

Apache Sqoop

Exercise 3



IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

Trendytech Insights legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.

Sqoop Export

Login to MySQL

```
mysql -u root -p
```



Enter password: **cloudera**

Create a database (*banking*) and use it in MySQL:

```
create database banking;
```

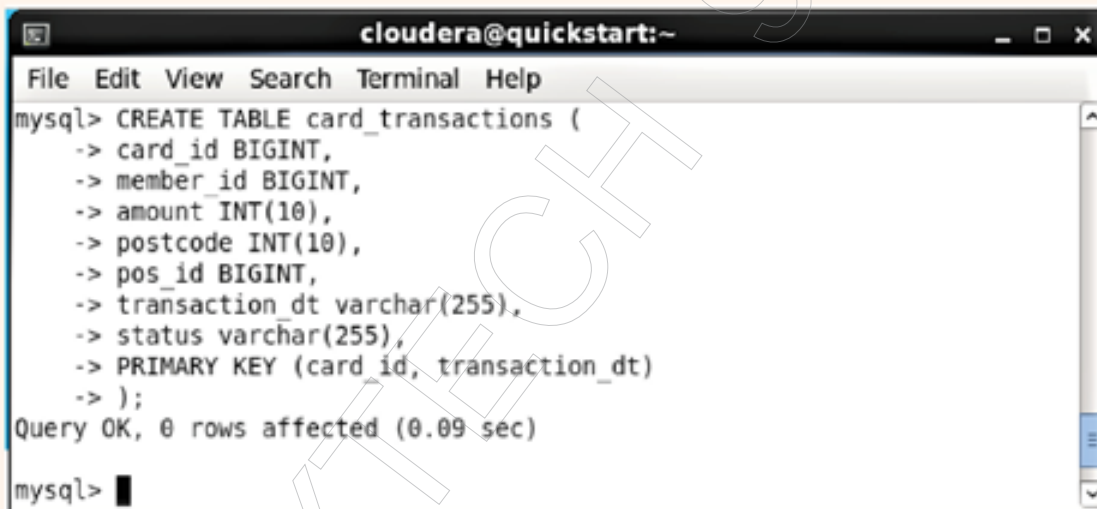
```
use banking;
```



Create a table (card_transactions) inside banking database of MySQL:

```
CREATE TABLE card_transactions (  
  card_id BIGINT,  
  member_id BIGINT,  
  amount INT(10),  
  postcode INT(10),  
  pos_id BIGINT,  
  transaction_dt varchar(255),  
  status varchar(255),  
  PRIMARY KEY (card_id, transaction_dt)  
);
```

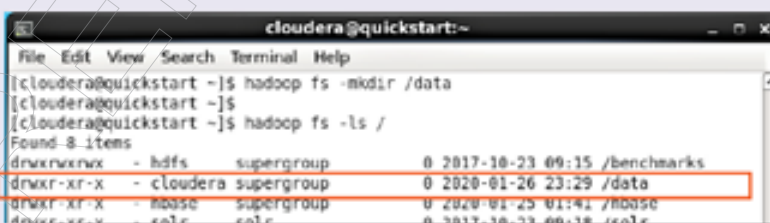
composite primary key



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> CREATE TABLE card_transactions (  
  -> card_id BIGINT,  
  -> member_id BIGINT,  
  -> amount INT(10),  
  -> postcode INT(10),  
  -> pos_id BIGINT,  
  -> transaction_dt varchar(255),  
  -> status varchar(255),  
  -> PRIMARY KEY (card_id, transaction_dt)  
  -> );  
Query OK, 0 rows affected (0.09 sec)  
mysql>
```

Create a directory inside HDFS:

```
hadoop fs -mkdir /data
```

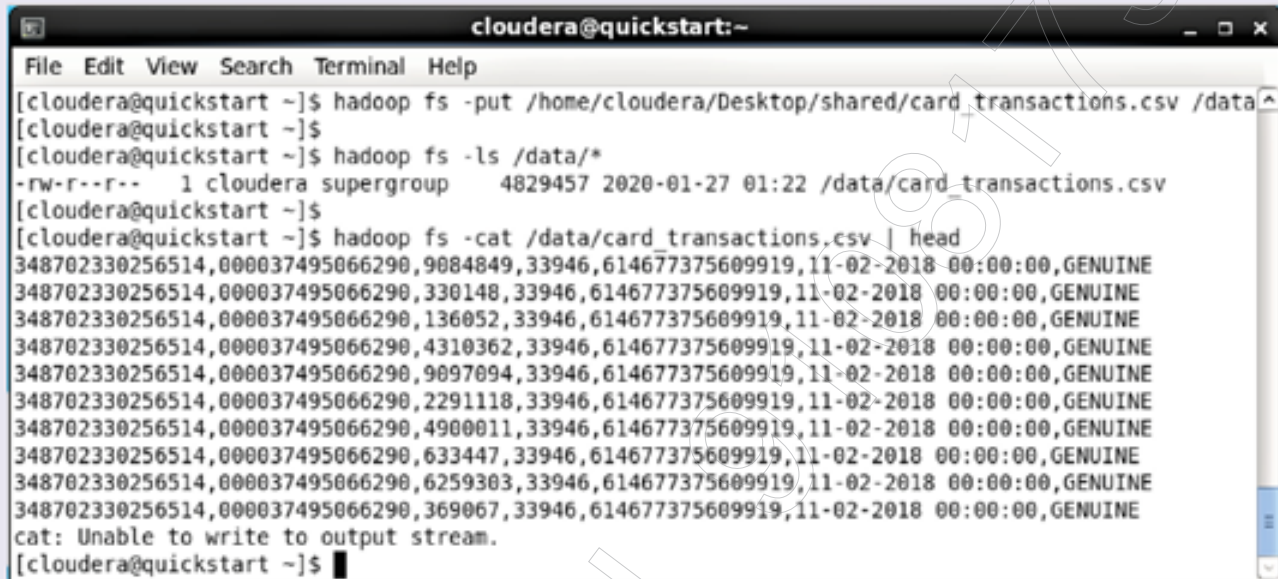


```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -mkdir /data  
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ hadoop fs -ls /  
Found 8 items  
drwxrwxrwx - hdfs supergroup 0 2017-10-23 09:15 /benchmarks  
drwxr-xr-x - cloudera supergroup 0 2020-01-26 23:29 /data  
drwxr-xr-x - hbase supergroup 0 2020-01-25 01:41 /hbase  
drwxr-xr-x - colr colr 0 2017-10-23 09:18 /colr
```

No need to create, if already available

Move the csv file (*card_transactions*) to the HDFS data directory:

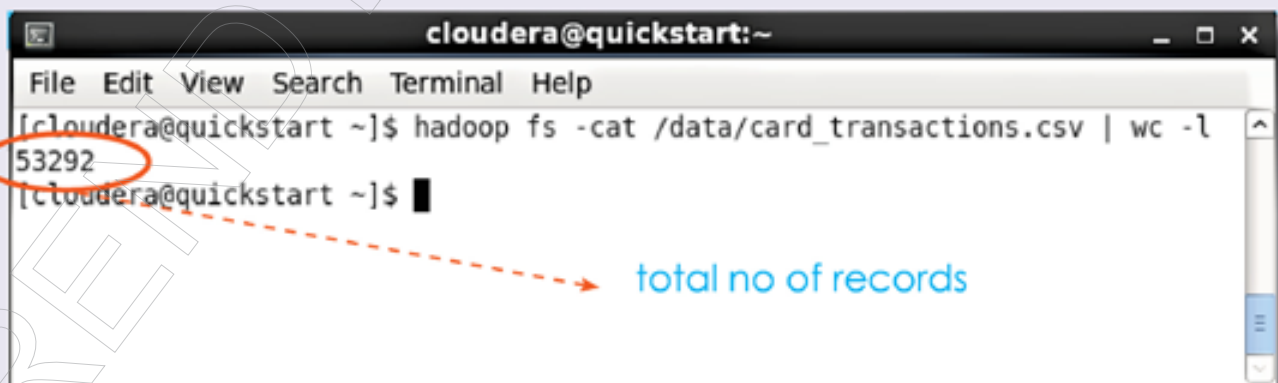
```
hadoop fs -put /home/cloudera/Desktop/shared/card_transactions.csv /data
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Desktop/shared/card_transactions.csv /data  
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ hadoop fs -ls /data/*  
-rw-r--r-- 1 cloudera supergroup 4829457 2020-01-27 01:22 /data/card_transactions.csv  
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ hadoop fs -cat /data/card_transactions.csv | head  
348702330256514,000037495066290,9084849,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,330148,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,136052,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,4310362,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,9097094,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,2291118,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,4900011,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,633447,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,6259303,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
348702330256514,000037495066290,369067,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
cat: Unable to write to output stream.  
[cloudera@quickstart ~]$
```

To check number of records present in a HDFS file (*card_transactions*) :

