root
```

New Parameter Used:

--fields-terminated-by

(the value of this parameter will specify the special character that is to be used as a separator between the fields. '\t' represents tab which is equal to 4 spaces)

Step Visualization:

The screenshot shows a Linux desktop interface with a terminal window and a text editor window.

Terminal Window:

```
[jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop2 --fields-terminated-by '\t' --username root -password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:22:07 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:22:07 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:22:07 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:22:07 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:22:08 INFO manager.SqlManager: Executing SQL statement: SELECT * FROM `employees` AS t1 LIMIT 1
18/03/02
18/03/02
Note: /tmp/part-m-00000(1) ->/Downloads
```

Text Editor Window:

The text editor window displays the contents of the file `part-m-00000(1)` located at `/tmp/part-m-00000(1)`. The file contains the following data:

ID	First Name	Last Name	Extension	Email	Order ID	Null	Title
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	null	President
1056	Patterson	Mary	x4611	mpatters@classicmodelcars.com	1	1002	VP Sales
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep

Step Description:

In this step we will make use of only one mapper for performing our task. This is also called as Sqoop Sequential Import.

This is one of the options available when your are working with a table that does not have a primary key.

Commands:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop3 -m 1 --username root -password root
```

New Parameter Used:

-m

(the value of this parameter specifies the number of mappers or the degree of parallelism to be applied in the job execution. In this case we are specifying it as 1 which is called as sequential import)

Step Visualization:

```
File Edit View Search Terminal Help
[jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees -m 1 --target-dir sqoop3 --username root --password root
Warning: /usr/lib/sqoop/..../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation
Warning: /us Applications Places Terminal
Please set $HBASE_HOME to the root of your HBase installation
18/03/02 15:31:39 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
Note: /tmp/sqoop-jayantmohite/compile/f4fc8f8325f087b335ac9210e44e08e9/employees.java uses or overrides a deprecated API.
Note: Recompiling with -Xlint:deprecation for details.
18/03/02 15:31:39 INFO tool.CodeGenTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:31:40 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:31:40 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:31:42 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:31:42 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:31:42 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
18/03/02 15:31:49 INFO sqoop.Jayantmohite/compile/f4fc8f8325f087b335ac9210e44e08e9/employees.java uses or overrides a deprecated API.
Note: Recompiling with -Xlint:deprecation for details.
18/03/02 15:31:50 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/f4fc8f8325f087b335ac9210e44e08e9/employees.jar
18/03/02 15:31:50 WARN manager.MySQLManager: It looks like you are importing from mysql.
18/03/02 15:31:50 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/03/02 15:31:50 INFO manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/03/02 15:31:50 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/03/02 15:31:50 INFO mapred.ImportJobBase: Beginning import of employees
18/03/02 15:32:08 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:32:10 INFO mapred.JobClient: Running job: job_201803021449_0004
18/03/02 15:32:11 INFO mapred.JobClient: map 0% reduce 0%
18/03/02 15:32:39 INFO mapred.JobClient: map 100% reduce 0%
18/03/02 15:32:48 INFO mapred.JobClient: Job complete: job_201803021449_0004
18/03/02 15:32:48 INFO mapred.JobClient: Counters: 23
18/03/02 15:32:49 INFO mapred.JobClient: File System Counters
18/03/02 15:32:49 INFO mapred.JobClient: FILE: Number of bytes read=0
18/03/02 15:32:49 INFO mapred.JobClient: FILE: Number of bytes written=383932
18/03/02 15:32:49 INFO mapred.JobClient: FILE: Number of read operations=0
18/03/02 15:32:49 INFO mapred.JobClient: FILE: Number of large read operations=0
18/03/02 15:32:49 INFO mapred.JobClient: FILE: Number of write operations=0
18/03/02 15:32:49 INFO mapred.JobClient: HDFS: Number of bytes read=87
18/03/02 15:32:49 INFO mapred.JobClient: HDFS: Number of bytes written=1661
18/03/02 15:32:49 INFO mapred.JobClient: HDFS: Number of read operations=1
18/03/02 15:32:49 INFO mapred.JobClient: HDFS: Number of large read operations=0
18/03/02 15:32:49 INFO mapred.JobClient: HDFS: Number of write operations=1
18/03/02 15:32:49 INFO mapred.JobClient: Launched map tasks=1
18/03/02 15:32:49 INFO mapred.JobClient: Total time spent by all maps in occupied slots (ms)=31978
18/03/02 15:32:49 INFO mapred.JobClient: Total time spent by all reduces in occupied slots (ms)=0
18/03/02 15:32:49 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
```

