

Experiment 7 - Demonstrate the use of Sqoop tool to transfer data between Hadoop & relational database servers

Sqoop:

The traditional application management system, that is, the interaction of applications with relational database using RDBMS, is one of the sources that generate Big Data. Such Big Data, generated by RDBMS, is stored in Relational Database Servers in the relational database structure.

When Big Data storages and analyzers such as MapReduce, Hive, HBase, Cassandra, Pig, etc. of the Hadoop ecosystem came into picture, they required a tool to interact with the relational database servers for importing and exporting the Big Data residing in them. Here, Sqoop occupies a place in the Hadoop ecosystem to provide feasible interaction between relational database server and Hadoop's HDFS.

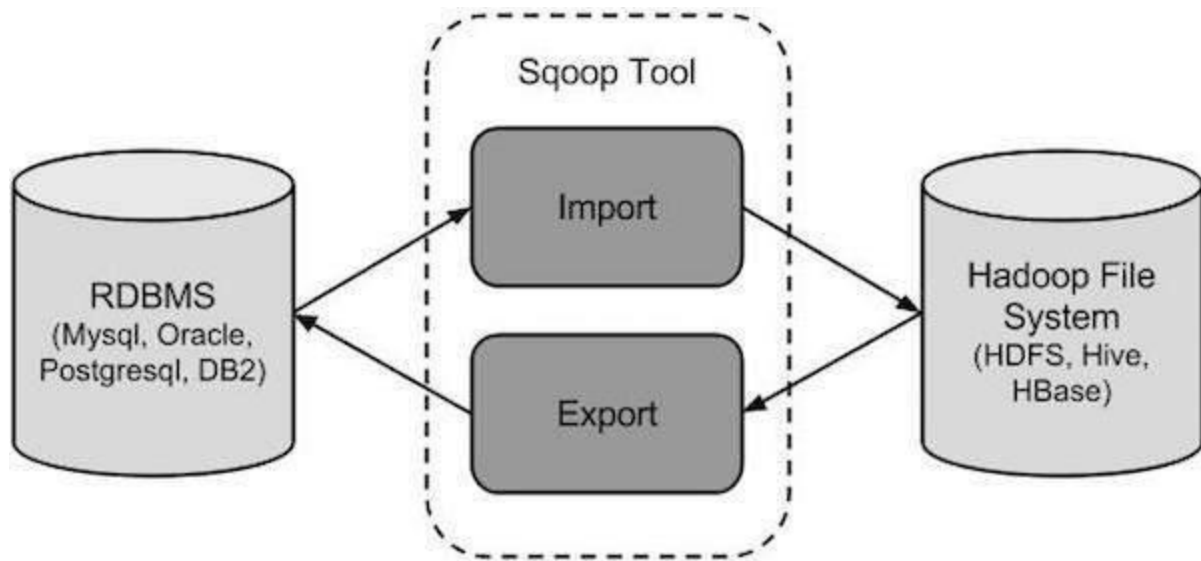
Sqoop – “SQL to Hadoop and Hadoop to SQL”

Sqoop is a tool designed to transfer data between Hadoop and external datastores such as relational databases and enterprise data warehouses.

It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases.

It imports data from external datastores into HDFS, Hive, and HBase.

Working of Sqoop:



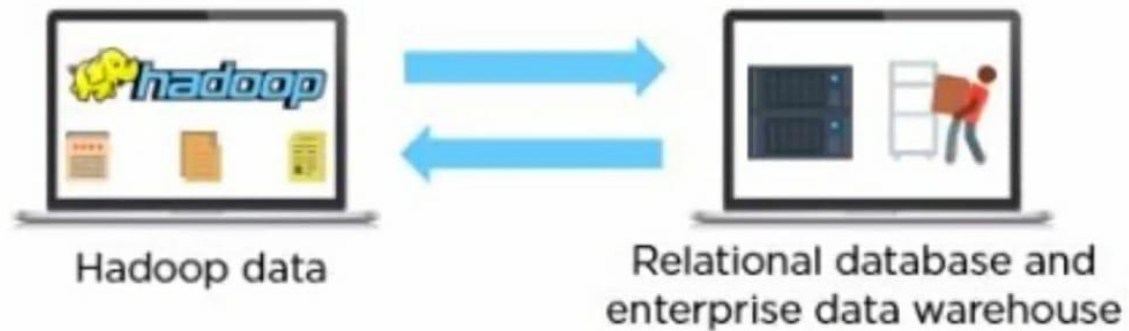
There are mainly two functions associated with Apache Sqoop tool which are Sqoop Import and Sqoop export.

1. Sqoop Import

Sqoop Import This is an important function which executes the task of data importing from external sources (RDBMS) to HDFS. In HDFS, each row of a table is considered as a record. The entire records are stored in a text format in the text files or as binary data in Sequence files.

2. Sqoop Export

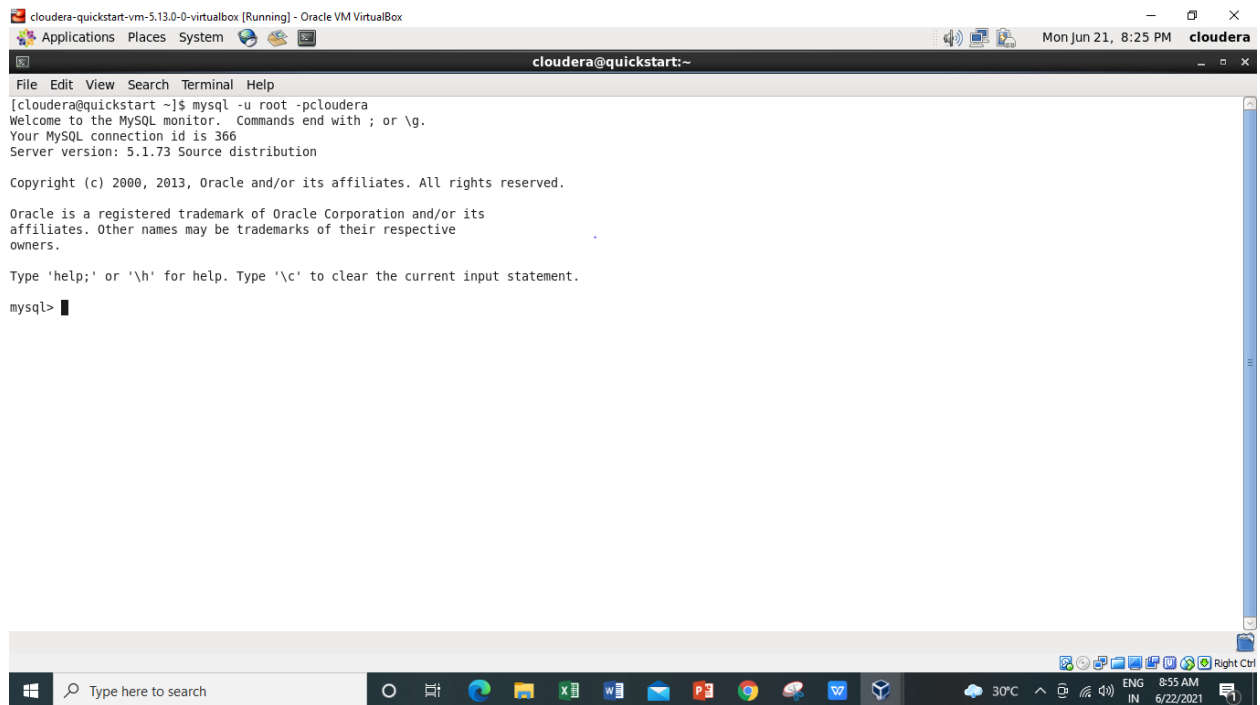
Sqoop Export This function performs bulk data exportation tasks from the HDFS to RDBMS. Once the modifications are done to the imported records you will get a result set and the next process is to send back the data to the relational database (RDBMS). Sqoop export function reads a group of delimited text files from HDFS in parallel, divides the files into records, and stores them as new rows in a targeted database table.