```
hadoop fs -cat /data/card_transactions.csv | wc -l
```



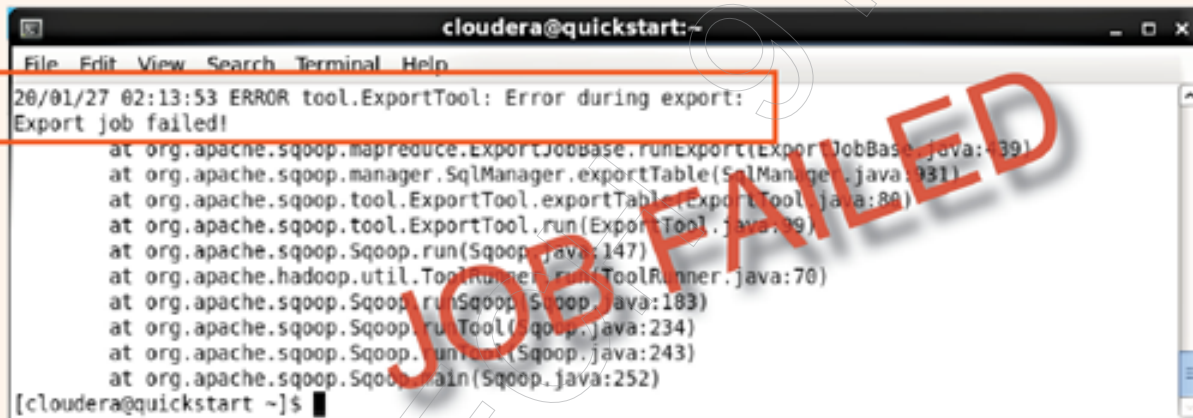
```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -cat /data/card_transactions.csv | wc -l  
53292  
[cloudera@quickstart ~]$
```

total no of records

Move data from HDFS to Sqoop using sqoop export:

Now, we will try to export the HDFS file (card_transactions.csv) to newly created MySQL table **card_transactions** using **Sqoop export**.

```
sqoop export \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table card_transactions \  
--export-dir /data/card_transactions.csv \  
--fields-terminated-by ','
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
20/01/27 02:13:53 ERROR tool.ExportTool: Error during export:  
Export job failed!  
at org.apache.sqoop.mapreduce.ExportJobBase.runExport(ExportJobBase.java:39)  
at org.apache.sqoop.manager.SqlManager.exportTable(SqlManager.java:931)  
at org.apache.sqoop.tool.ExportTool.exportTable(ExportTool.java:86)  
at org.apache.sqoop.tool.ExportTool.run(ExportTool.java:99)  
at org.apache.sqoop.Sqoop.run(Sqoop.java:147)  
at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)  
at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)  
at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)  
at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)  
at org.apache.sqoop.Sqoop.main(Sqoop.java:252)  
[cloudera@quickstart ~]$
```

Note: The Job will fail

The `card_transactions` table data should look like below:

card_id	merchant_id	amount	card_type	card_holder_name	transaction_date	transaction_time	transaction_status
6478888441720966	273246841077378	3094329	59043	885476654650050	25-12-2016	07:08:31	GENUINE
6478888441720966	273246841077378	7937580	18254	691255588719240	26-01-2017	11:36:19	GENUINE
6478888441720966	273246841077378	6317165	25685	276217887671106	26-03-2017	14:07:43	GENUINE
6478888441720966	273246841077378	3868955	59758	32934628245497	26-09-2017	13:27:30	GENUINE
6478888441720966	273246841077378	6127519	39040	104442873565255	27-03-2017	16:27:45	GENUINE
6478888441720966	273246841077378	8853112	31782	394565472842075	27-06-2017	13:29:39	GENUINE
6478888441720966	273246841077378	8118813	39361	537355573005274	27-10-2017	17:20:36	GENUINE
6478888441720966	273246841077378	4342111	57766	499153824060670	28-01-2018	04:25:10	GENUINE
6478888441720966	273246841077378	8464524	36420	497898769747957	28-08-2017	15:59:49	GENUINE
6478888441720966	273246841077378	6618273	17772	708575154329682	28-10-2016	13:58:29	GENUINE
6478888441720966	273246841077378	7857927	80863	554649332037371	28-10-2017	12:39:17	GENUINE
6478888441720966	273246841077378	1438636	82431	360359919733360	29-06-2017	16:24:23	GENUINE
6478888441720966	273246841077378	6114961	64078	359729828861623	30-03-2017	19:57:12	GENUINE
6478888441720966	273246841077378	7365055	40075	657752585733304	31-03-2017	20:31:54	GENUINE
6478888441720966	273246841077378	7158863	24563	44686270157711	31-05-2017	20:42:02	GENUINE
6478888441720966	273246841077378	9718094	31567	652954787139096	31-07-2017	01:33:29	GENUINE
6478888441720966	273246841077378	9154188	98350	557363801899838	31-10-2016	17:27:56	GENUINE

How to track the failed job:

- Go to the log and find [The url to track the job](#)
- Copy the given url