Author: Jayant Mohite

Step Description:

In this step we will further use the parameter of specifying the number of mappers to provide our own choice of degree of parallelism or number of mappers.

You can use this option in order to reduce the time consumption for the import in case of tables with large numbers of records.

Command:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop4 -m 5 --username root -password root
```

Step Visualization:

```
[jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees -m 5 --target-dir sqoop4 --username root --password root
Warning: /usr/lib/sqoop/.../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/.../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:34:31 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:34:31 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:34:32 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:34:32 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:34:33 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:34:33 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:34:33 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/b17a87fcbe6fdf1a63aa37ef39529b2/employees.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:34:42 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/b17a87fcbe6fdf1a63aa37ef39529b2/employees.jar
18/03/02 15:34:42 WARN manager.MySQLManager: It looks like you are importing from mysql.
18/03/02 15:34:42 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/03/02 15:34:42 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/03/02 15:34:42 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/03/02 15:34:42 INFO mapreduce.ImportJobBase: Beginning import of employees
18/03/02 15:35:01 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:35:01 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`employeeNumber`), MAX(`employeeNumber`) FROM `employees`
18/03/02 15:35:01 INFO db.IntegerSplitter: Split size: 140; Num splits: 5 from: 1002 to: 1702
18/03/02 15:35:03 INFO mapred.JobClient: Running job: job_201803021440_0005
18/03/02 15:35:04 INFO mapred.JobClient: map 0% reduce 0%
```

Step Description:

In this step we will specify the fields which should be used instead of the primary key to split the job into multiple tasks.

This is one of the options available when you are working with a table that does not have a primary key field.

Command:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop5 --split-by officeCode --username root -password root
```

New Parameter Used:

--split-by

(the value of this field will specify the field that should be used instead of the primary key field or incase of a table which does not contain a primary key field)

Step Visualization:

```
jayantmohite@localhost:~$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --split-by officeCode --target-dir sqoop5 --username root -password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:38:43 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:38:43 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:38:43 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:38:43 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:38:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:38:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
18/03/02 15:38:44 org.apache.hadoop.mapred.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/e3162ae62d17a2954fc4b642623120d6/employees.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:38:44 org.apache.hadoop.mapred.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/e3162ae62d17a2954fc4b642623120d6/employees.jar
18/03/02 15:38:53 WARN manager.MySQLManager: It looks like you are importing from mysql.
18/03/02 15:38:53 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/03/02 15:38:53 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/03/02 15:38:53 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/03/02 15:38:53 INFO mapreduce.ImportJobBase: Beginning import of employees
18/03/02 15:39:06 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:39:06 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`officeCode`), MAX(`officeCode`) FROM `employees`
18/03/02 15:39:07 WARN db.TextSplitter: Generating splits for a textual index column.
18/03/02 15:39:07 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.
18/03/02 15:39:07 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.
18/03/02 15:39:09 INFO mapred.JobClient: Running job: job_201803021449_0007
18/03/02 15:39:10 INFO mapred.JobClient: map 0% reduce 0%
```

Step Description:

In this step, instead of importing the complete data set of the table, we will specify a criteria based on which the records should be fetched.

You can also specify the projection which will limit the fields that are displayed in the output.

