# 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.

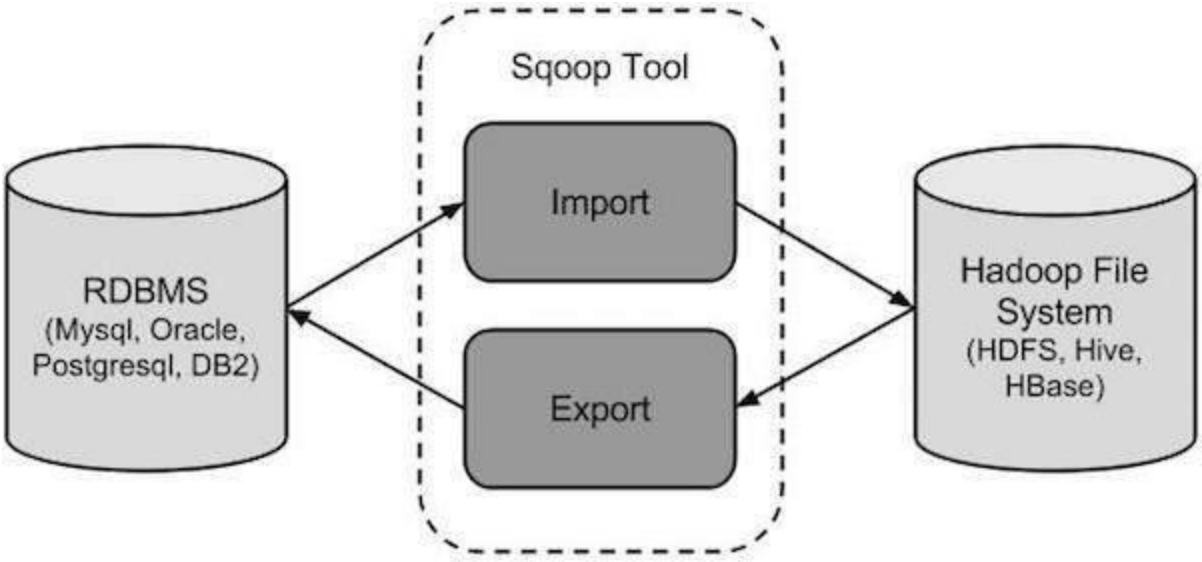
Sqoop − "SQL to Hadoop and Hadoop to SQL"

Sqoop is a tool designed to transfer data between Hadoop and external datastores suac 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:



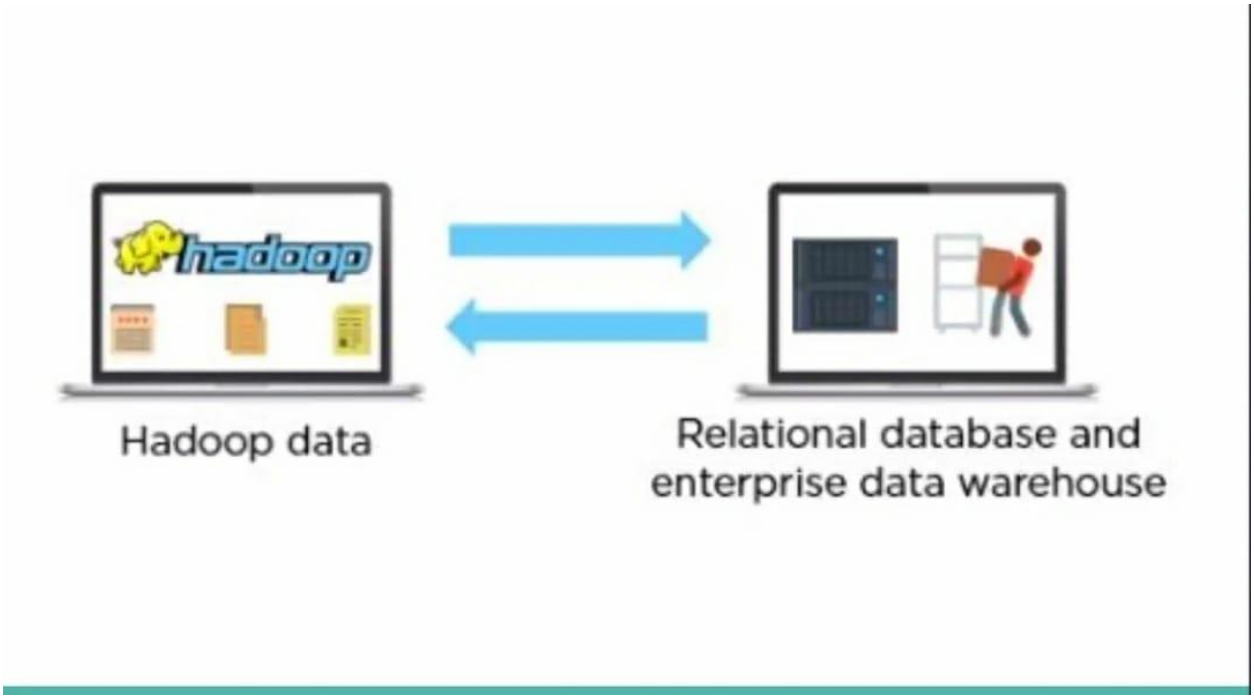

There are mainly two functions associated with Apache Sqoop tool which are Sqoop Import and Sqoop export.

## 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

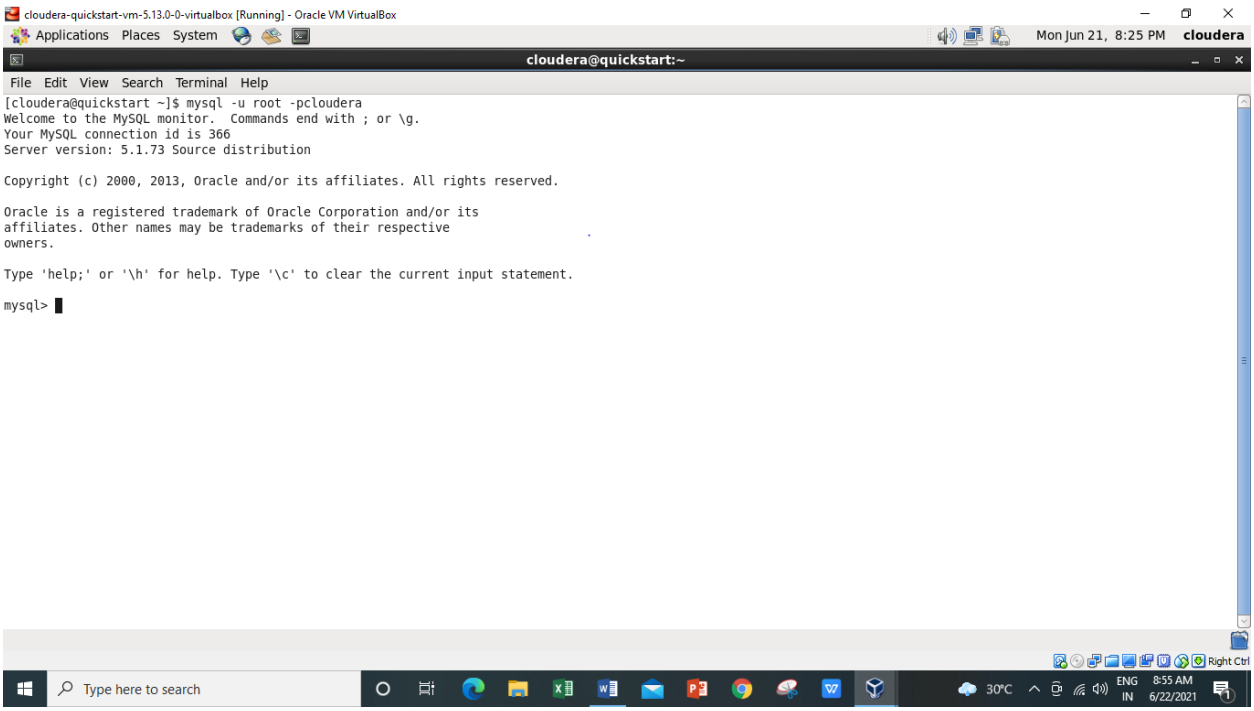
**Sqoop Export** This function performs bulk data exportation tasks from the HDFS to RDBMS. Once the modifications are done to the imported records you will get a result set and the next process is to send back the data to the relational database (RDBMS). Sqoop export function reads a group of delimited 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.