```

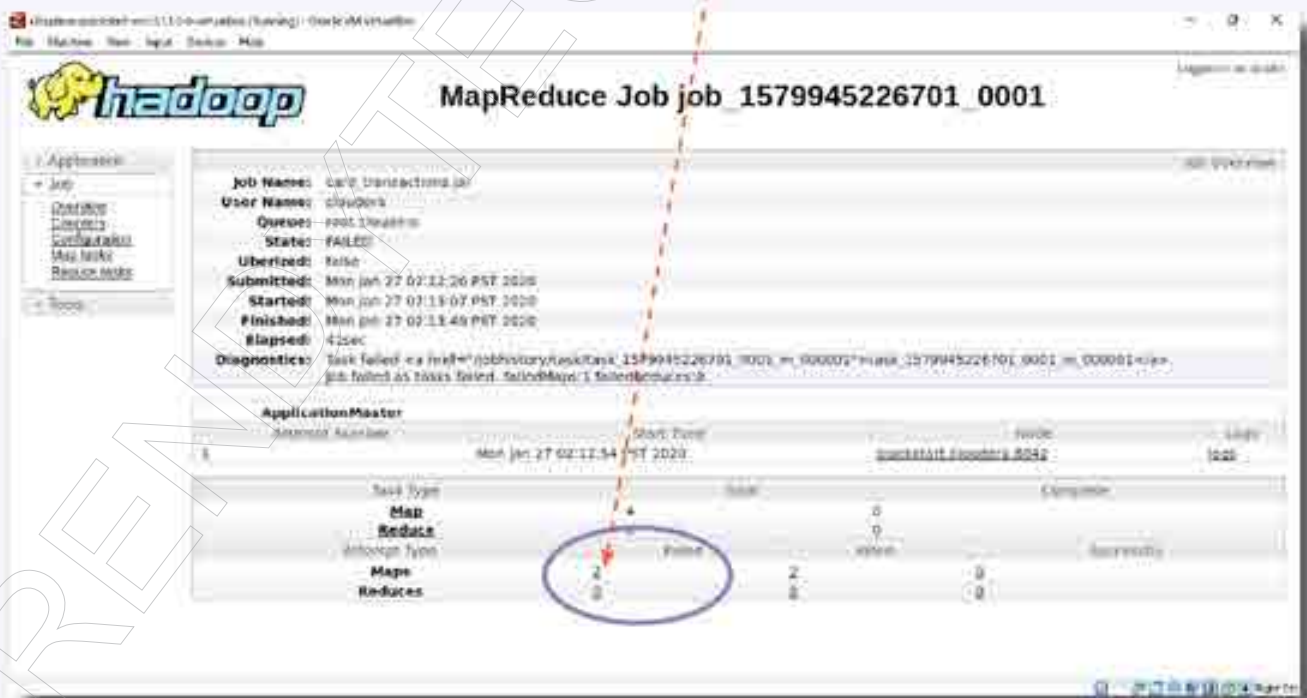
20/01/27 02:12:18 INFO input.FileInputFormat: Total input paths to process : 1
20/01/27 02:12:18 INFO mapreduce.JobSubmitter: number of splits:4
20/01/27 02:12:19 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.map.speculative
20/01/27 02:12:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579945226701_0001
20/01/27 02:12:28 INFO impl.YarnClientImpl: Submitted application application_1579945226701_0001
20/01/27 02:12:28 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/p
roxy/application_1579945226701_0001/
20/01/27 02:12:28 INFO mapreduce.Job: application_1579945226701_0001
20/01/27 02:13:09 INFO mapreduce.Job: application_1579945226701_0001 running in uber mode : false
20/01/27 02:13:09 INFO mapreduce.Job: application_1579945226701_0001
20/01/27 02:13:51 INFO mapreduce.Job: application_1579945226701_0001
20/01/27 02:13:52 INFO mapreduce.Job: application_1579945226701_0001
20/01/27 02:13:53 INFO mapreduce.Job: application_1579945226701_0001 failed with state FAILED due to:
Task failed task_1579945226701_0001
Job failed as tasks failed. failedMap

```

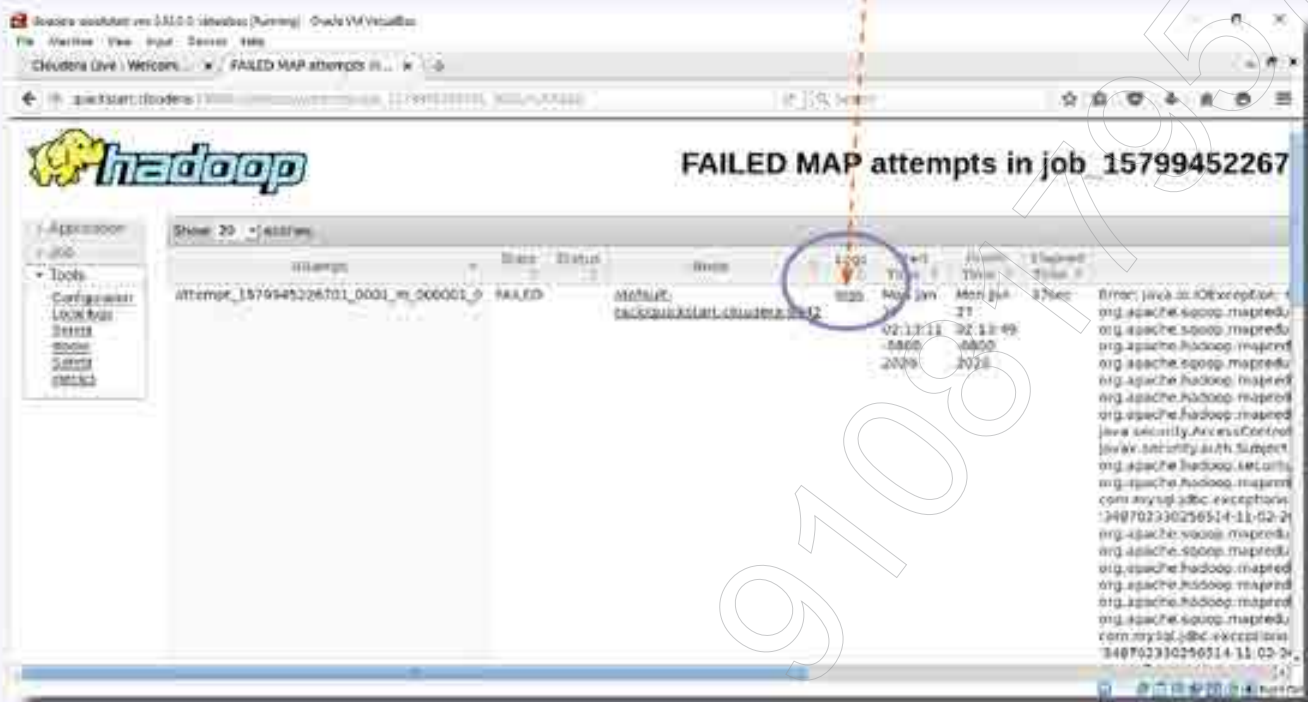

- Go to the web browser and **paste** the url and enter



- Job history will show all details - **click** the hyperlink under **failed**



- In the FAILED MAP attempts window - **click** on logs under **logs**

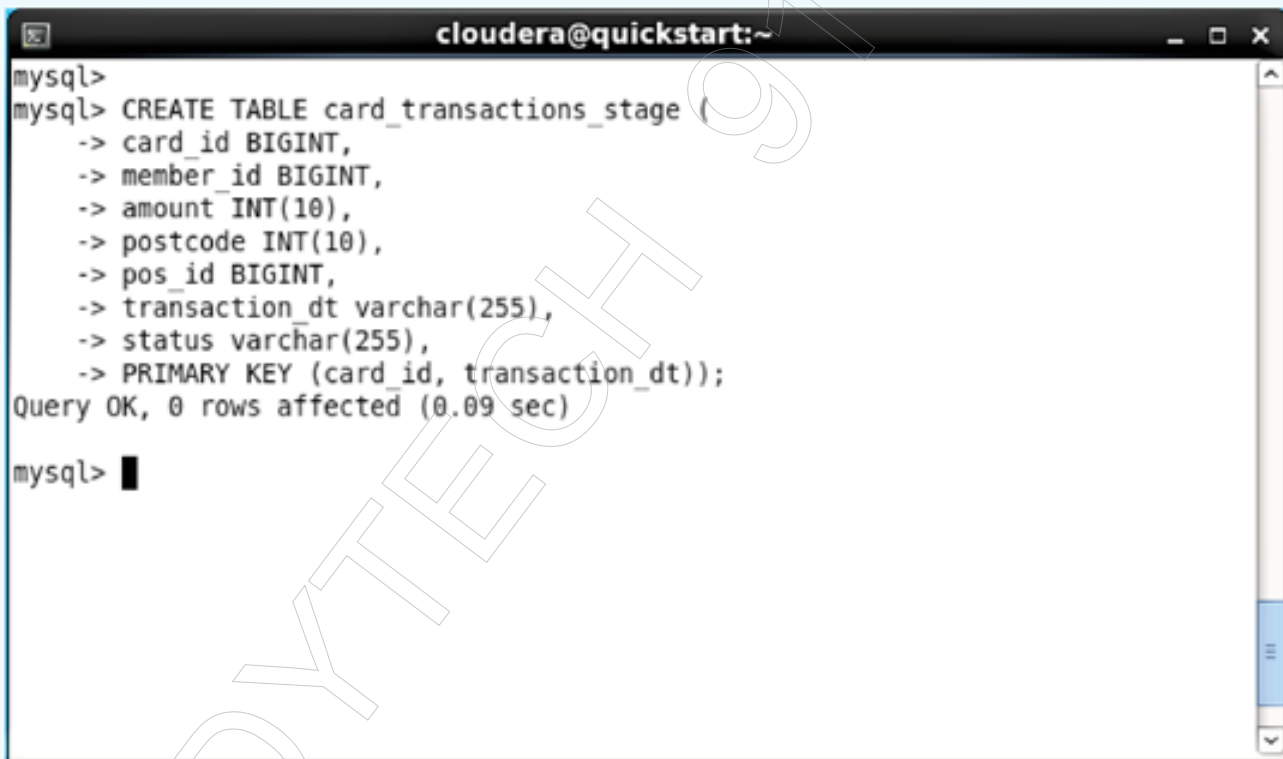


- In the Log window, you will find the actual **errors**



Create a staging table `card_transactions_stage` in mysql:

```
CREATE TABLE card_transactions_stage (  
  card_id BIGINT,  
  member_id BIGINT,  
  amount INT(10),  
  postcode INT(10),  
  pos_id BIGINT,  
  transaction_dt varchar(255),  
  status varchar(255),  
  PRIMARY KEY (card_id, transaction_dt));
```

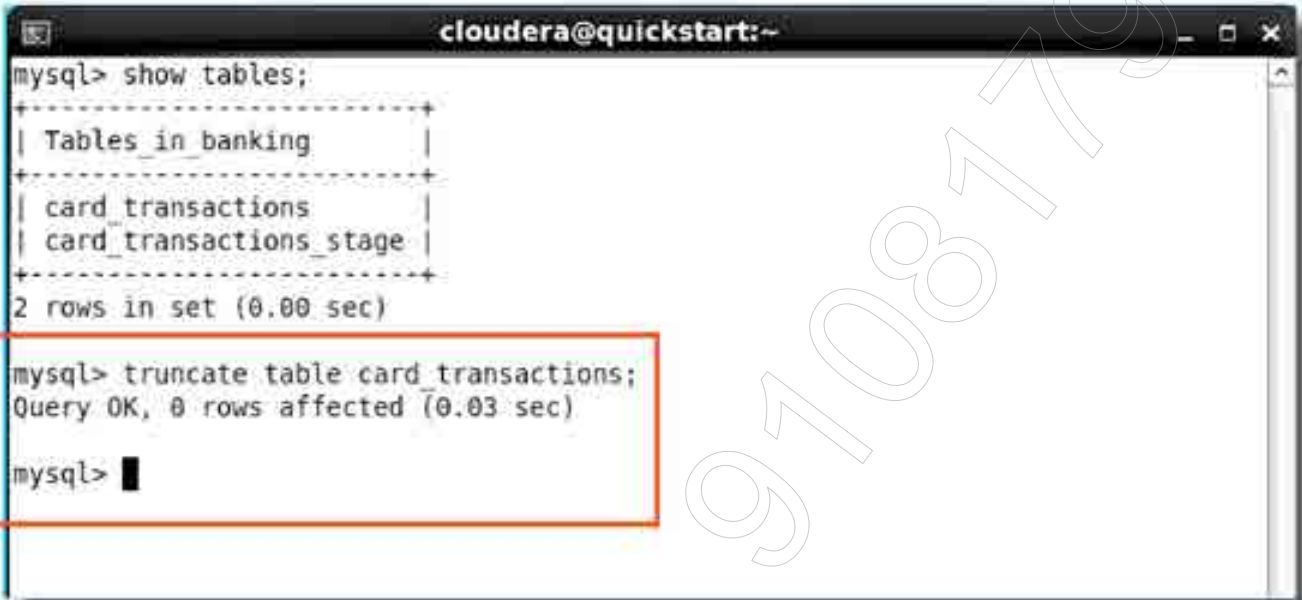


The screenshot shows a terminal window titled "cloudera@quickstart:~". Inside, a MySQL prompt "mysql>" is followed by the command to create the table "card_transactions_stage". The command is displayed in a multi-line format with arrows indicating continuation. The output shows "Query OK, 0 rows affected (0.09 sec)". The prompt "mysql>" is followed by a cursor.

```
mysql>  
mysql> CREATE TABLE card_transactions_stage (  
-> card_id BIGINT,  
-> member_id BIGINT,  
-> amount INT(10),  
-> postcode INT(10),  
-> pos_id BIGINT,  
-> transaction_dt varchar(255),  
-> status varchar(255),  
-> PRIMARY KEY (card_id, transaction_dt));  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> █
```

Truncate *card_transactions* table from mysql:

```
truncate table card_transactions;
```



A terminal window titled 'cloudera@quickstart:~' showing MySQL commands and output. The first command is 'mysql> show tables;', which returns a table with two rows: 'Tables_in_banking' and 'card_transactions', and 'card_transactions_stage'. The second command is 'mysql> truncate table card_transactions;', which returns 'Query OK, 0 rows affected (0.03 sec)'. The third command is 'mysql>' followed by a cursor. A red box highlights the second command and its output.

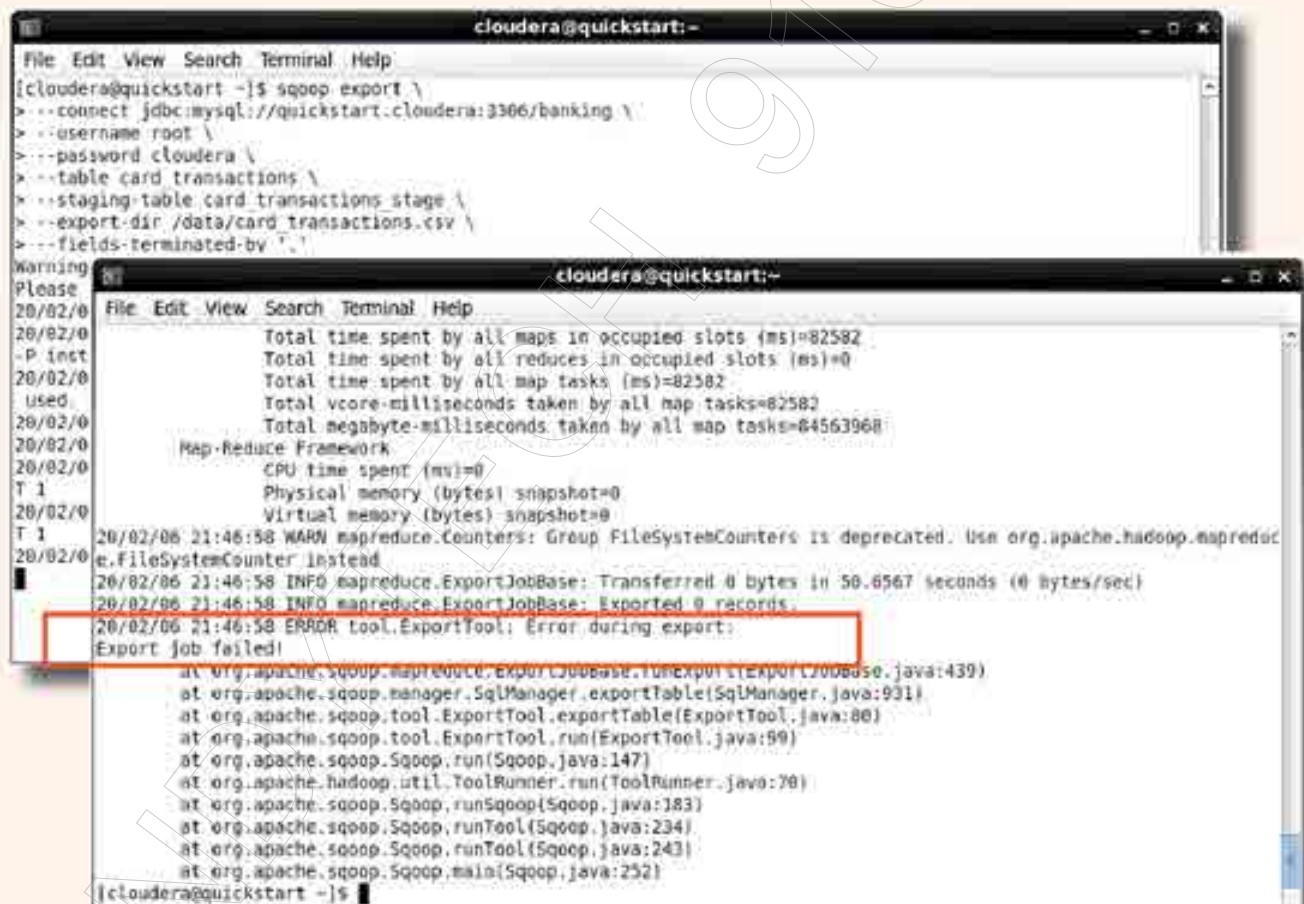
```
mysql> show tables;
+-----+
| Tables_in_banking |
+-----+
| card_transactions |
| card_transactions_stage |
+-----+
2 rows in set (0.00 sec)

mysql> truncate table card_transactions;
Query OK, 0 rows affected (0.03 sec)