Steps: Demonstrate the use of Sqoop tool to transfer data between Hadoop & relational database servers

- 1) Starting the mysql by giving username as **root** and password as **cloudera**.

mysql -u root -pcloudera



2) Now using below command it will displaying or give the list of databases which are already present or exist in mysql.

show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cm |
| firehose |
| hue |
| metastore |
| mysql |
| nav |
| navms |
| oozie |
| retail_db |
| rman |
| sentry |
+-----+
12 rows in set (0.21 sec)

mysql>
```

3) Now we are using the existing database i.e. retail_db which are already present in mysql.

use retail_db;

```
mysql> use retail_db;
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> █
```

So right now we are under retail_db database.

4) Now to see the tables under a specific database so we will be using the same command which is used to display the databases.

show tables;

```
mysql> show tables;
+-----+
| Tables_in_retail_db |
+-----+
| categories           |
| customers             |
| departments          |
| order_items          |
| orders               |
| products              |
+-----+
6 rows in set (0.00 sec)

mysql> █
```

5) Here we are displaying all the records present in customers tables using below command.

select * from customers;

```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 12422 | Mary   | Blevins | XXXXXXXX | XXXXXXXX | 6473 Bright Expressway | Caguas | PR | 00725 |
| 12423 | Stephen | Smith   | XXXXXXXX | XXXXXXXX | 3445 Harvest Campus   | Palmdale | CA | 93550 |
| 12424 | Judy   | Phillips | XXXXXXXX | XXXXXXXX | 4534 Cinder Concession | San Diego | CA | 92111 |
| 12425 | Mary   | Smith   | XXXXXXXX | XXXXXXXX | 1050 Grand Forest Towers | Caguas | PR | 00725 |
| 12426 | Jordan | Valdez  | XXXXXXXX | XXXXXXXX | 5561 Quiet Loop       | Brooklyn | NY | 11210 |
| 12427 | Mary   | Smith   | XXXXXXXX | XXXXXXXX | 3662 Round Barn Gate   | Plano | TX | 75093 |
| 12428 | Jeffrey | Travis  | XXXXXXXX | XXXXXXXX | 1552 Burning Dale Highlands | Caguas | PR | 00725 |
| 12429 | Mary   | Smith   | XXXXXXXX | XXXXXXXX | 92 Sunny Bear Villas   | Gardena | CA | 90247 |
| 12430 | Hannah | Brown   | XXXXXXXX | XXXXXXXX | 8316 Pleasant Bend    | Caguas | PR | 00725 |
| 12431 | Mary   | Rios    | XXXXXXXX | XXXXXXXX | 1221 Cinder Pines     | Kaneohe | HI | 96744 |
| 12432 | Angela | Smith   | XXXXXXXX | XXXXXXXX | 1525 Jagged Barn Highlands | Caguas | PR | 00725 |
| 12433 | Benjamin | Garcia | XXXXXXXX | XXXXXXXX | 5459 Noble Brook Landing | Levittown | NY | 11756 |
| 12434 | Mary   | Mills   | XXXXXXXX | XXXXXXXX | 9720 Colonial Parade   | Caguas | PR | 00725 |
| 12435 | Laura  | Horton  | XXXXXXXX | XXXXXXXX | 5736 Honey Downs      | Summerville | SC | 29483 |
+-----+-----+-----+-----+-----+-----+-----+-----+
12435 rows in set (0.26 sec)

mysql>
```

As it is a huge table. It contains total 12435 rows or records.

6) Let see any other table. Here we are displaying department table it has 6 rows in it.

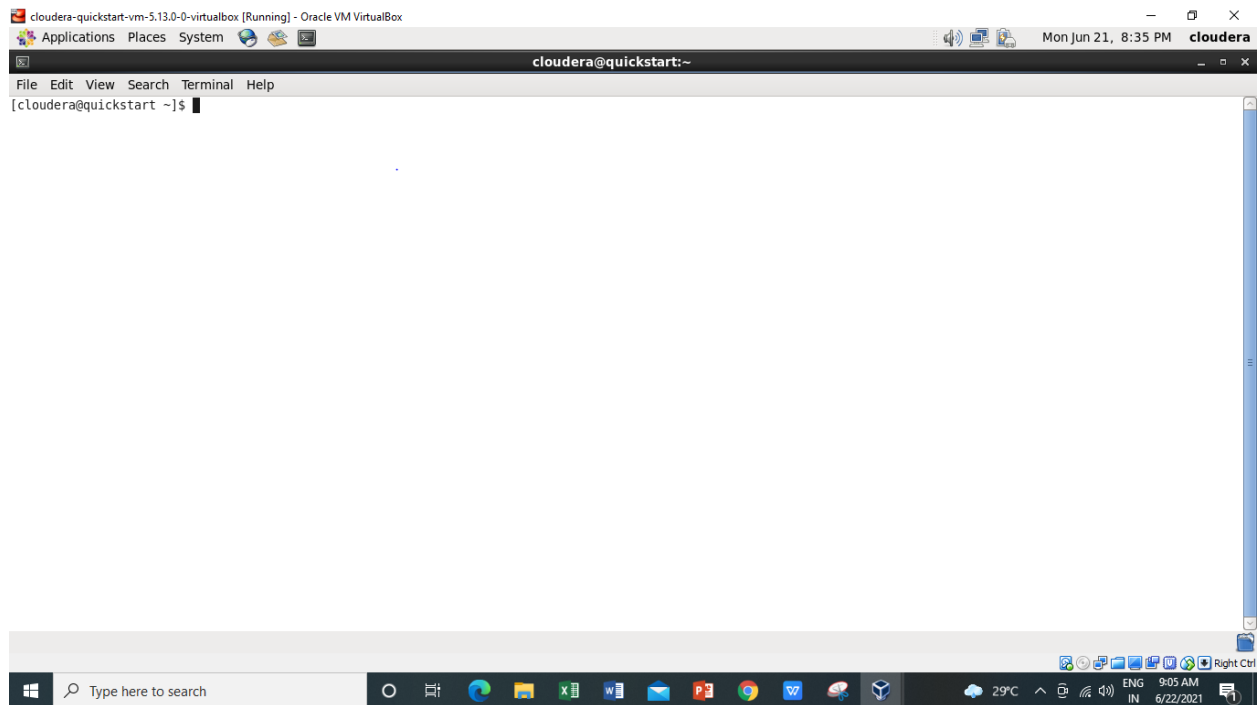
select * from departments;