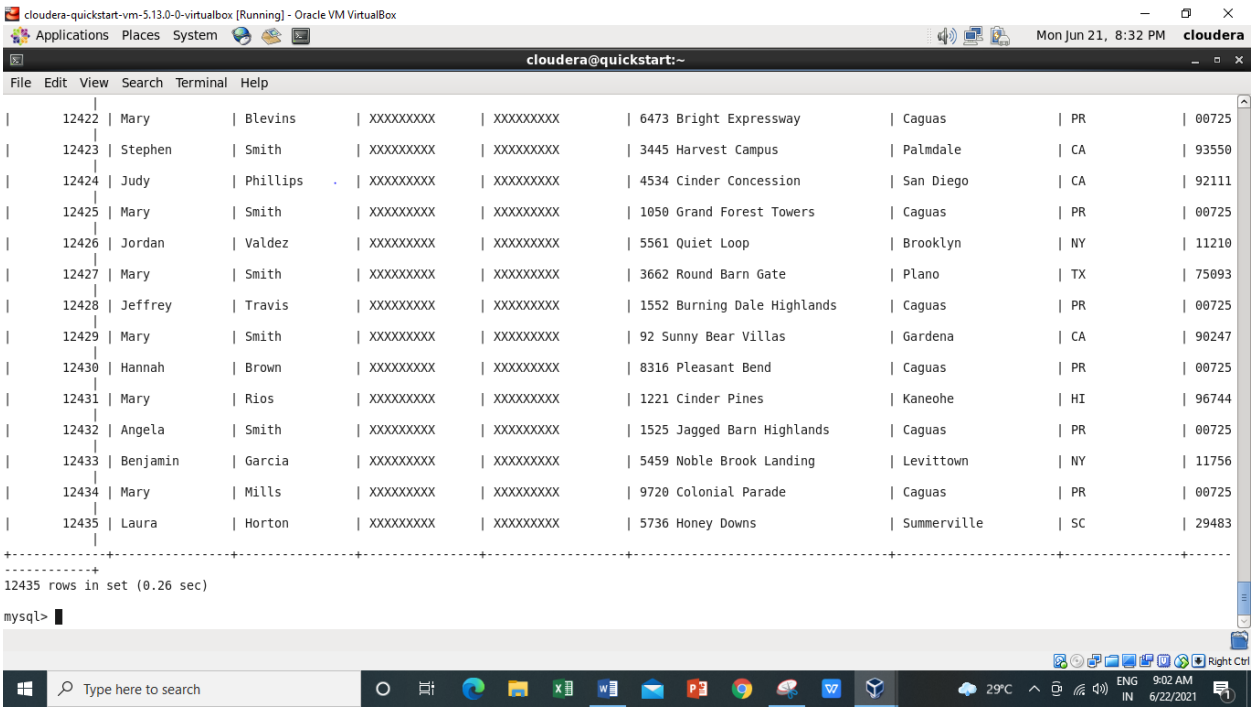
**show tables;**

```
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;**




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.

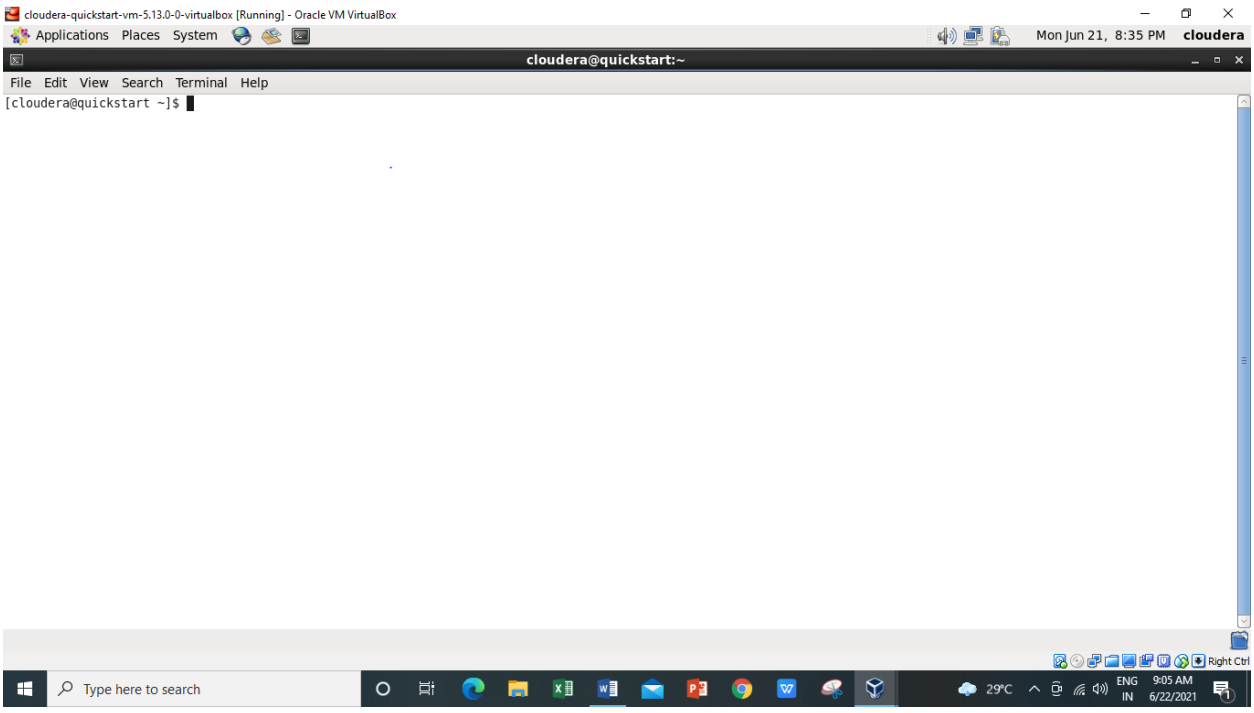
**select * from departments;**