mysql> █
```


Export data from hdfs to mysql using staging table:

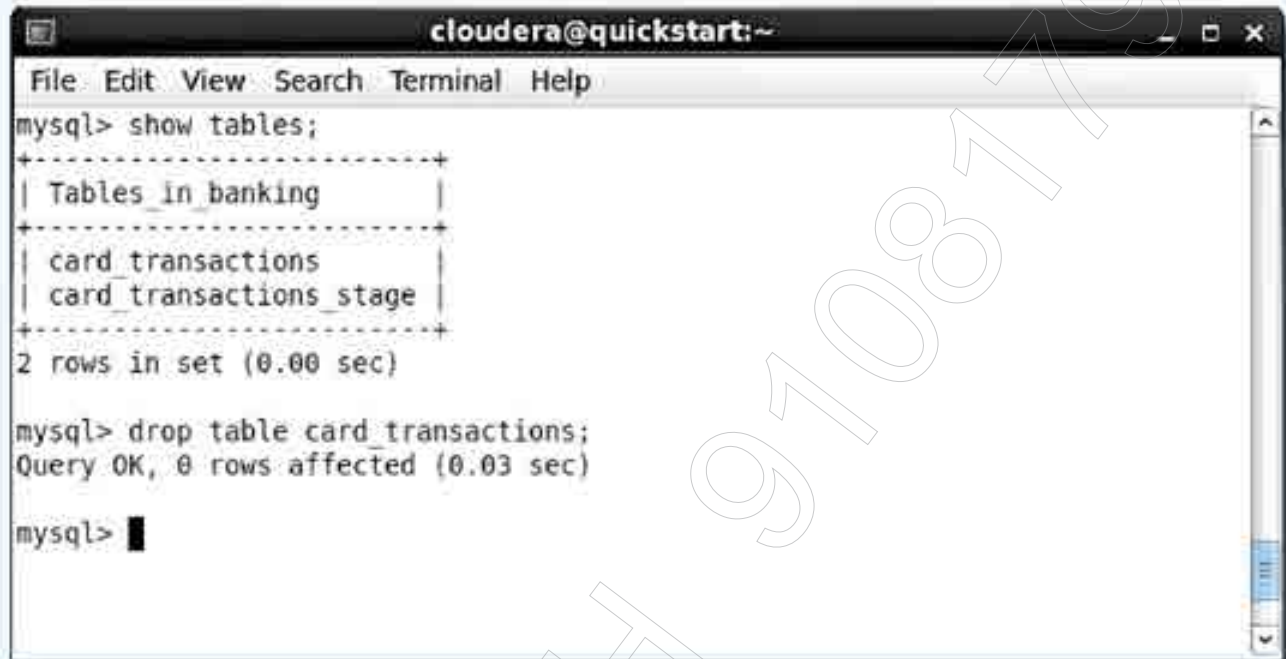
```
scoop export \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table card_transactions \  
--staging-table card_transactions_stage \  
--export-dir /data/card_transactions.csv \  
--fields-terminated-by ','
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ scoop export \  
> --connect jdbc:mysql://quickstart.cloudera:3306/banking \  
> --username root \  
> --password cloudera \  
> --table card_transactions \  
> --staging-table card_transactions_stage \  
> --export-dir /data/card_transactions.csv \  
> --fields-terminated-by ','  
Warning  
Please  
20/02/06 21:46:58 INFO mapreduce.ExportJobBase: Transferred 0 bytes in 50.6567 seconds (0 bytes/sec)  
20/02/06 21:46:58 INFO mapreduce.ExportJobBase: Exported 0 records.  
20/02/06 21:46:58 ERROR tool.ExportTool: Error during export:  
Export job failed!  
at org.apache.scoop.mapreduce.ExportJobBase.runExport(ExportJobBase.java:439)  
at org.apache.scoop.manager.SqlManager.exportTable(SqlManager.java:931)  
at org.apache.scoop.tool.ExportTool.exportTable(ExportTool.java:88)  
at org.apache.scoop.tool.ExportTool.run(ExportTool.java:99)  
at org.apache.scoop.Scoop.run(Scoop.java:147)  
at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)  
at org.apache.scoop.Scoop.runScoop(Scoop.java:183)  
at org.apache.scoop.Scoop.runTool(Scoop.java:234)  
at org.apache.scoop.Scoop.runTool(Scoop.java:243)  
at org.apache.scoop.Scoop.main(Scoop.java:252)  
[cloudera@quickstart ~]$
```


Drop both the staging and the actual table from mysql:

```
drop table card_transactions;
```



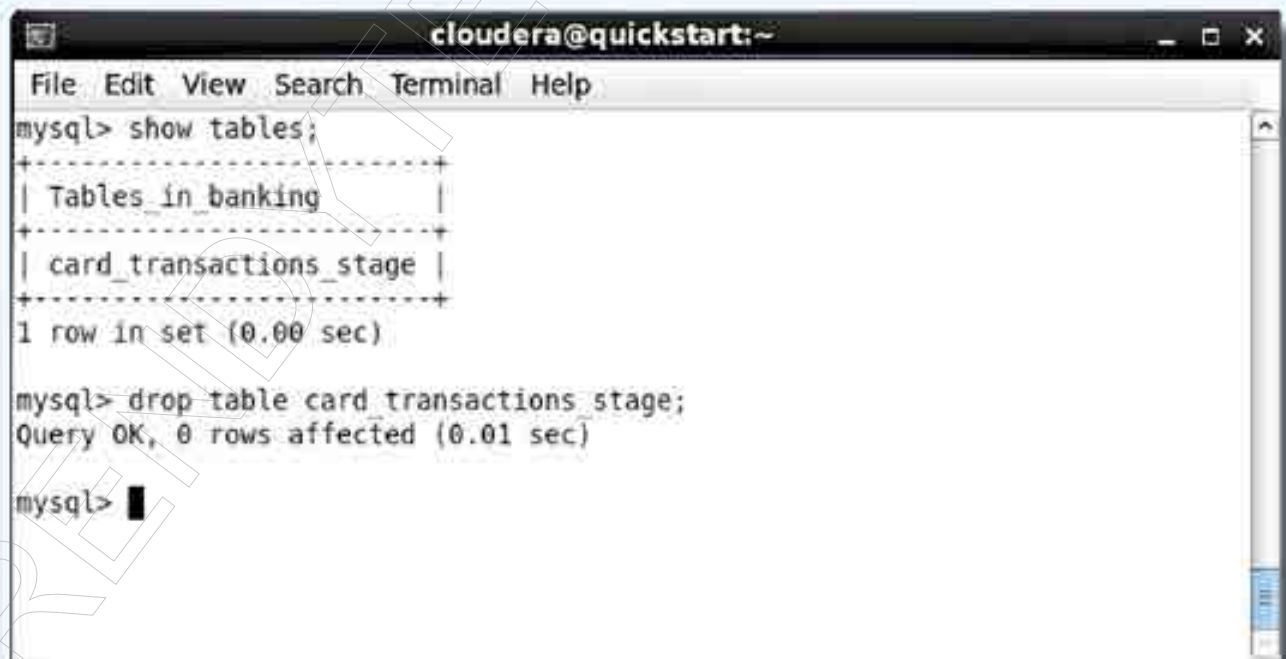
A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
mysql> show tables;
+-----+
| Tables_in_banking |
+-----+
| card_transactions  |
| card_transactions_stage |
+-----+
2 rows in set (0.00 sec)

mysql> drop table card_transactions;
Query OK, 0 rows affected (0.03 sec)

mysql> █
```

```
drop table card_transactions_stage;
```