```
mysql> select * from departments;
+-----+-----+
| department_id | department_name |
+-----+-----+
| 2 | Fitness |
| 3 | Footwear |
| 4 | Apparel |
| 5 | Golf |
| 6 | Outdoors |
| 7 | Fan Shop |
+-----+-----+
6 rows in set (0.07 sec)
```

```
mysql>
```

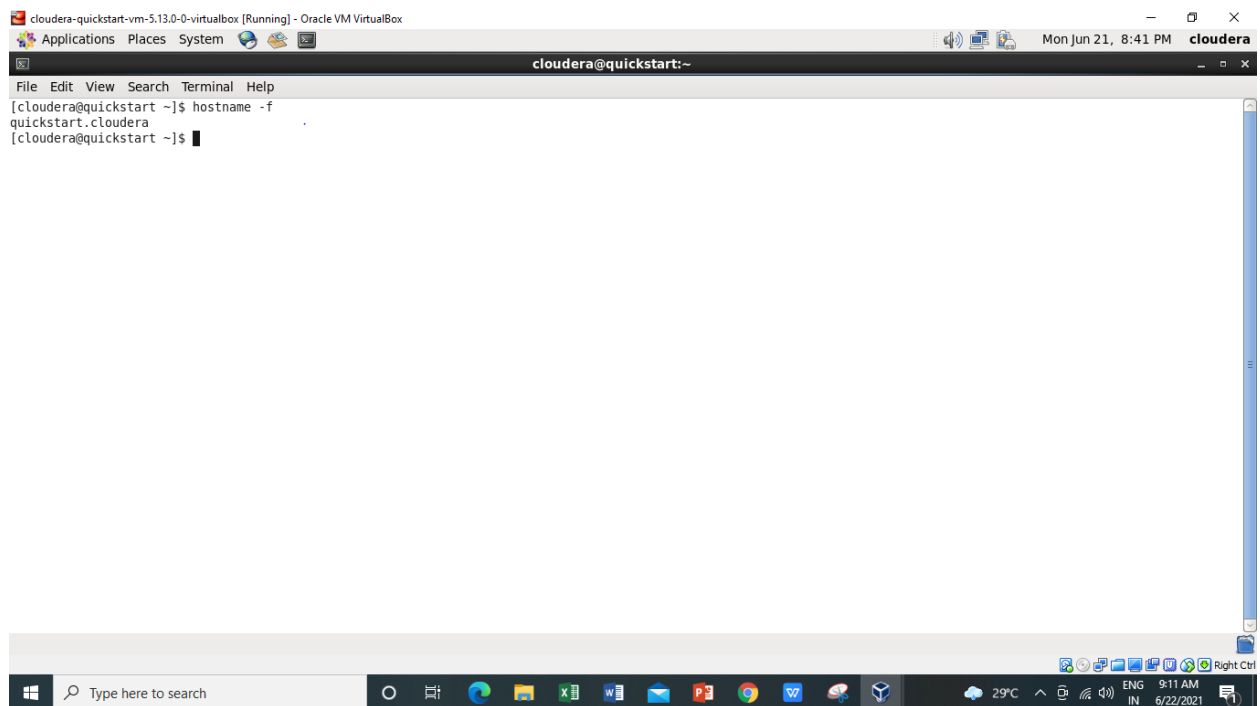
These are the different departments i.e. department_name with their respective department_id.

7) Open the new terminal for running command for sqoop.



8) We will require hostname for this sqoop .

hostname -f

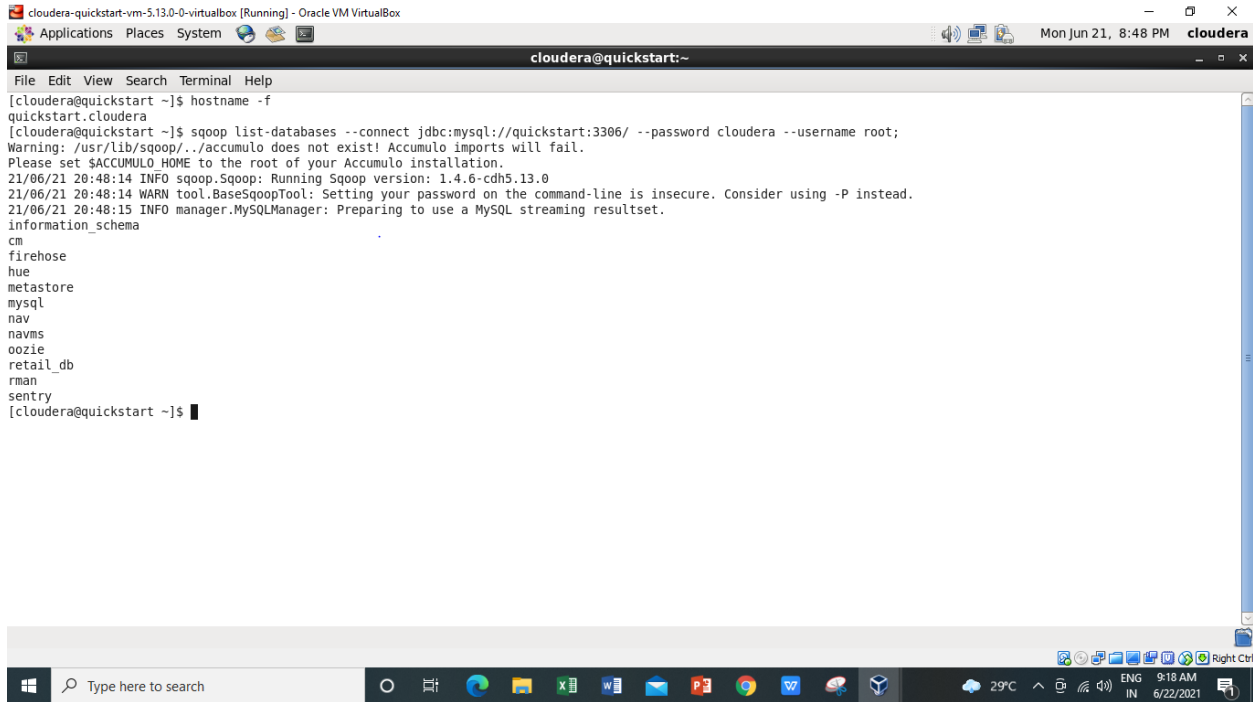


So it is giving as **quickstart.cloudera** as the name of the host that we are already connected to.

9) Then if we want to list down all the databases we will use below sqoop command.

sqoop list-databases --connect jdbc:mysql://quickstart:3306/ --password cloudera --username root;

So we have studied in this sqoop that we can make use of jdbc or the odbc type driver. So the applications that support the jdbc will be connecting them with the jdbc driver. So here we are using mysql which we are going to connect this with the jdbc . mysql running on **quickstart:3306** then we have to mention password which is **cloudera** and username i.e. **root**.



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hostname -f
quickstart.cloudera
[cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://quickstart:3306/ --password cloudera --username root;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/06/21 20:48:14 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/06/21 20:48:14 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/06/21 20:48:15 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
oozie
retail_db
rman
sentry
[cloudera@quickstart ~]$
```

These are the lists of same databases which are present in mysql and the information also present here. So we are connecting the sqoop with mysql with the help of jdbc.