```
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>
```

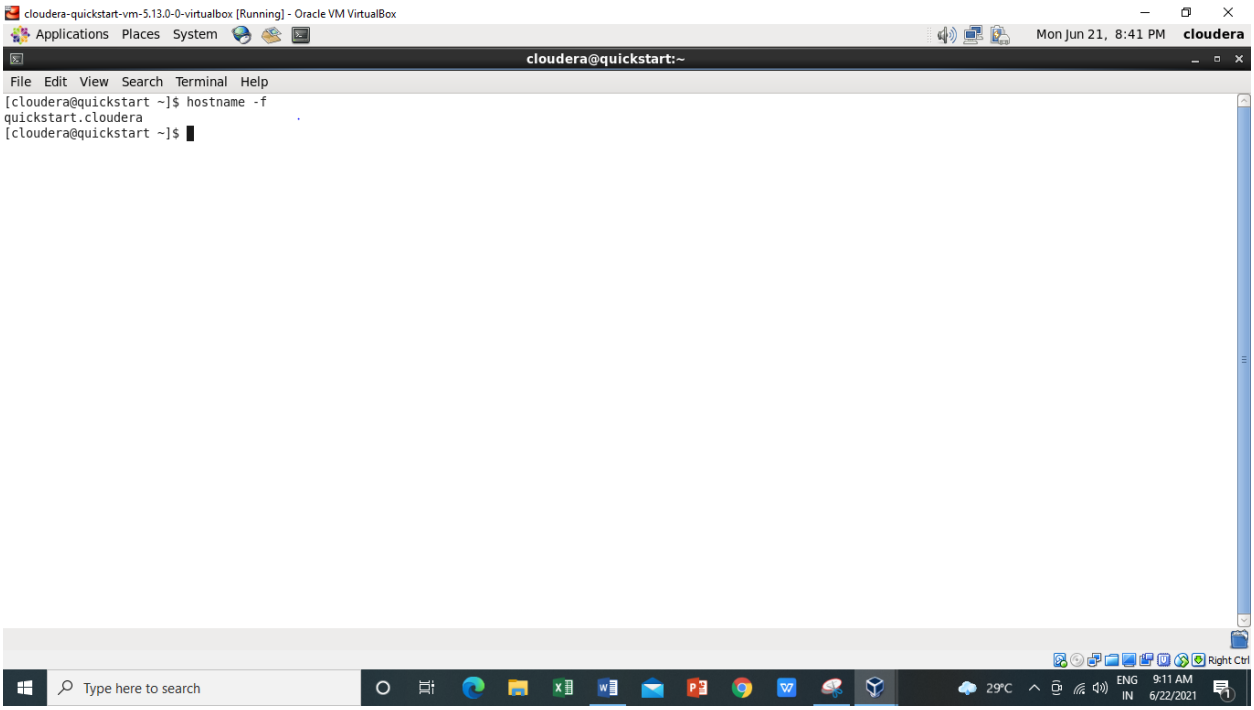
These are the different departments i.e. department_name with their respective department_id.

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




8) We will require hostname for this sqoop .

**hostname –f**

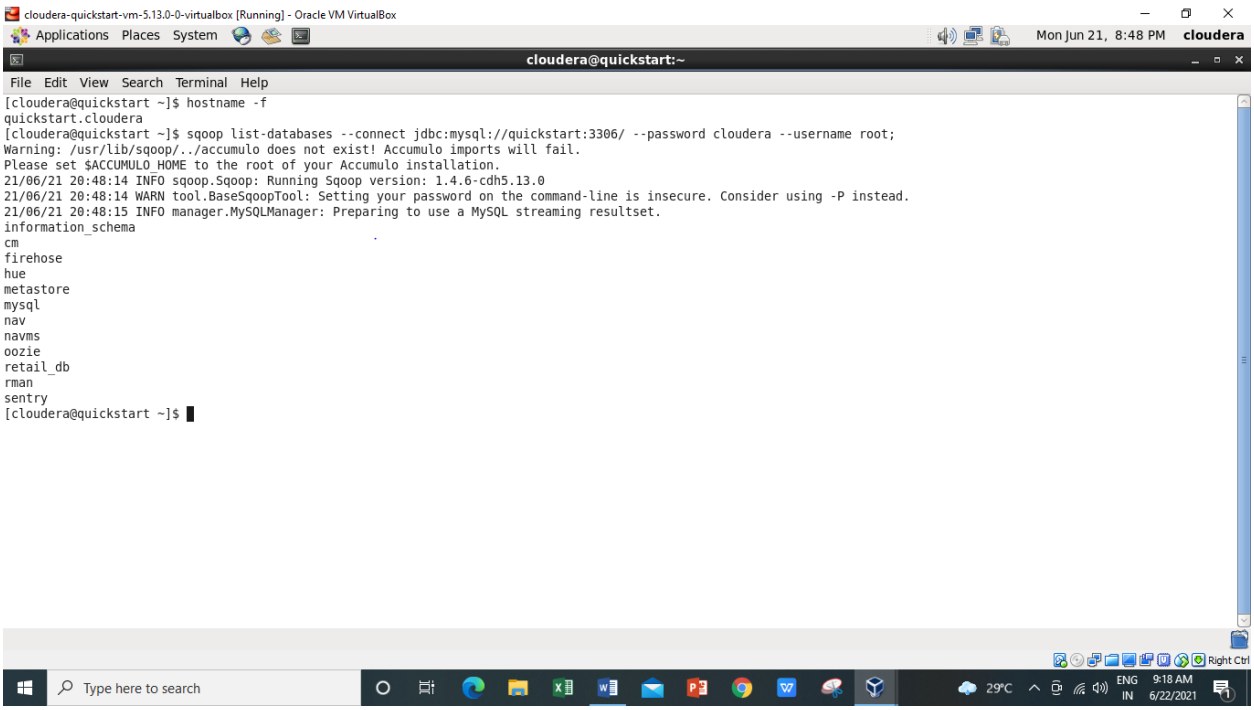



So it is giving as **quickstart.cloudera** as the name of the host that we are already connected to.

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**.




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.

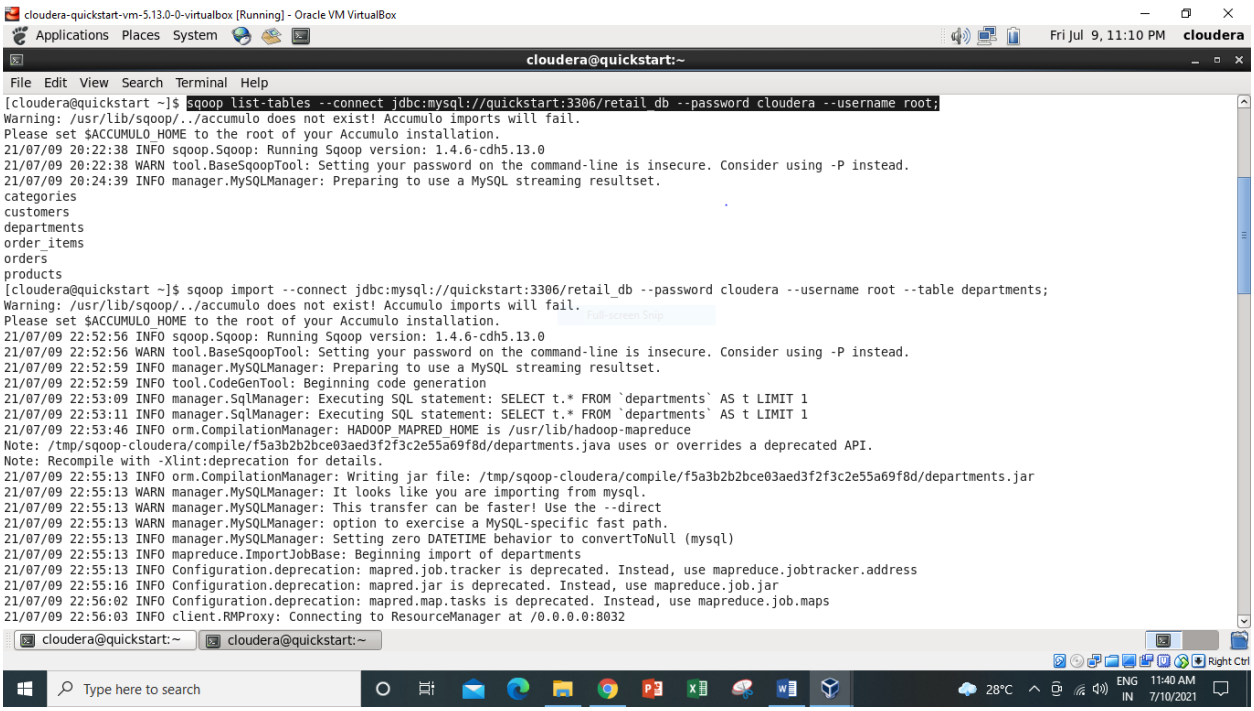
10) Now we will list out all the tables using below command.

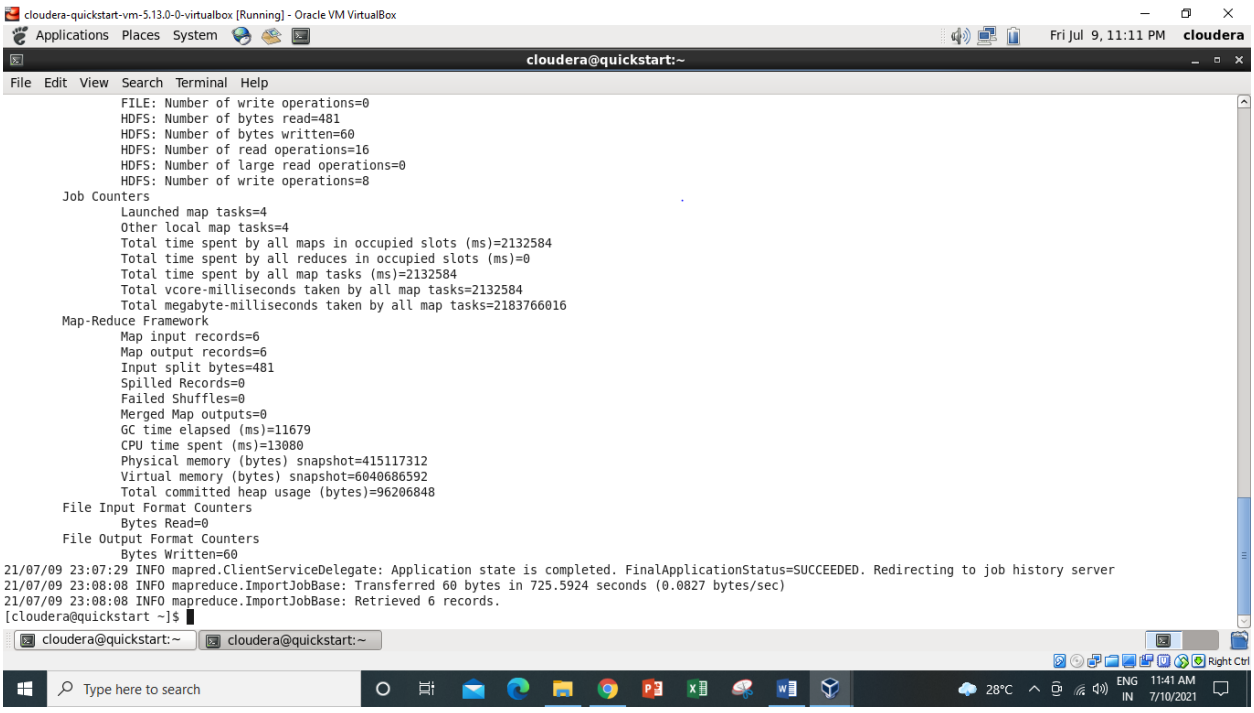
**sqoop list-tables --connect jdbc:mysql://quickstart:3306/retail_db --password cloudera --username root;**