A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

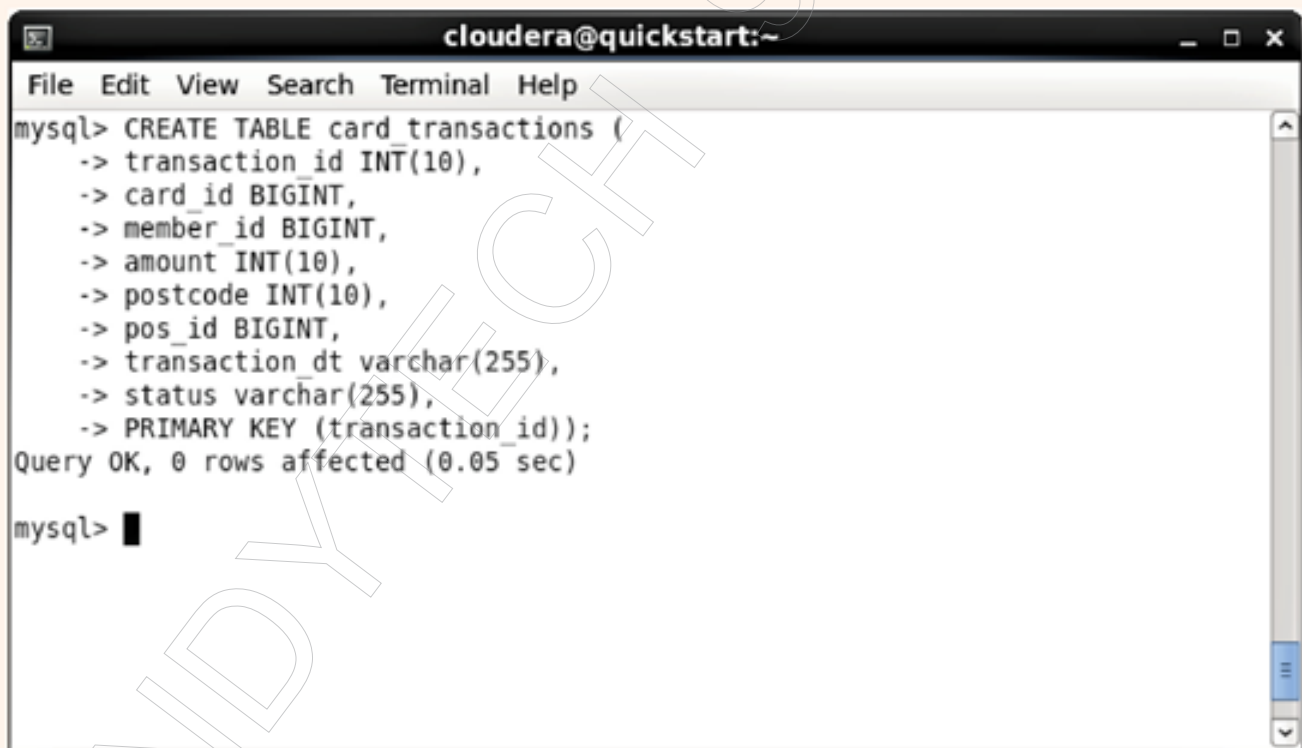
```
mysql> show tables;
+-----+
| Tables_in_banking |
+-----+
| card_transactions_stage |
+-----+
1 row in set (0.00 sec)

mysql> drop table card_transactions stage;
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

Create again same two (*card_transactions* & *card_transactions_stage*) tables in banking database of MySQL:

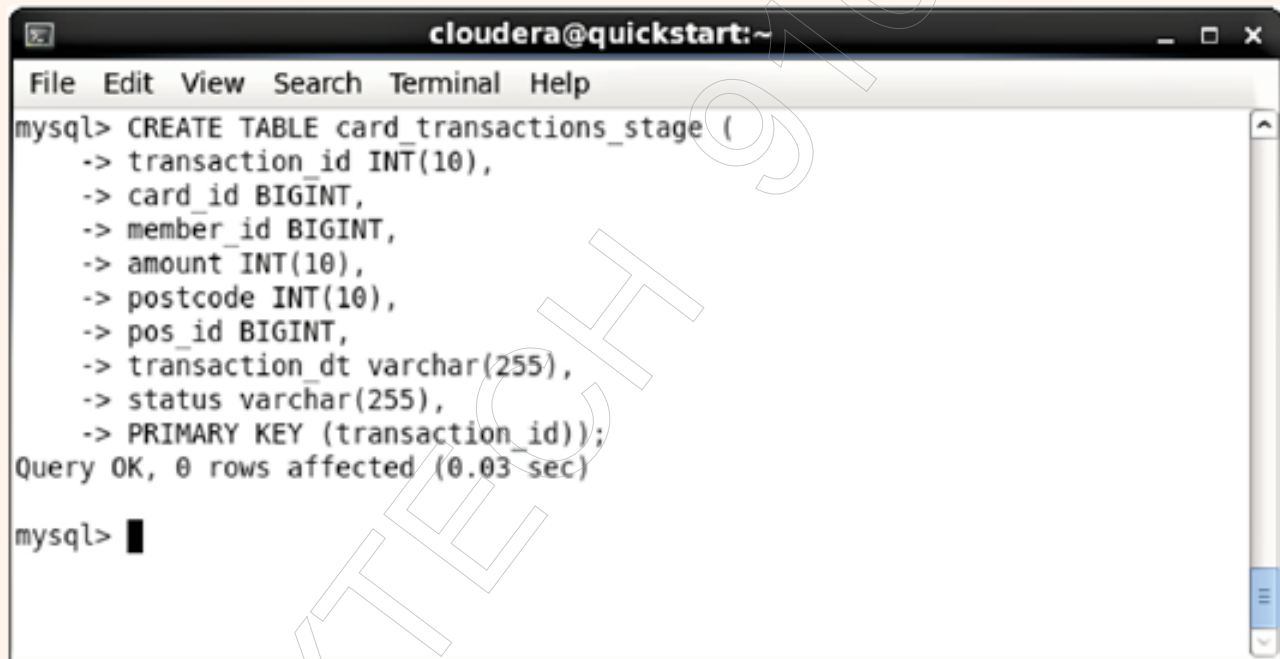
```
CREATE TABLE card_transactions (  
transaction_id INT(10),  
card_id BIGINT,  
member_id BIGINT,  
amount INT(10),  
postcode INT(10),  
pos_id BIGINT,  
transaction_dt varchar(255),  
status varchar(255),  
PRIMARY KEY (transaction_id));
```



The screenshot shows a terminal window titled "cloudera@quickstart:~". The terminal displays the following commands and output:

```
mysql> CREATE TABLE card_transactions (  
-> transaction_id INT(10),  
-> card_id BIGINT,  
-> member_id BIGINT,  
-> amount INT(10),  
-> postcode INT(10),  
-> pos_id BIGINT,  
-> transaction_dt varchar(255),  
-> status varchar(255),  
-> PRIMARY KEY (transaction_id));  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> █
```

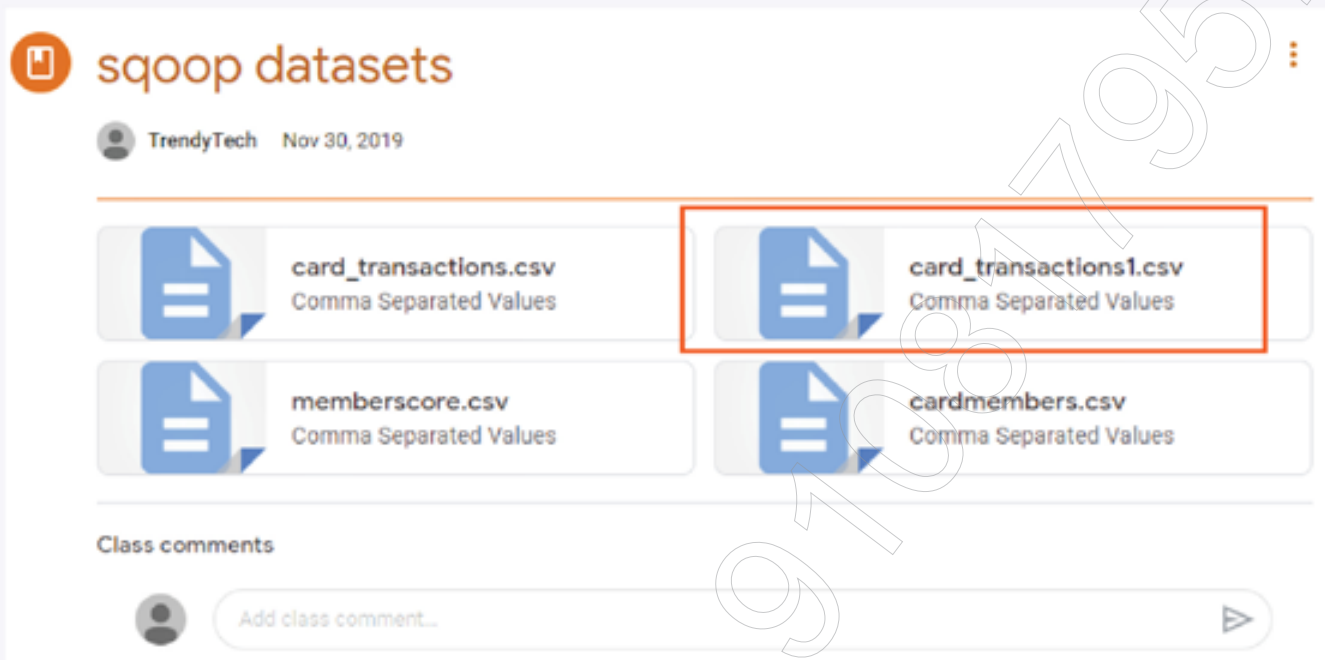
```
CREATE TABLE card_transactions_stage (  
transaction_id INT(10),  
card_id BIGINT,  
member_id BIGINT,  
amount INT(10),  
postcode INT(10),  
pos_id BIGINT,  
transaction_dt varchar(255),  
status varchar(255),  
PRIMARY KEY (transaction_id));
```



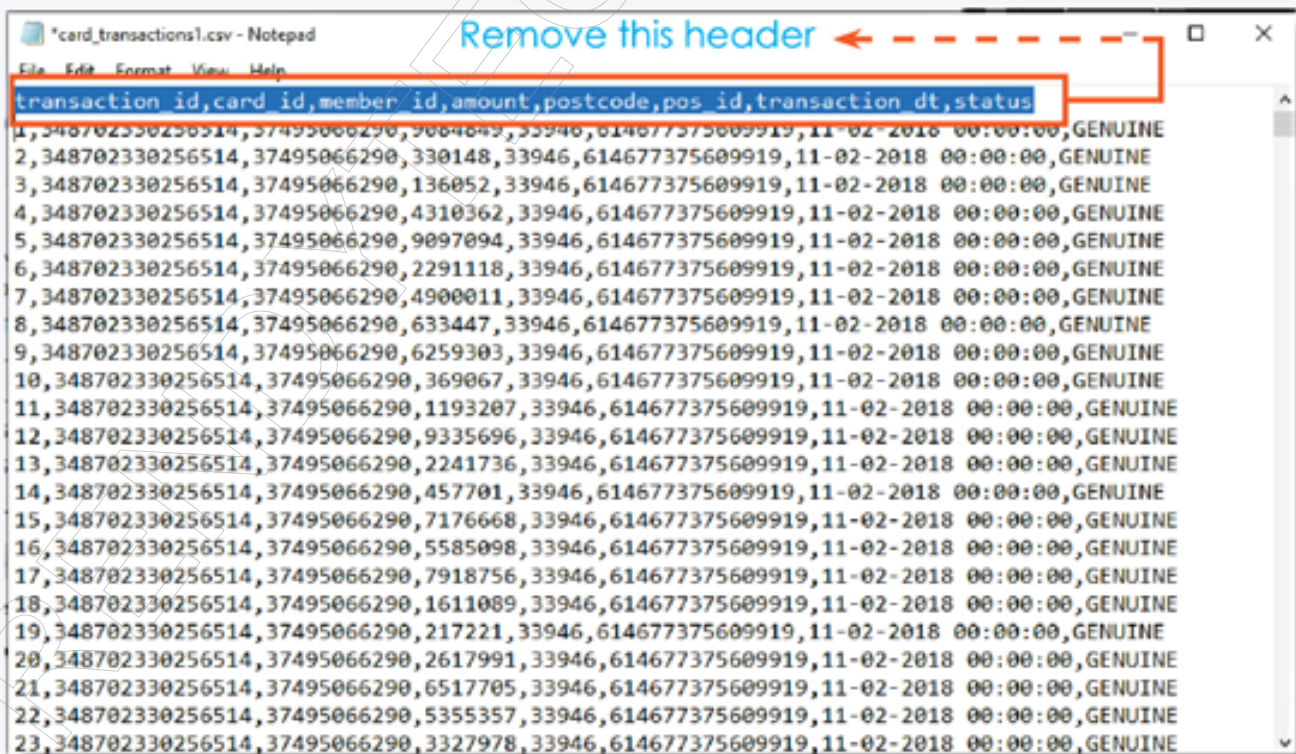
The screenshot shows a terminal window titled "cloudera@quickstart:~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a MySQL prompt "mysql>" followed by the same CREATE TABLE command as above. The command is executed line-by-line, indicated by "->" on each line. After the command, the terminal displays "Query OK, 0 rows affected (0.03 sec)". The prompt "mysql>" is followed by a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
mysql> CREATE TABLE card_transactions_stage (  
-> transaction_id INT(10),  
-> card_id BIGINT,  
-> member_id BIGINT,  
-> amount INT(10),  
-> postcode INT(10),  
-> pos_id BIGINT,  
-> transaction_dt varchar(255),  
-> status varchar(255),  
-> PRIMARY KEY (transaction_id);  
Query OK, 0 rows affected (0.03 sec)  
mysql> █
```