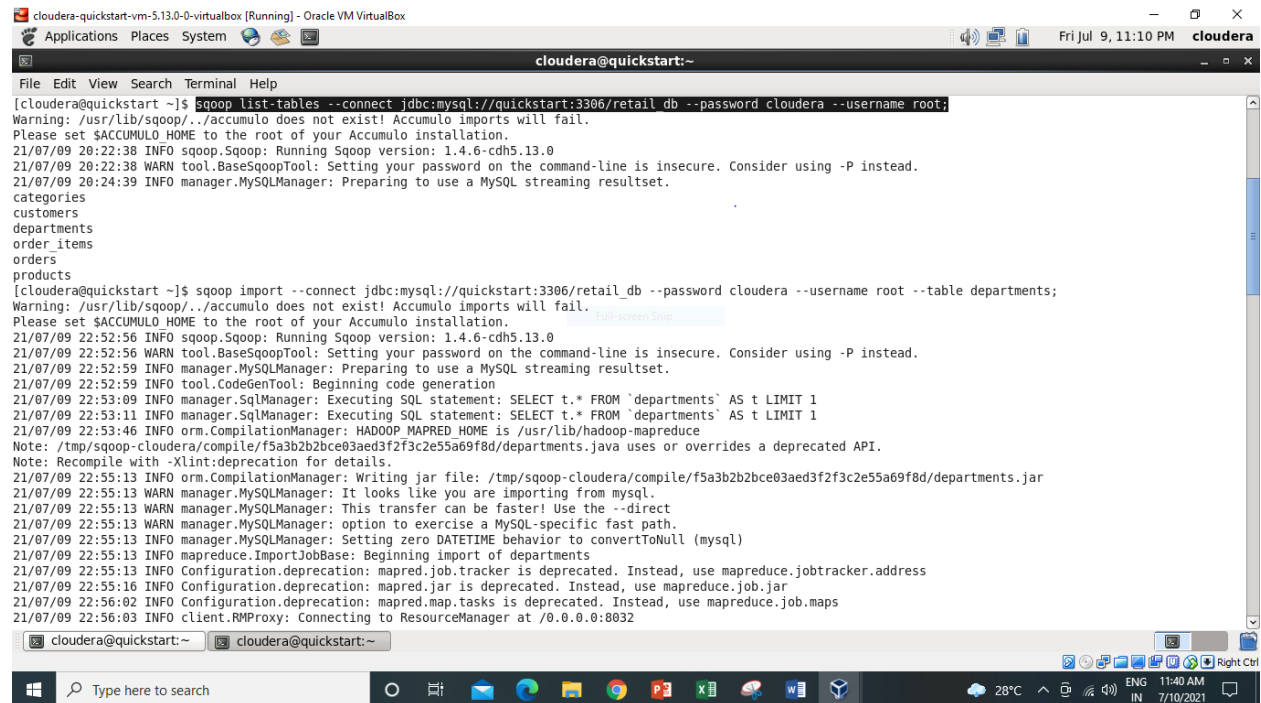
10) Now we will list out all the tables using below command.

sqoop list-tables --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root;

```
[cloudera@quickstart ~]$ sqoop list-tables --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/06/21 20:54:08 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/06/21 20:54:08 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/06/21 20:54:09 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
categories
customers
departments
order_items
orders
products
[cloudera@quickstart ~]$
```


11) Now we will start with the import and export tools of the Hadoop. We want to Import table “departments” from retail_db database which are present inside in mysql.

sqoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments;



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ sqoop list-tables --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/07/09 20:22:38 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/07/09 20:22:38 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/07/09 20:24:39 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
categories
customers
departments
order items
orders
products
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/07/09 22:52:56 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/07/09 22:52:56 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/07/09 22:52:59 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/07/09 22:52:59 INFO tool.CodeGenTool: Beginning code generation
21/07/09 22:53:09 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/09 22:53:11 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/09 22:53:46 INFO orm.CompilationManager: HADOOP MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/f5a3b2b2bce03aed3f2f3c2e55a69f8d/departments.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/07/09 22:55:13 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/f5a3b2b2bce03aed3f2f3c2e55a69f8d/departments.jar
21/07/09 22:55:13 WARN manager.MySQLManager: It looks like you are importing from mysql.
21/07/09 22:55:13 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
21/07/09 22:55:13 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
21/07/09 22:55:13 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
21/07/09 22:55:13 INFO mapreduce.ImportJobBase: Beginning import of departments
21/07/09 22:55:13 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/07/09 22:55:16 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/07/09 22:56:02 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/07/09 22:56:03 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
```

```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
FILE: Number of write operations=0
HDFS: Number of bytes read=481
HDFS: Number of bytes written=60
HDFS: Number of read operations=16
HDFS: Number of large read operations=0
HDFS: Number of write operations=8
Job Counters
  Launched map tasks=4
  Other local map tasks=4
  Total time spent by all maps in occupied slots (ms)=2132584
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=2132584
  Total vcore-milliseconds taken by all map tasks=2132584
  Total megabyte-milliseconds taken by all map tasks=2183766016
Map-Reduce Framework
  Map input records=6
  Map output records=6
  Input split bytes=481
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=11679
  CPU time spent (ms)=13080
  Physical memory (bytes) snapshot=415117312
  Virtual memory (bytes) snapshot=6040686592
  Total committed heap usage (bytes)=96206848
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=60
21/07/09 23:07:29 INFO mapred.ClientServiceDelegate: Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
21/07/09 23:08:08 INFO mapreduce.ImportJobBase: Transferred 60 bytes in 725.5924 seconds (0.0827 bytes/sec)
21/07/09 23:08:08 INFO mapreduce.ImportJobBase: Retrieved 6 records.
[cloudera@quickstart ~]$
```

12) Now we will see whether all departments table successfully imported from mysql in Hadoop (hdfs) or not using below command.

hadoop fs -ls

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 3 items
drwxr-xr-x - cloudera cloudera 0 2021-07-01 10:13 Training
drwxr-xr-x - cloudera cloudera 0 2021-07-09 23:07 departments
-rw-r--r-- 1 cloudera supergroup 100 2021-06-05 12:49 output_new
[cloudera@quickstart ~]$
```

Departments table is now successfully imported in hdfs.