```
[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 reatail_db database which are present inside in mysql.**

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






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 deparment using below command.

**hadoop fs –ls deparments;**

```
cloudera@quickstart ~]$ hadoop fs -ls departments;
ound 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 ~]$
```

cloudera@quickstart:~ 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***

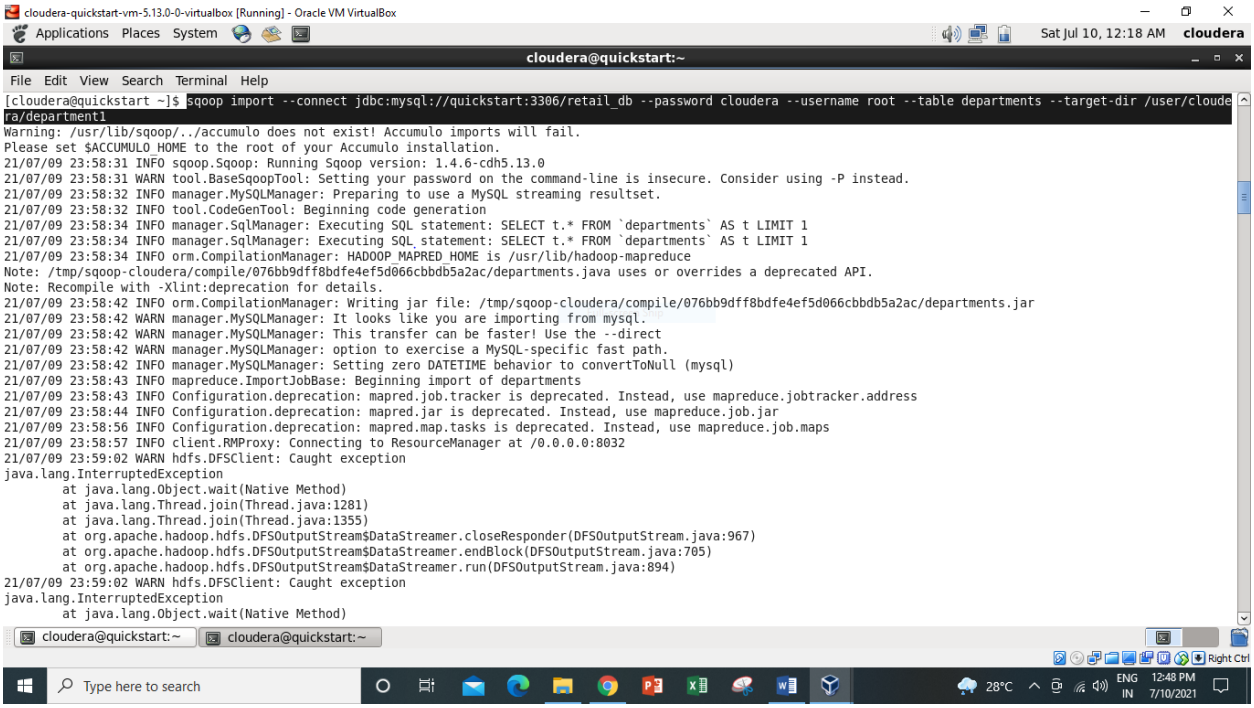
```
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/departments/part*
2,Fitness
3,Footwear
4,Apparel
5,Golf
6,Outdoors
7,Fan Shop
[cloudera@quickstart ~]$
```

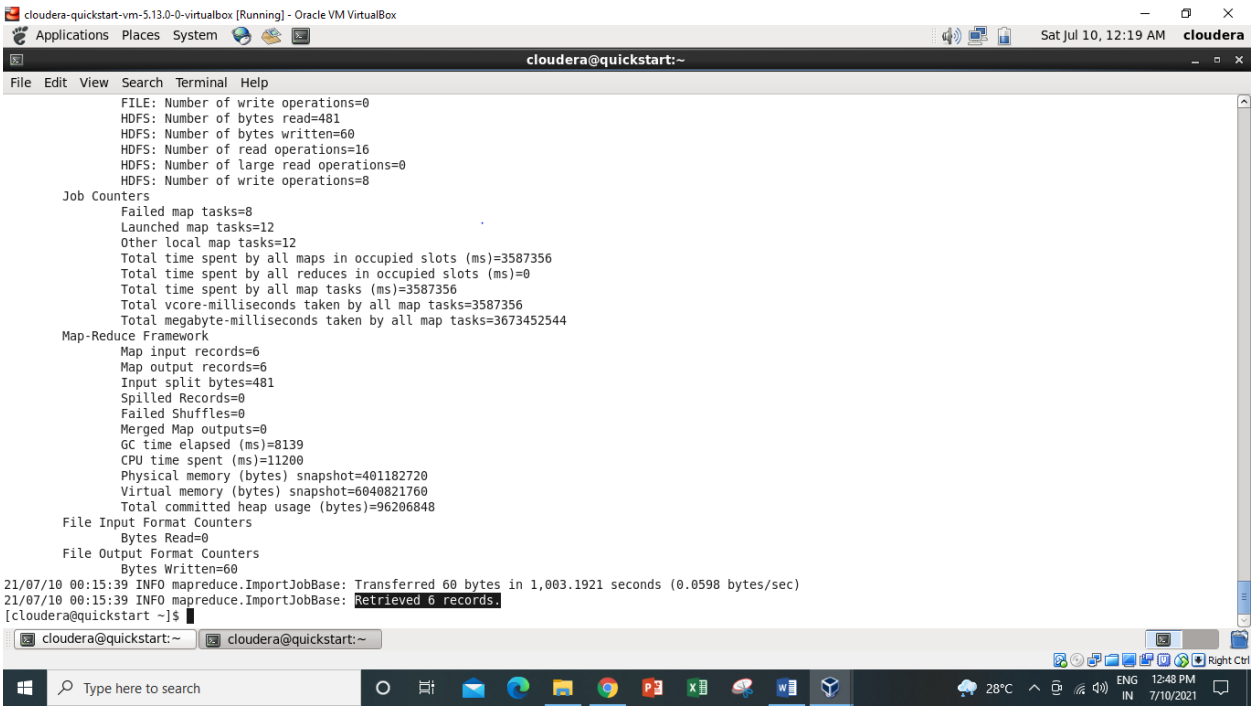
cloudera@quickstart:~ cloudera@quickstart:~

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.

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

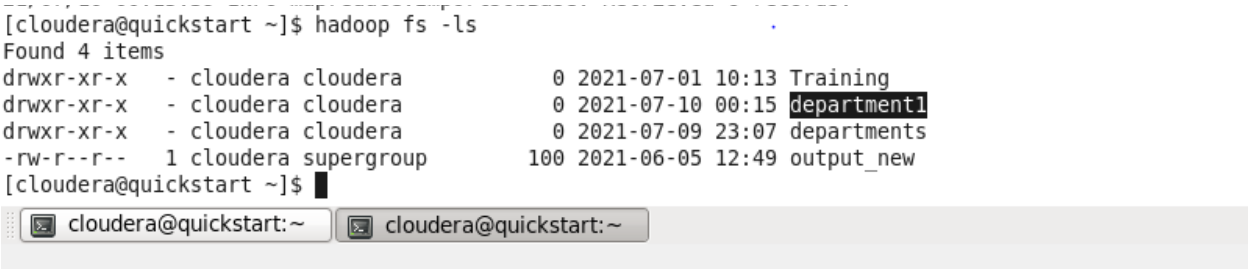



**As we can see 6 records are retrived successfully.**

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

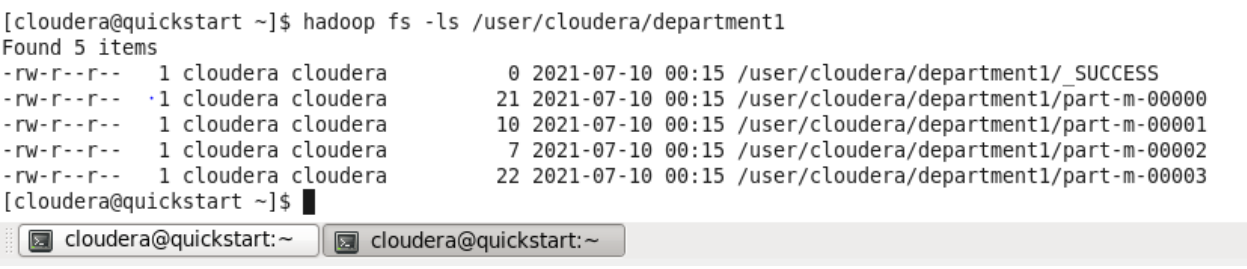
**hadoop fs -ls**




So here we have department1 directory.