Go to [Google classroom](#) and download the *card_transactions1.csv*:

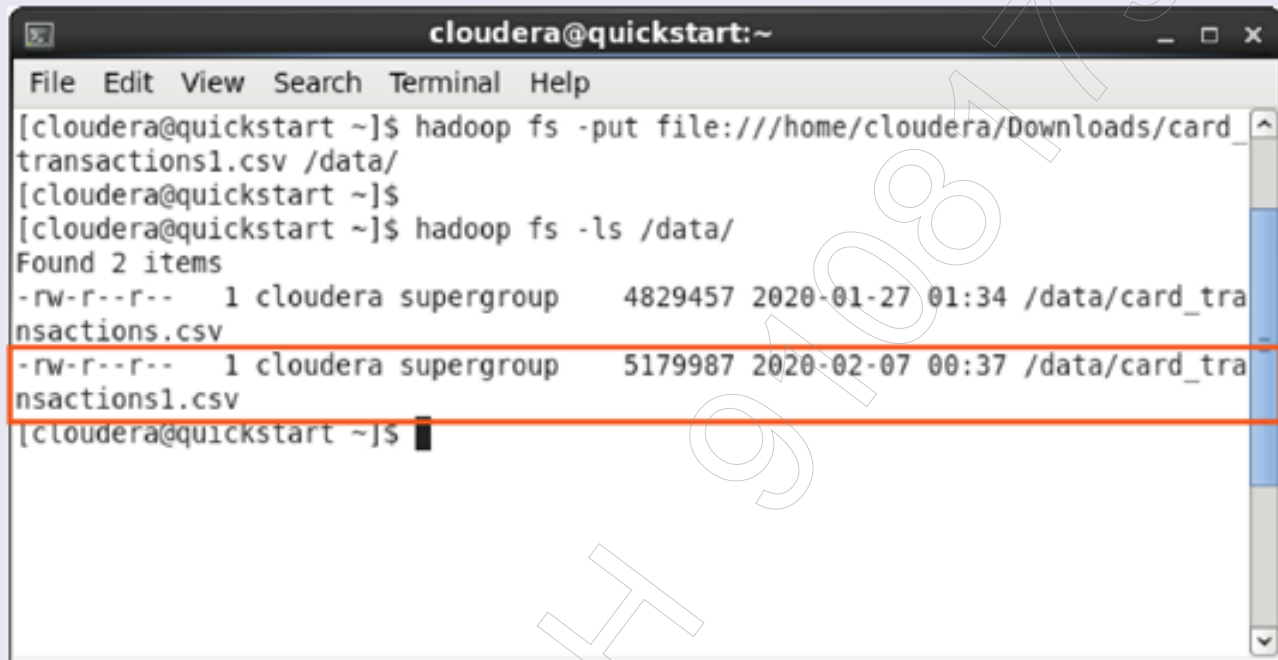


Remove the header from the *card_transactions1.csv* file and save:



Move the *card_transactions1.csv* file to the HDFS data directory:

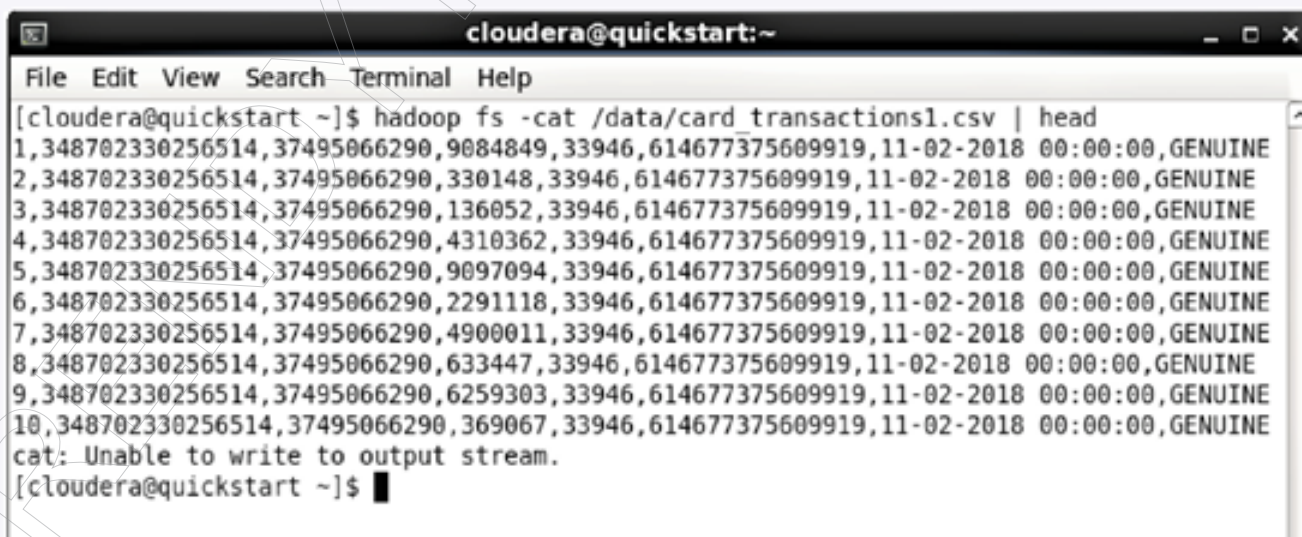
```
hadoop fs -put file:///home/cloudera/Downloads/card_transactions1.csv /data/
```