13) Now we will see what inside this department using below command.

hadoop fs -ls departments;

```
[cloudera@quickstart ~]$ hadoop fs -ls departments;
Found 5 items
-rw-r--r-- 1 cloudera cloudera 0 2021-07-09 23:07 departments/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 21 2021-07-09 23:06 departments/part-m-00000
-rw-r--r-- 1 cloudera cloudera 10 2021-07-09 23:06 departments/part-m-00001
-rw-r--r-- 1 cloudera cloudera 7 2021-07-09 23:06 departments/part-m-00002
-rw-r--r-- 1 cloudera cloudera 22 2021-07-09 23:06 departments/part-m-00003
[cloudera@quickstart ~]$
```

As we can see there are once SUCCESS file and four part-m files which has got the output present inside the departments.

14) Now we will see what there inside this some of the part m files so that can be done with the help of below commands.

hadoop fs -cat /user/cloudera/departments/part-m-00000

hadoop fs -cat /user/cloudera/departments/part-m-00001

hadoop fs -cat /user/cloudera/departments/part-m-00002

hadoop fs -cat /user/cloudera/departments/part-m-00003

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part-m-00000
2,Fitness
3,Footwear
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part-m-00001
4,Apparel
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part-m-00002
5,Golf
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part-m-00003
6,Outdoors
7,Fan Shop
[cloudera@quickstart ~]$
```



15) If want to display output of all part-m files together we will use below command.

hadoop fs -cat /user/cloudera/departments/part*

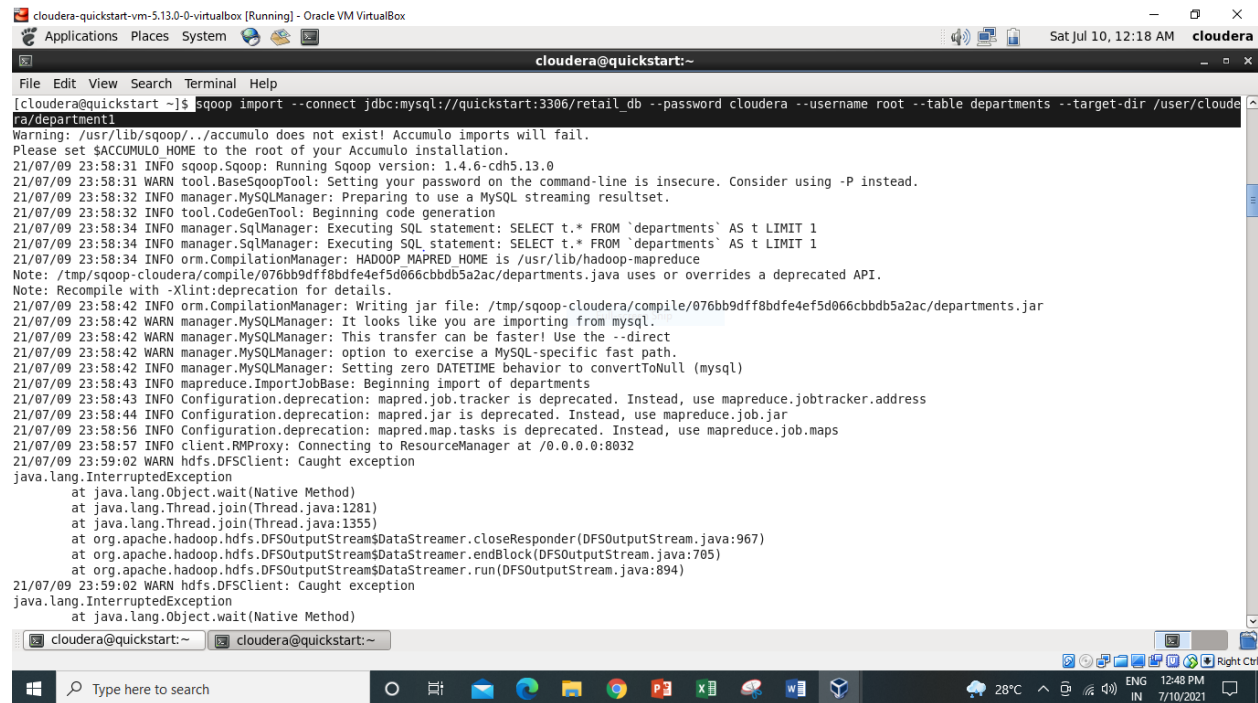
```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part*
2,Fitness
3,Footwear
4,Apparel
5,Golf
6,Outdoors
7,Fan Shop
[cloudera@quickstart ~]$
```



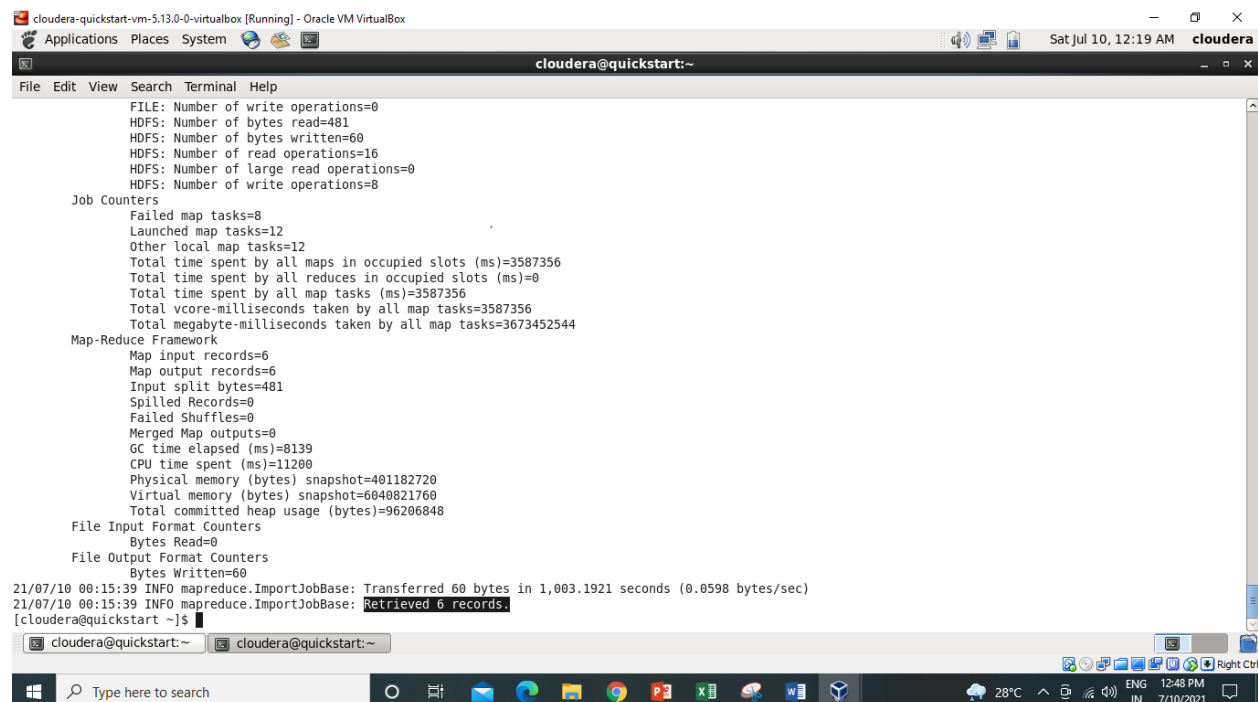
16) If we want to mention that where should we have our this output in the hdfs so for that we have to mention the target directory.

We will want to import my department table in --target directory as department1.

scoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments --target-dir /user/cloudera/department1



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~$ scoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments --target-dir /user/cloudera/department1
Warning: /usr/lib/scoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/07/09 23:58:31 INFO scoop.Scoop: Running Scoop version: 1.4.6-cdh5.13.0
21/07/09 23:58:31 WARN tool.BaseScoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/07/09 23:58:32 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/07/09 23:58:32 INFO tool.CodeGenTool: Beginning code generation
21/07/09 23:58:34 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/09 23:58:34 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/09 23:58:34 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/scoop-cloudera/compile/076bb9dff8bdf4ef5d066cbbdb5a2ac/departments.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/07/09 23:58:42 INFO orm.CompilationManager: Writing jar file: /tmp/scoop-cloudera/compile/076bb9dff8bdf4ef5d066cbbdb5a2ac/departments.jar
21/07/09 23:58:42 WARN manager.MySQLManager: It looks like you are importing from mysql.
21/07/09 23:58:42 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
21/07/09 23:58:42 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
21/07/09 23:58:42 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
21/07/09 23:58:43 INFO mapreduce.ImportJobBase: Beginning import of departments
21/07/09 23:58:43 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/07/09 23:58:44 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/07/09 23:58:56 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/07/09 23:58:57 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/07/09 23:59:02 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/07/09 23:59:02 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
```