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

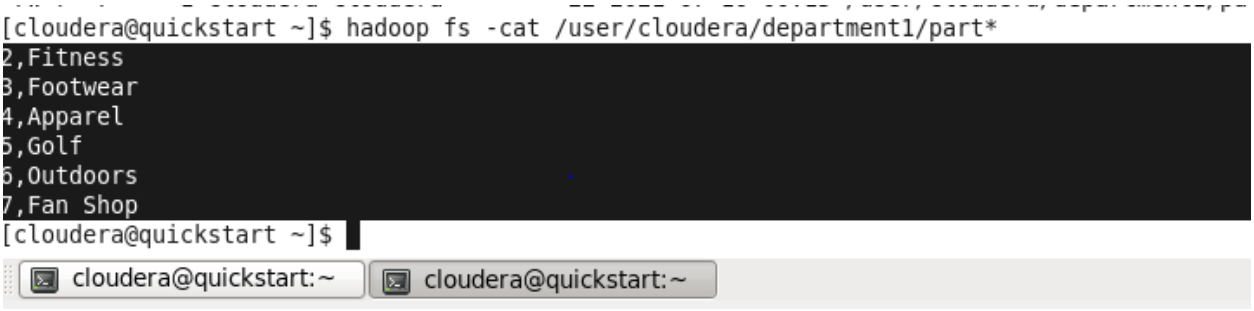
**hadoop fs -ls /user/cloudera/department1**