Commands:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop6 --where 'officeCode=1' --columns lastName,firstName --username root -password root
```

New Parameters Used:

--where

(this is the condition that needs to be executed in order to filter the output records)

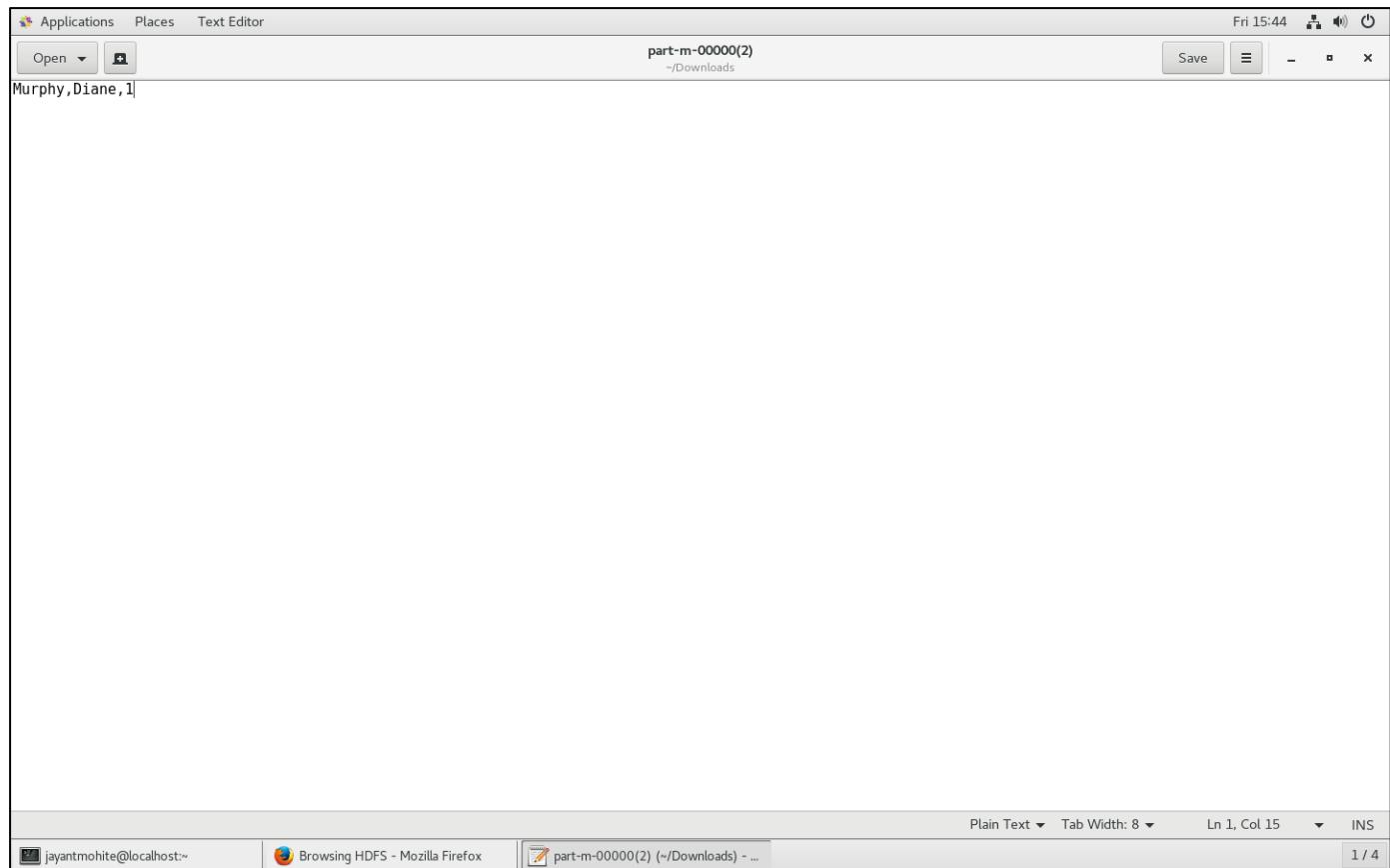
--columns

(this is a comma separated list of columns that should be a part of the output)

Step Visualization 1:

```
Fri 15:42 Applications Places Terminal jayantmohite@localhost:~ File Edit View Search Terminal Help [jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --where 'officeCode=1' --columns lastName,firstName, officeCode --target-dir sqoop6 --username root --password root Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail. Please set $HBASE_HOME to the root of your HBase installation. Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail. Please set $ACCUMULO_HOME to the root of your Accumulo installation. 18/03/02 15:41:26 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0 18/03/02 15:41:26 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead. 18/03/02 15:41:27 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset. 18/03/02 15:41:27 INFO tool.CodeGenTool: Beginning code generation 18/03/02 15:41:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1 18/03/02 15:41:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1 18/03/02 15:41:28 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce Note: /tmp/sqoop-jayantmohite/compile/3808599d16fc47f197bbc3309c5338aa/employees.java uses or overrides a deprecated API. Note: Recompile with -Xlint:deprecation for details. 18/03/02 15:41:38 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/3808599d16fc47f197bbc3309c5338aa/employees.jar 18/03/02 15:41:38 WARN manager.MySQLManager: It looks like you are importing from mysql. 18/03/02 15:41:38 WARN manager.MySQLManager: This transfer can be faster! Use the --direct 18/03/02 15:41:38 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path. 18/03/02 15:41:38 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql) 18/03/02 15:41:38 INFO mapreduce.ImportJobBase: Beginning import of employees 18/03/02 15:41:51 INFO db.DBInputFormat: Using read committed transaction isolation 18/03/02 15:41:51 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`employeeNumber`), MAX(`employeeNumber`) FROM `employees` WHERE ( officeCode=1 ) 18/03/02 15:41:51 INFO db.IntegerSplitter: Split size: 41; Num splits: 4 from: 1002 to: 1166 18/03/02 15:41:54 INFO mapred.JobClient: Running job: job_201803021449_0008 18/03/02 15:41:55 INFO mapred.JobClient: map 0% reduce 0%
```

Step Visualization 2:



Step Description:

In this step we will specify a query and the import will be the records fetched by the execution of this query. It can be a simple query, a query with criteria and projection, a query with joins, a sub query or a nested query as well.

By far you might have observed one thing that whenever you need to do something extra than what the default Sqoop command provides or if your need to override any default behavior of Sqoop, you need to introduce a new parameter.

If we consider a query as a parameter, then the query itself can be of different types and after debugging may produce multiple parameters with different values like table_name, criteria, projections, join tables, join conditions, etc.

If the user has to manage this and specify this then it would turn out to be an overhead and then Sqoop instead of reducing workload will rather turn out increasing it.

So in order to avoid this situation, sqoop manages the debugging of the query on its end so that the user does not have to get into these complications.

So no matter what sort of query it is, you need to compulsory have a where clause in the query with the variable \$CONDITIONS.

Basically this variable is used by sqoop to store the parameters and their values that would be generated after debugging the query.

Command:

```
[jayantmohite@localhost] $ sqoop import --connect jdbc:mysql://localhost/classicmodels --table employees --target-dir sqoop7 --username root -password root
```

New Parameter used:

--query

(this has to be the exact query you want to execute. Any query that produces a valid result on the database can be used here.)

Step Visualization:

```
[jayantmohite@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/classicmodels --query 'select firstName,lastName from employees where officeCode=1 and $CONDITIONS' -m 1 --target-dir sqoop7 --username root --password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:46:38 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:46:39 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:46:39 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:46:39 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:46:41 INFO manager.SqlManager: Executing SQL statement: select firstName,lastName from employees where officeCode=1 and (1 = 0)
18/03/02 15:46:41 INFO manager.SqlManager: Executing SQL statement: select firstName,lastName from employees where officeCode=1 and (1 = 0)
18/03/02 15:46:41 INFO manager.SqlManager: Executing SQL statement: select firstName,lastName from employees where officeCode=1 and (1 = 0)
18/03/02 15:46:41 INFO orm.CompilationManager: HADOOP MAPRED HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/c1da43f4f7e0fa92433d4e9a18a026a6/QueryResult.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:46:50 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/c1da43f4f7e0fa92433d4e9a18a026a6/QueryResult.jar
18/03/02 15:46:50 INFO mapreduce.ImportJobBase: Beginning query import.
18/03/02 15:47:06 INFO db.DBInputFormat: Using read committed transaction isolation
18/03/02 15:47:08 INFO mapred.JobClient: Running job: job_201803021449_0009
18/03/02 15:47:09 INFO mapred.JobClient: map 0% reduce 0%
```

The terminal window has a title bar 'jayantmohite@localhost:~'. The bottom status bar shows 'jayantmohite@localhost:~' and '[Browsing HDFS - Mozilla Firefox]'. The bottom right corner of the window frame says '1 / 4'.

Step Description:

In this step we will export create a table in MySQL database by name jayant_table. Our aim is to add data to this table from a text file which is saved in HDFS.

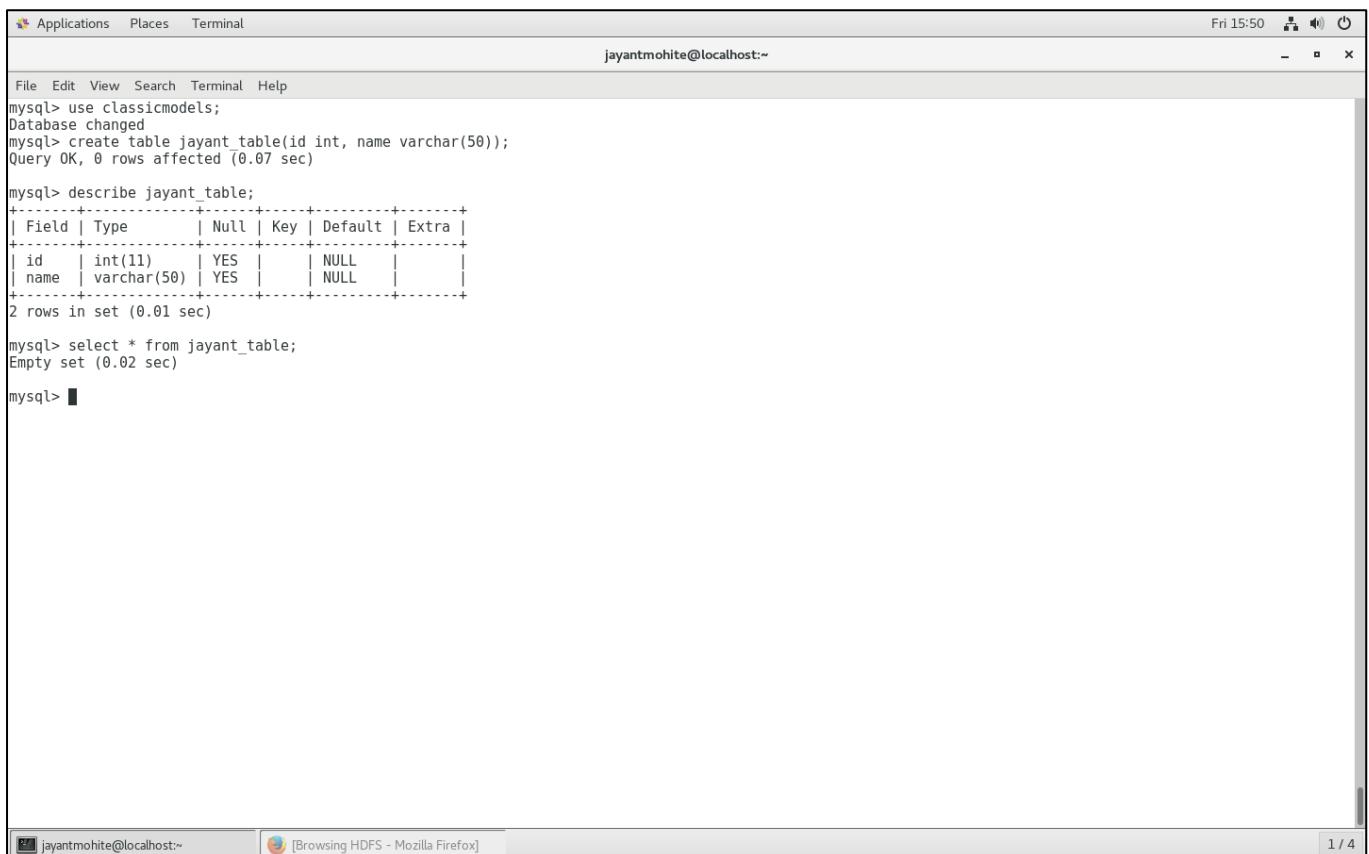
The schema of this table will resemble the input file which is store in HDFS.

Commands:

```
mysql> use classicmodels;
```

```
mysql> create table jayant_table(id int, name varchar(50))
```

Step Visualization:



```
Applications Places Terminal Fri 15:50
jayantmohite@localhost:~ - x
File Edit View Search Terminal Help
mysql> use classicmodels;
Database changed
mysql> create table jayant_table(id int, name varchar(50));
Query OK, 0 rows affected (0.07 sec)

mysql> describe jayant_table;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| id   | int(11) | YES  |     | NULL    |       |
| name | varchar(50)| YES |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from jayant_table;
Empty set (0.02 sec)

mysql> ■
```

Step Description:

In this step we will create our input file by name jayant_file.txt using the file editor called as gedit available in CentOS. We will store this file in our home location on the Linux File System.

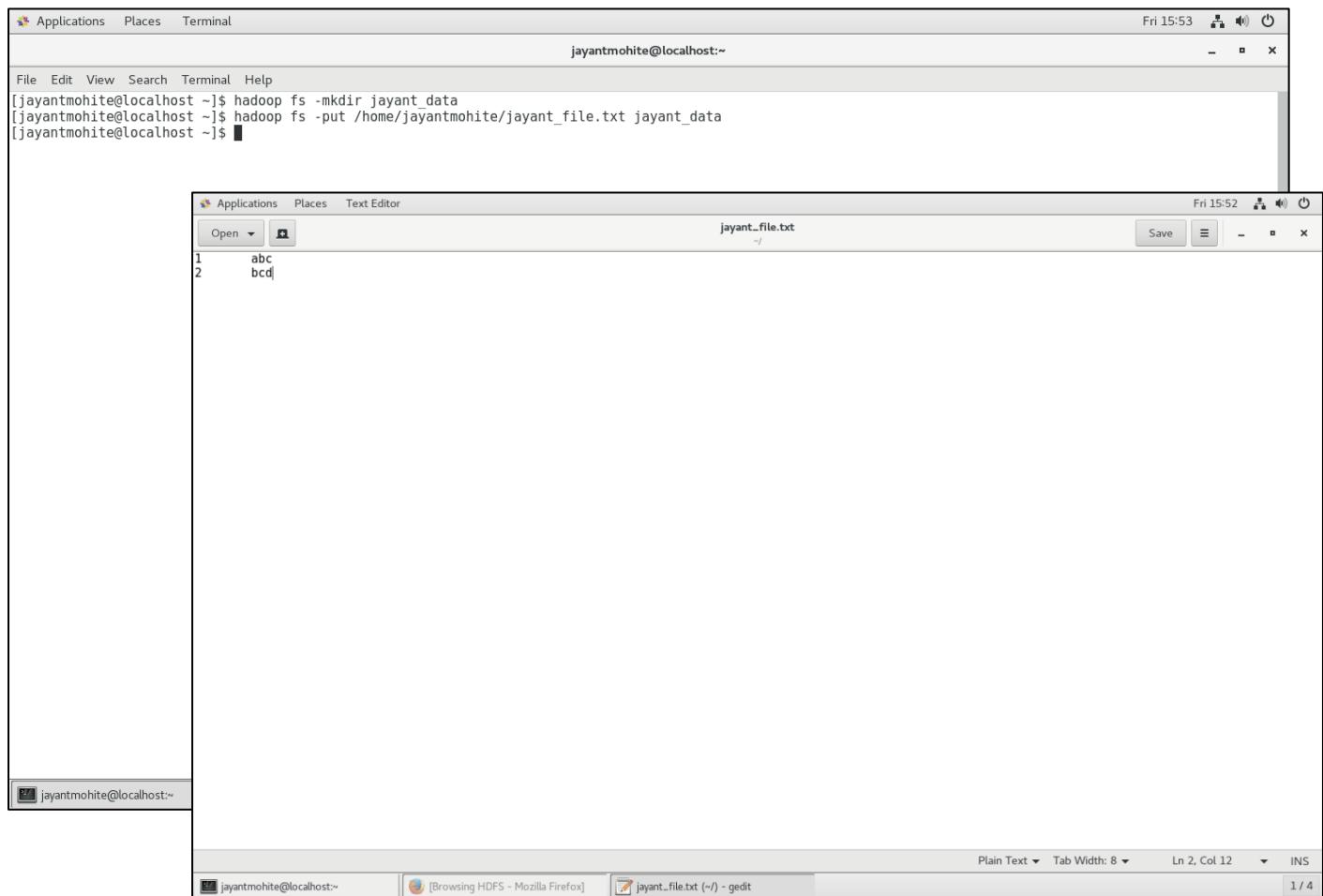
We will create a directory by name jayant_data in our HDFS and store the file in this directory

Commands:

```
[jayantmohite@localhost] $ hadoop fs -mkdir <enter name of directory to be created as
jayant_data>
(this will create the directory under the location /user/jayantmohite/)
```

```
[jayantmohite@localhost] $ hadoop fs -put <enter source location as
/home/jayantmohite/jayant_file.txt> <enter destination location as jayant_data>
```

Step Visualization:



Step Description:

In this step we will load the data from the file named jayant_file.txt stored in hdfs location of /user/jayantmohite/jayant_data into the table jayant_table that we created earlier in MySQL database.

Commands:

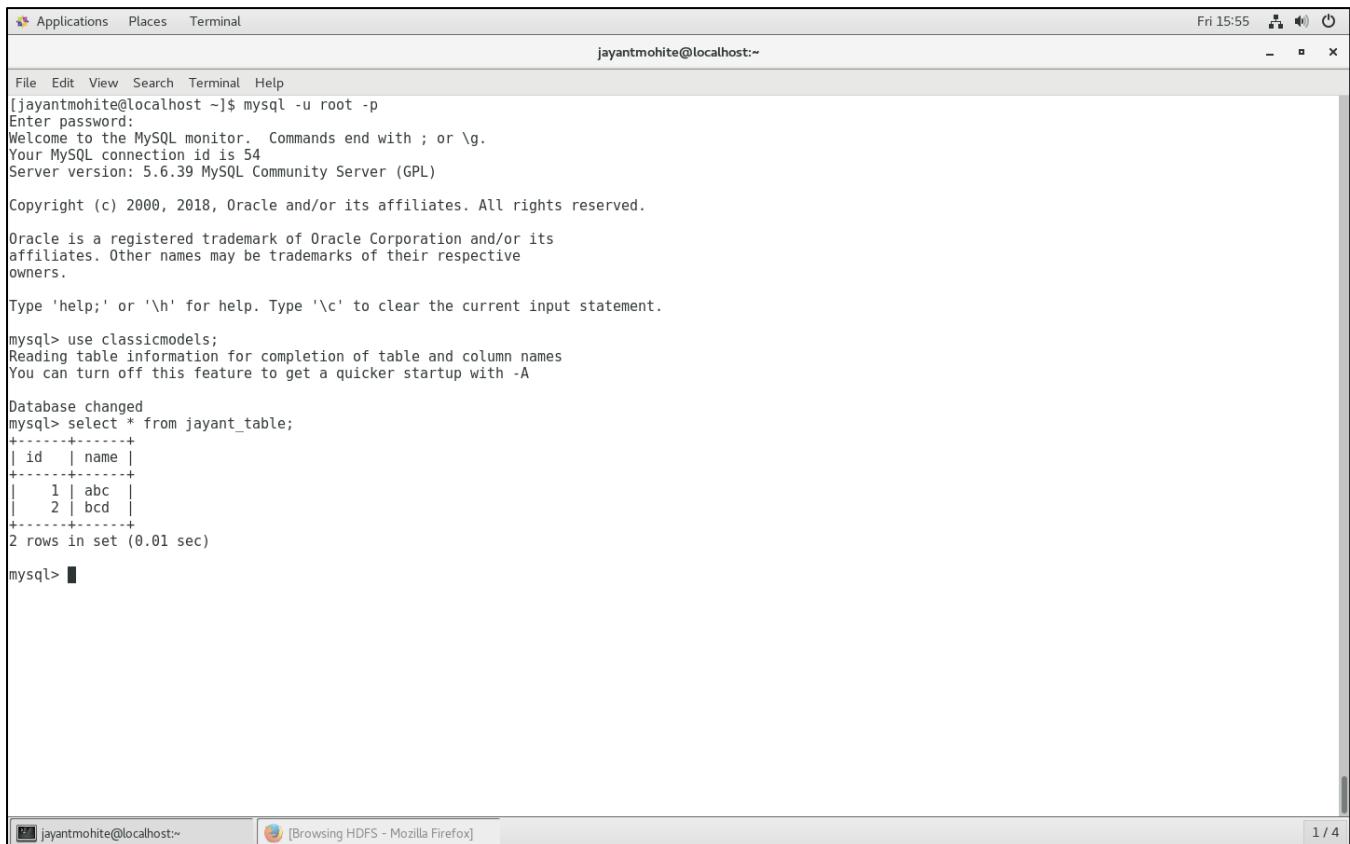
```
[jayantmohite@localhost] $ sqoop export --connect jdbc:mysql://localhost/classicmodels --table jayant_table --export-dir jayant_data --fields-terminated by '\t' -m 1 --username root -password root
```

Step Visualization 1:

```
[jayantmohite@localhost ~]$ sqoop export --connect jdbc:mysql://localhost/classicmodels --table jayant_table -m 1 --export-dir jayant_data --fields-terminated -by '\t' --username root --password root
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/03/02 15:53:30 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.14.0
18/03/02 15:53:30 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/03/02 15:53:30 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/03/02 15:53:30 INFO tool.CodeGenTool: Beginning code generation
18/03/02 15:53:31 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `jayant_table` AS t LIMIT 1
18/03/02 15:53:31 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `jayant_table` AS t LIMIT 1
18/03/02 15:53:31 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-0.20-mapreduce
Note: /tmp/sqoop-jayantmohite/compile/df01ed374de39db76d09f3348ecc1054/jayant_table.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/03/02 15:53:41 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-jayantmohite/compile/df01ed374de39db76d09f3348ecc1054/jayant_table.jar
18/03/02 15:53:41 INFO mapreduce.ExportJobBase: Beginning export of jayant_table
18/03/02 15:53:59 INFO input.FileInputFormat: Total input paths to process : 1
18/03/02 15:53:59 INFO input.FileInputFormat: Total input paths to process : 1
18/03/02 15:54:03 INFO mapred.JobClient: Running job: job_201803021449_0010
18/03/02 15:54:04 INFO mapred.JobClient: map 0% reduce 0%
```

Now login to your MySQL shell and verify the result

Step Visualization 2:



```
[jayantmohite@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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> use classicmodels;
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> select * from jayant_table;
+----+-----+
| id | name |
+----+-----+
| 1  | abc  |
| 2  | bcd  |
+----+-----+
2 rows in set (0.01 sec)

mysql> █
```

With this we complete the Sqoop Tutorial. Now you can go ahead and explore more options available and explained in the theory section of Sqoop.