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~$ scoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments --target-dir /user/cloudera/department1
FILE: Number of write operations=0
HDFS: Number of bytes read=481
HDFS: Number of bytes written=60
HDFS: Number of read operations=16
HDFS: Number of large read operations=0
HDFS: Number of write operations=8
Job Counters
  Failed map tasks=8
  Launched map tasks=12
  Other local map tasks=12
  Total time spent by all maps in occupied slots (ms)=3587356
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=3587356
  Total vcore-milliseconds taken by all map tasks=3587356
  Total megabyte-milliseconds taken by all map tasks=3673452544
Map-Reduce Framework
  Map input records=6
  Map output records=6
  Input split bytes=481
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=8139
  CPU time spent (ms)=11200
  Physical memory (bytes) snapshot=401182720
  Virtual memory (bytes) snapshot=6040821760
  Total committed heap usage (bytes)=96206848
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=60
21/07/10 00:15:39 INFO mapreduce.ImportJobBase: Transferred 60 bytes in 1,003.1921 seconds (0.0598 bytes/sec)
21/07/10 00:15:39 INFO mapreduce.ImportJobBase: Retrieved 6 records.
cloudera@quickstart:~$
```

As we can see 6 records are retrieved successfully.

17) Now let's check it using below command.

hadoop fs -ls

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 4 items
drwxr-xr-x - cloudera cloudera      0 2021-07-01 10:13 Training
drwxr-xr-x - cloudera cloudera      0 2021-07-10 00:15 department1
drwxr-xr-x - cloudera cloudera      0 2021-07-09 23:07 departments
-rw-r--r--  1 cloudera supergroup 100 2021-06-05 12:49 output_new
[cloudera@quickstart ~]$
```

So here we have department1 directory.

18) Now we will check what is present inside this department1 directory using below command.