**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***

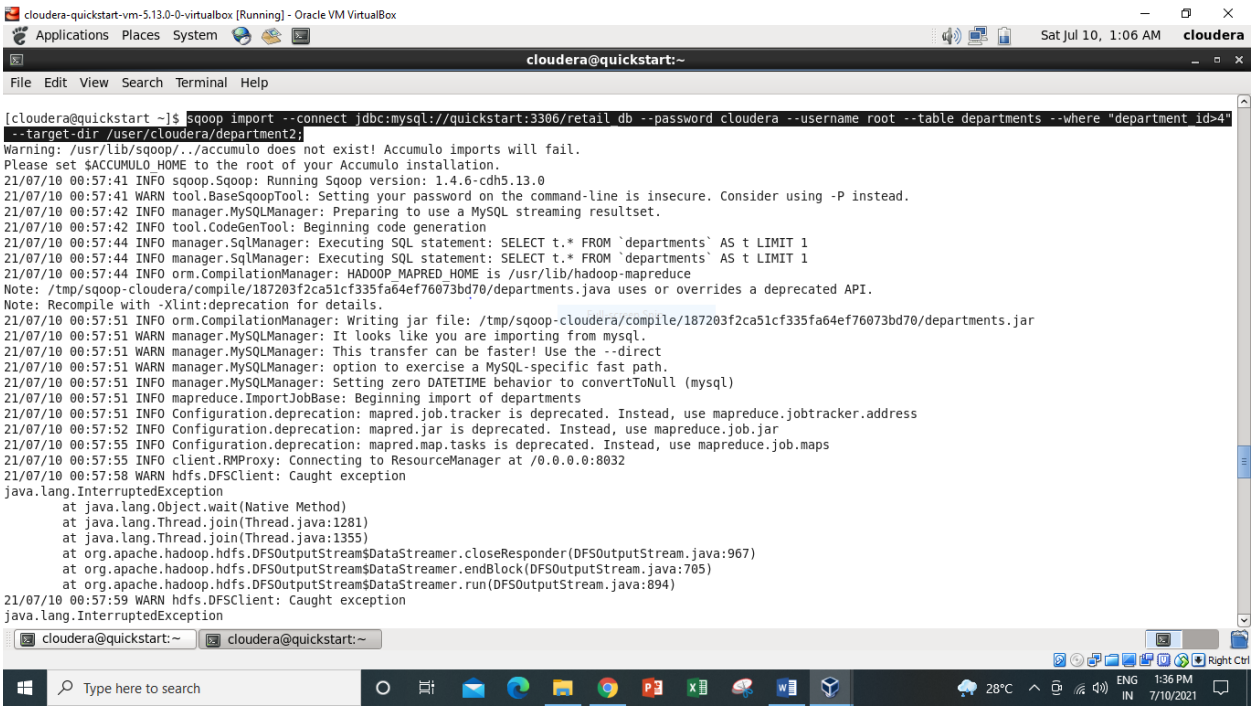


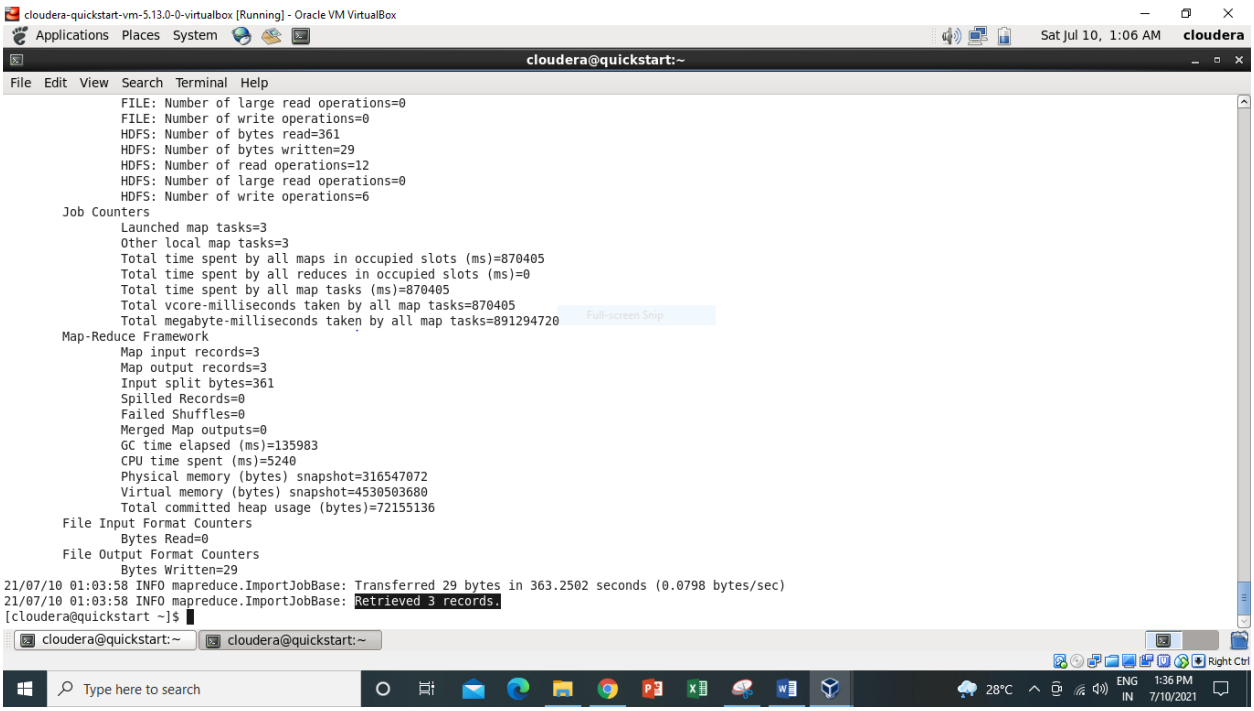

**So we have got the output.**

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;**






**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 ~]$
```

cloudera@quickstart:~ cloudera@quickstart:~

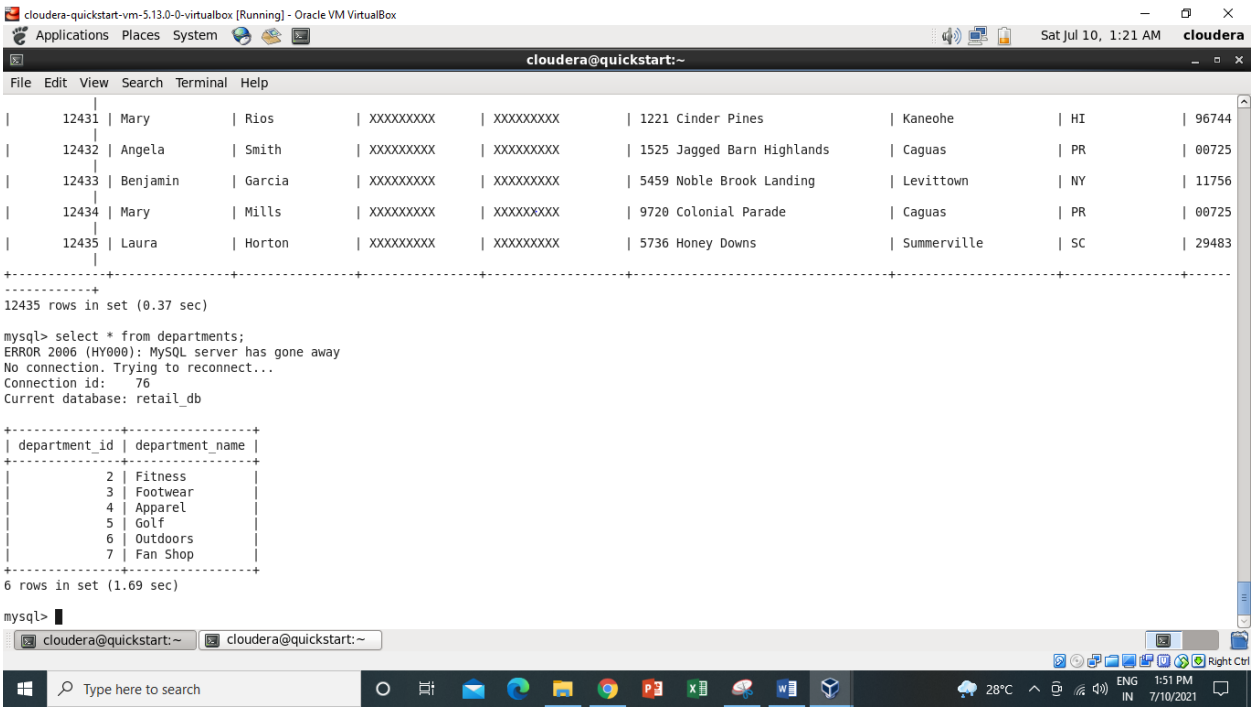
22) Now will read  the content of these part-m files using cat command.

**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 ~]$
```

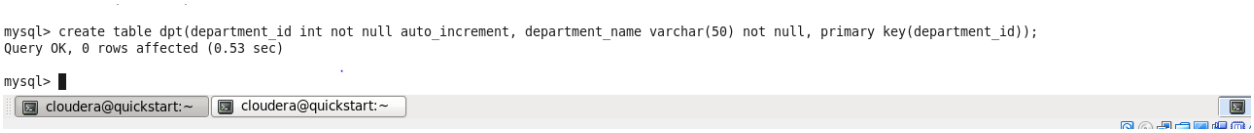
cloudera@quickstart:~ cloudera@quickstart:~

**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.**




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));**




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;**

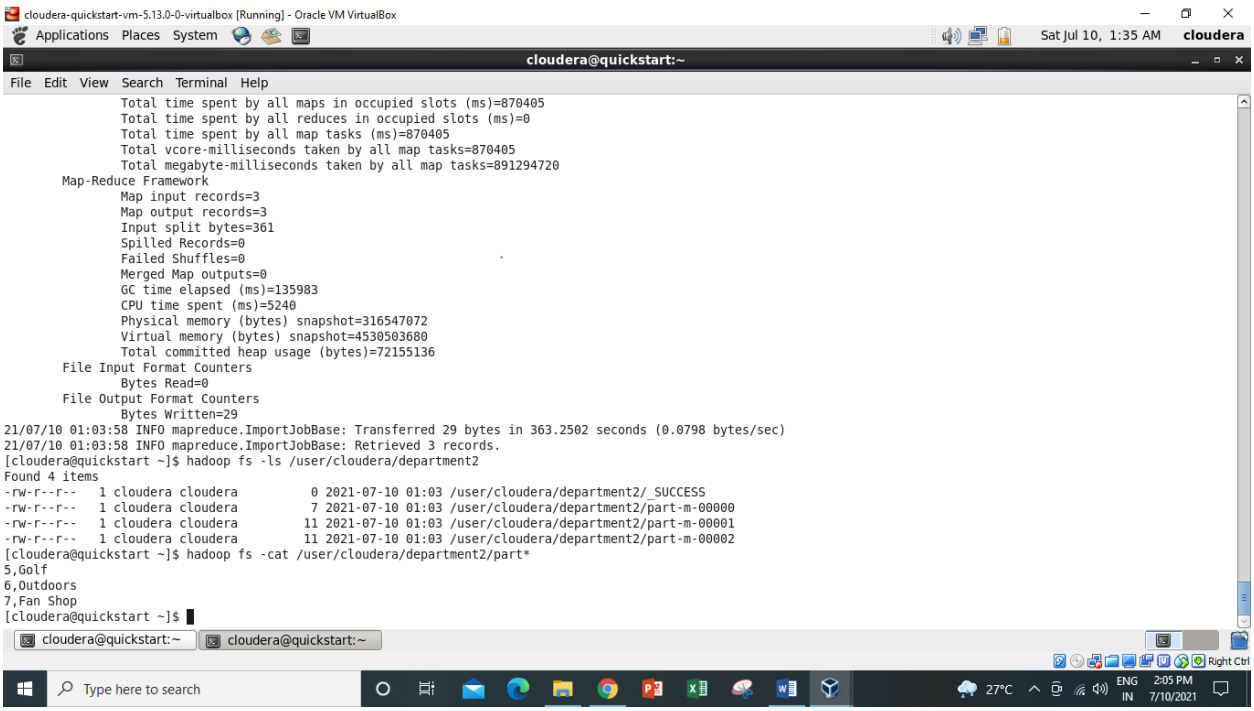
```
mysql> Select * from dpt;
Empty set (0.06 sec)

mysql>
```