hadoop fs -ls /user/cloudera/department1

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/department1
Found 5 items
-rw-r--r--  1 cloudera cloudera      0 2021-07-10 00:15 /user/cloudera/department1/_SUCCESS
-rw-r--r--  1 cloudera cloudera    21 2021-07-10 00:15 /user/cloudera/department1/part-m-00000
-rw-r--r--  1 cloudera cloudera    10 2021-07-10 00:15 /user/cloudera/department1/part-m-00001
-rw-r--r--  1 cloudera cloudera      7 2021-07-10 00:15 /user/cloudera/department1/part-m-00002
-rw-r--r--  1 cloudera cloudera    22 2021-07-10 00:15 /user/cloudera/department1/part-m-00003
[cloudera@quickstart ~]$
```

So it has different part-m files.

19) Now we will read the content of these part-m files using below command.

hadoop fs -cat /user/cloudera/department1/part*

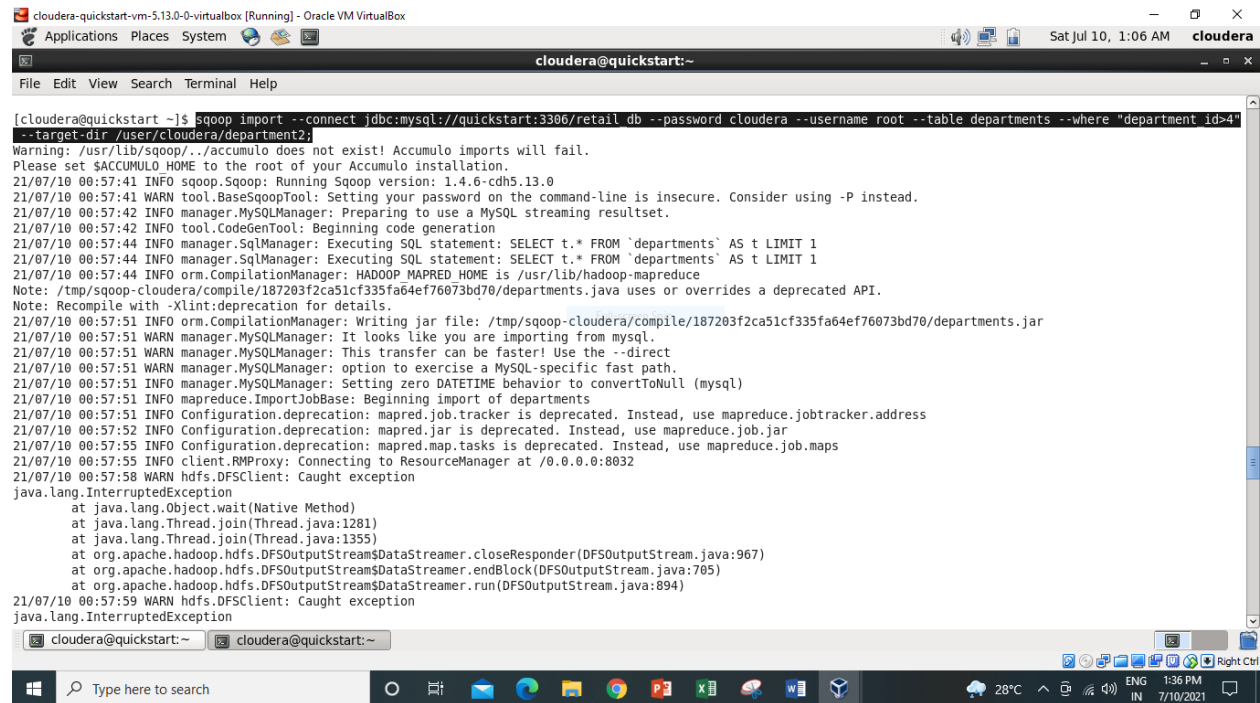
```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/department1/part*
2,Fitness
3,Footwear
4,Apparel
5,Golf
6,Outdoors
7,Fan Shop
[cloudera@quickstart ~]$
```

So we have got the output.

20) Now we will filter out some or specific rows only from the departments table and have it in hdfs but before we will apply some conditions on the rows of the departments table and whichever rows will satisfy the condition only those rows are would be stored in the hdfs.

We want to fetch only those departments where department_id is greater than 4.

sqoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments --where "department_id>4" --target-dir /user/cloudera/department2;



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help

[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table departments --where "department_id>4" --target-dir /user/cloudera/department2;
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/07/10 00:57:41 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/07/10 00:57:41 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/07/10 00:57:42 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/07/10 00:57:42 INFO tool.CodeGenTool: Beginning code generation
21/07/10 00:57:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/10 00:57:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `departments` AS t LIMIT 1
21/07/10 00:57:44 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/187203f2ca51cf335fa64ef76073bd70/departments.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/07/10 00:57:51 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/187203f2ca51cf335fa64ef76073bd70/departments.jar
21/07/10 00:57:51 WARN manager.MySQLManager: It looks like you are importing from mysql.
21/07/10 00:57:51 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
21/07/10 00:57:51 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
21/07/10 00:57:51 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
21/07/10 00:57:51 INFO mapreduce.ImportJobBase: Beginning import of departments
21/07/10 00:57:51 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/07/10 00:57:52 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/07/10 00:57:55 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/07/10 00:57:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/07/10 00:57:58 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/07/10 00:57:59 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
cloudera@quickstart:~ cloudera@quickstart:~
```

```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=361
HDFS: Number of bytes written=29
HDFS: Number of read operations=12
HDFS: Number of large read operations=0
HDFS: Number of write operations=6
Job Counters
  Launched map tasks=3
  Other local map tasks=3
  Total time spent by all maps in occupied slots (ms)=870405
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=870405
  Total vcore-milliseconds taken by all map tasks=870405
  Total megabyte-milliseconds taken by all map tasks=891294720
Map-Reduce Framework
  Map input records=3
  Map output records=3
  Input split bytes=361
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=135983
  CPU time spent (ms)=5240
  Physical memory (bytes) snapshot=316547072
  Virtual memory (bytes) snapshot=4530503680
  Total committed heap usage (bytes)=72155136
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=29
21/07/10 01:03:58 INFO mapreduce.ImportJobBase: Transferred 29 bytes in 363.2502 seconds (0.0798 bytes/sec)
21/07/10 01:03:58 INFO mapreduce.ImportJobBase: Retrieved 3 records.
[cloudera@quickstart ~]$
```

As we can see 3 records are retrieved successfully.

21) Now we will check it using below command.

hadoop fs -ls /user/cloudera/department2

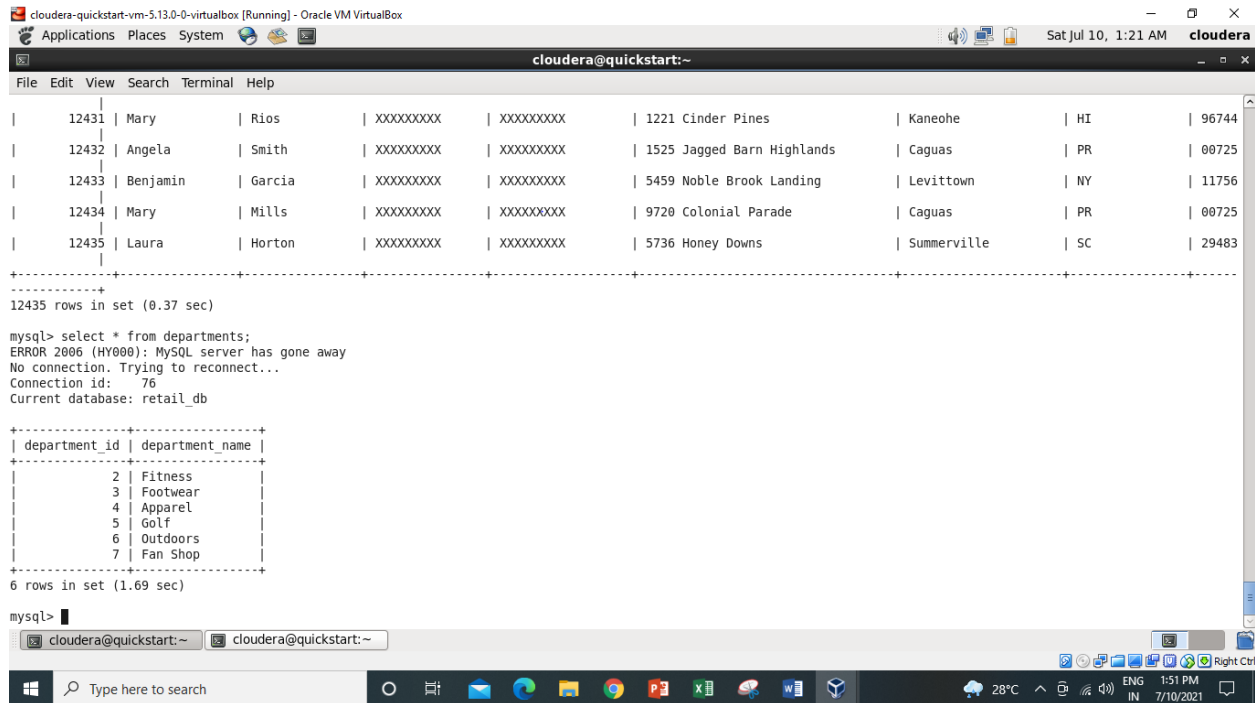
```
[cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/department2
Found 4 items
-rw-r--r-- 1 cloudera cloudera 0 2021-07-10 01:03 /user/cloudera/department2/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 7 2021-07-10 01:03 /user/cloudera/department2/part-m-00000
-rw-r--r-- 1 cloudera cloudera 11 2021-07-10 01:03 /user/cloudera/department2/part-m-00001
-rw-r--r-- 1 cloudera cloudera 11 2021-07-10 01:03 /user/cloudera/department2/part-m-00002
[cloudera@quickstart ~]$
```

22) Now will read the content of these part-m files using cat command.

hadoop fs -cat /user/cloudera/department2/part*

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/department2/part*
5,Golf
6,Outdoors
7,Fan Shop
[cloudera@quickstart ~]$
```


23) Now we will see the Export command. So what the export tool does is it will export the data from our hdfs to the RDBMS. So for that we need to have some table in mysql with some records so for that we will now move to mysql.



The screenshot shows a terminal window with a MySQL query result and an error message. The query result is a table with 10 columns and 5 rows. The error message is "ERROR 2006 (HY000): MySQL server has gone away".