cloudera@quickstart:~ cloudera@quickstart:~

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

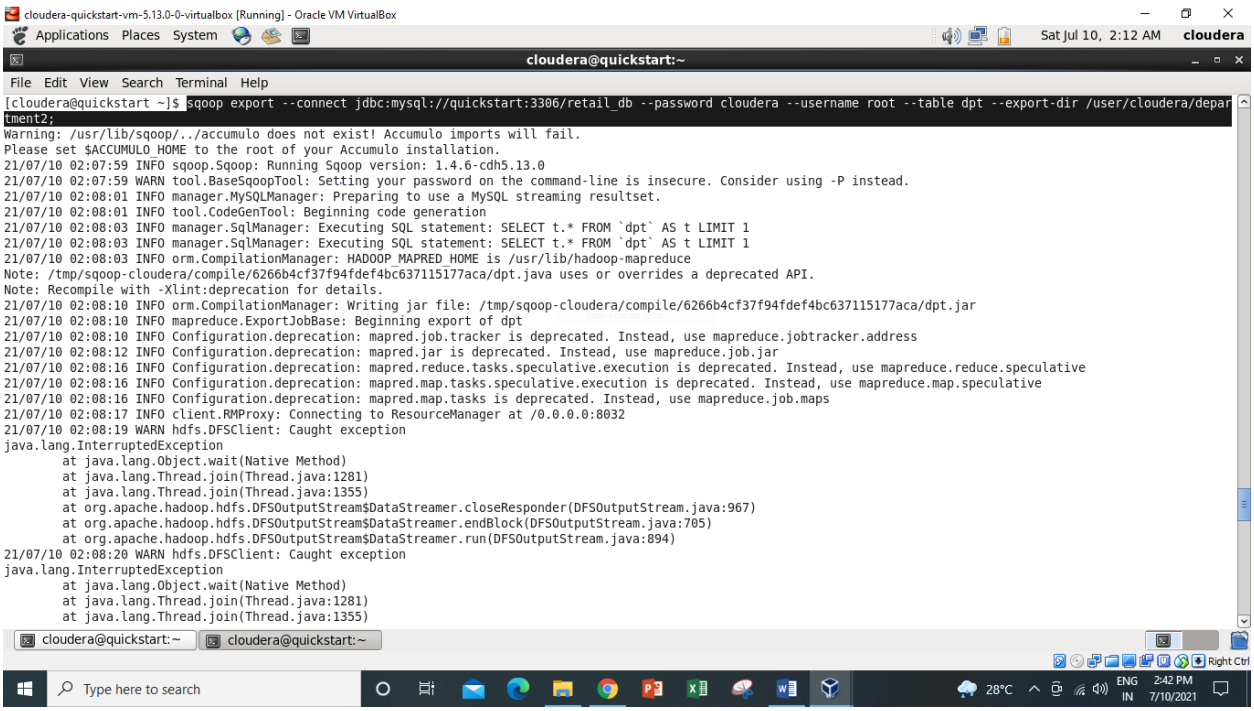
**26) Now we will exporting the data from the hdfs to dpt table of mysql. Now we will move to the sqoop terminal.**

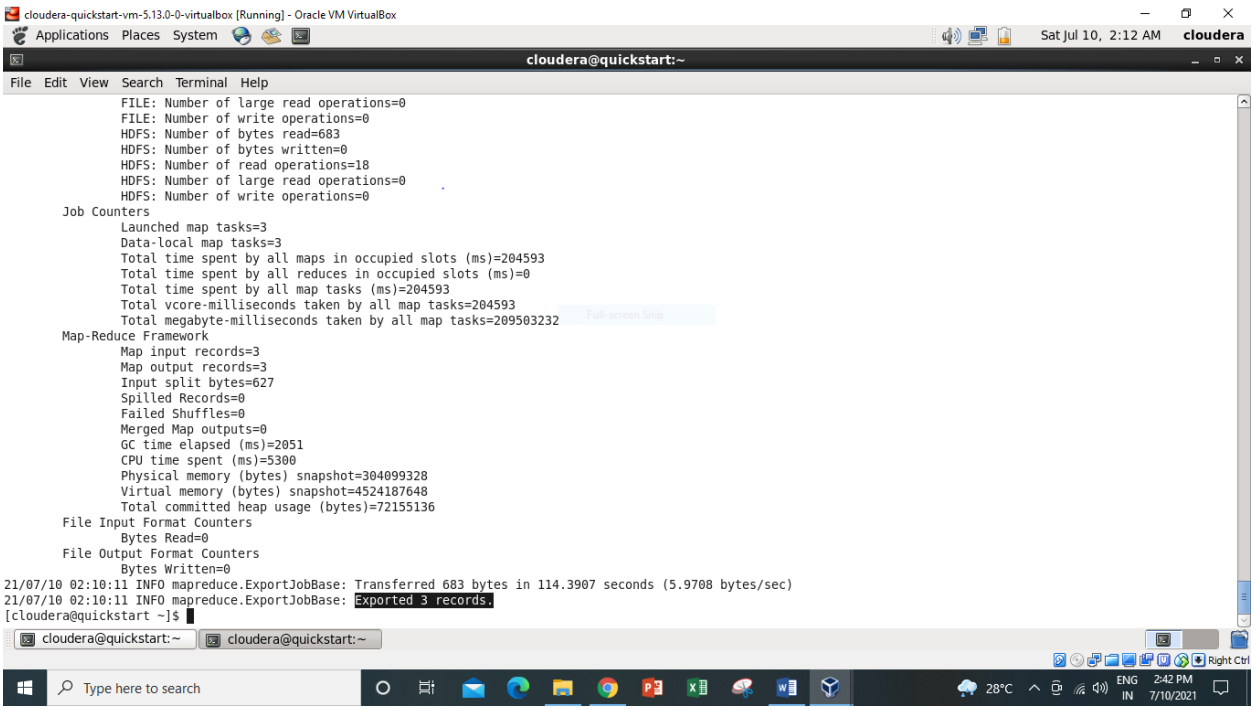



**27) So now we are performing export operation using below command.**

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;**




**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>
```

cloudera@quickstart:~ cloudera@quickstart:~

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