```
mysql> select * from departments;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 76
Current database: retail_db
```

department_id	department_name
2	Fitness
3	Footwear
4	Apparel
5	Golf
6	Outdoors
7	Fan Shop

```
mysql>
```

24) So here we will create the table “dpt” and it will be having two attributes as “department_id” and “ department_name”.

create table dpt(department_id int not null auto_increment, department_name varchar(50) not null, primary key(department_id));

```
mysql> create table dpt(department_id int not null auto_increment, department_name varchar(50) not null, primary key(department_id));
Query OK, 0 rows affected (0.53 sec)
```

```
mysql>
```

So here we can see the “dpt” table created successfully.

25) Now we want to check what we have inside this dpt table.

Select * from dpt;

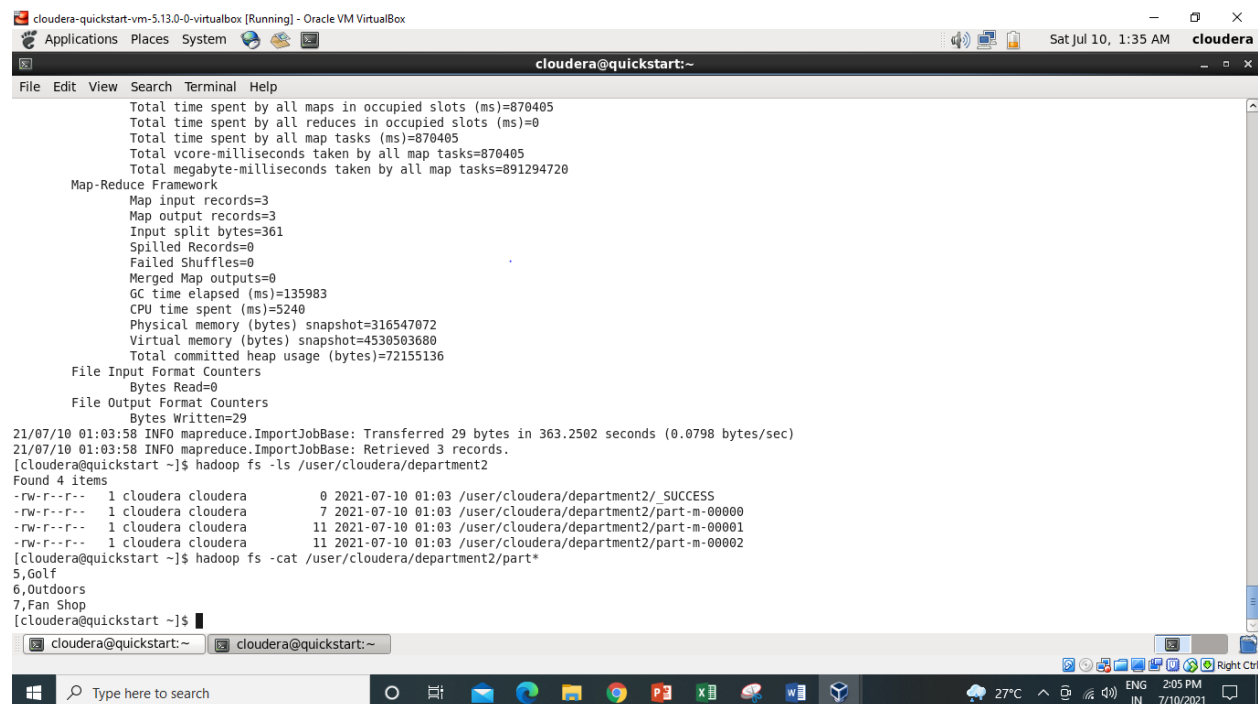

```
mysql> Select * from dpt;  
Empty set (0.06 sec)
```

```
mysql> █
```



As we have not inserted any records inside the dpt table so that's why it is showing as Empty set.

26) Now we will exporting the data from the hdfs to dpt table of mysql. Now we will move to the sqoop terminal.



27) So now we are performing export operation using below command.

we are trying to export department2 which are present in cloudera to inside our dpt table which are present inside mysql.

sqoop export --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table dpt --export-dir /user/cloudera/department2;

```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root --table dpt --export-dir /user/cloudera/department2:
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/07/10 02:07:59 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/07/10 02:07:59 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/07/10 02:08:01 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/07/10 02:08:01 INFO tool.CodeGenTool: Beginning code generation
21/07/10 02:08:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `dpt` AS t LIMIT 1
21/07/10 02:08:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `dpt` AS t LIMIT 1
21/07/10 02:08:03 INFO orm.CompilationManager: HADOOP MAPRED HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/6266b4cf37f94def4bc637115177aca/dpt.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/07/10 02:08:10 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/6266b4cf37f94def4bc637115177aca/dpt.jar
21/07/10 02:08:10 INFO mapreduce.ExportJobBase: Beginning export of dpt
21/07/10 02:08:10 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/07/10 02:08:12 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/07/10 02:08:16 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
21/07/10 02:08:16 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.map.speculative
21/07/10 02:08:16 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/07/10 02:08:17 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/07/10 02:08:19 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedExce...
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/07/10 02:08:20 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedExce...
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
```

```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=683
HDFS: Number of bytes written=0
HDFS: Number of read operations=18
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Job Counters
  Launched map tasks=3
  Data-Local map tasks=3
  Total time spent by all maps in occupied slots (ms)=204593
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=204593
  Total vcore-milliseconds taken by all map tasks=204593
  Total megabyte-milliseconds taken by all map tasks=209503232
Map-Reduce Framework
  Map input records=3
  Map output records=3
  Input split bytes=627
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=2051
  CPU time spent (ms)=5300
  Physical memory (bytes) snapshot=304099328
  Virtual memory (bytes) snapshot=4524187648
  Total committed heap usage (bytes)=72155136
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=0
21/07/10 02:10:11 INFO mapreduce.ExportJobBase: Transferred 683 bytes in 114.3907 seconds (5.9708 bytes/sec)
21/07/10 02:10:11 INFO mapreduce.ExportJobBase: Exported 3 records.
[cloudera@quickstart ~]$
```

Now we have successfully exported 3 records.

28) Now we will see whether the records are successfully exported in dpt table which are present inside mysql using below command.

Select * from dpt;

```
mysql> select * from dpt;
```

department_id	department_name
5	Golf
6	Outdoors
7	Fan Shop

3 rows in set (0.04 sec)

```
mysql> █
```



As we can see these 3 records which are present in department2 table are successfully exported inside the dpt table of mysql.