A terminal window titled 'cloudera@quickstart:~' showing the execution of Hadoop commands. The first command is 'hadoop fs -put file:///home/cloudera/Downloads/card_transactions1.csv /data/'. The second command is 'hadoop fs -ls /data/'. The output shows two files in the directory: 'card_transactions.csv' and 'card_transactions1.csv'. The second file is highlighted with a red box.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -put file:///home/cloudera/Downloads/card_transactions1.csv /data/  
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ hadoop fs -ls /data/  
Found 2 items  
-rw-r--r-- 1 cloudera supergroup 4829457 2020-01-27 01:34 /data/card_transactions.csv  
-rw-r--r-- 1 cloudera supergroup 5179987 2020-02-07 00:37 /data/card_transactions1.csv  
[cloudera@quickstart ~]$
```

The *card_transactions* table data should look like below:

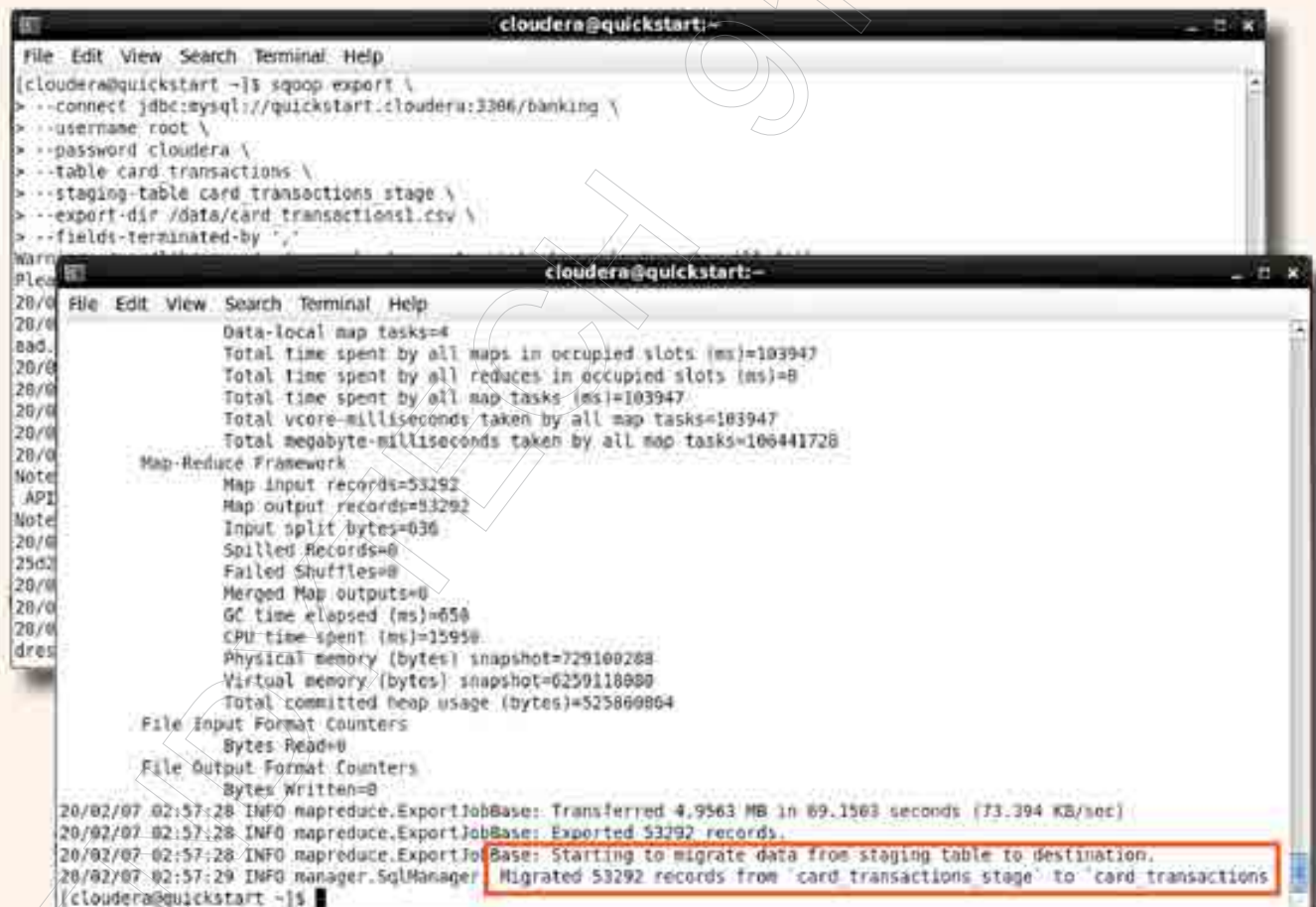


A terminal window titled 'cloudera@quickstart:~' showing the execution of the command 'hadoop fs -cat /data/card_transactions1.csv | head'. The output displays the first 10 lines of the CSV file, which contain transaction data. The last line of the output is 'cat: Unable to write to output stream.'

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -cat /data/card_transactions1.csv | head  
1,348702330256514,37495066290,9084849,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
2,348702330256514,37495066290,330148,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
3,348702330256514,37495066290,136052,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
4,348702330256514,37495066290,4310362,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
5,348702330256514,37495066290,9097094,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
6,348702330256514,37495066290,2291118,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
7,348702330256514,37495066290,4900011,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
8,348702330256514,37495066290,633447,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
9,348702330256514,37495066290,6259303,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
10,348702330256514,37495066290,369067,33946,614677375609919,11-02-2018 00:00:00,GENUINE  
cat: Unable to write to output stream.  
[cloudera@quickstart ~]$
```

Now export the **card_transactions1.csv** from hdfs to mysql through staging table:

```
sqoop export \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table card_transactions \  
--staging-table card_transactions_stage \  
--export-dir /data/card_transactions1.csv \  
--fields-terminated-by ','
```



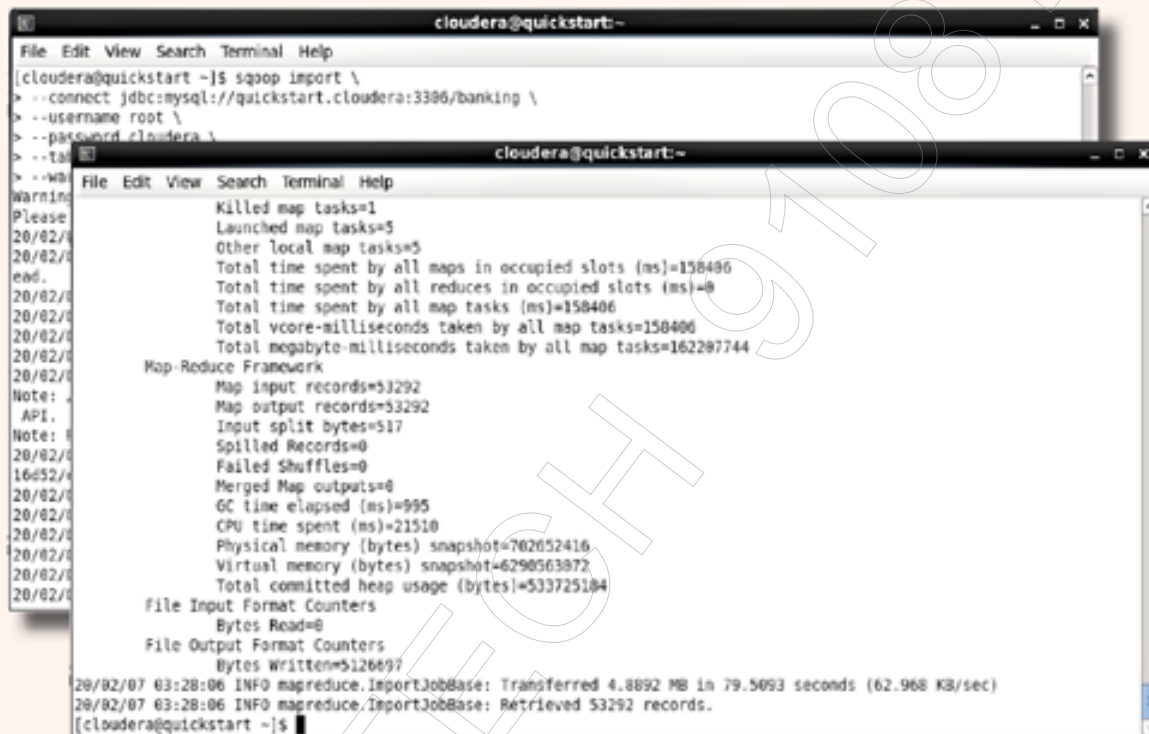
The first screenshot shows the command being executed in a terminal window titled 'cloudera@quickstart:-'. The second screenshot shows the output of the command, which includes various statistics and a summary of the export process. The output is as follows:

```
cloudera@quickstart:-  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ sqoop export \  
> --connect jdbc:mysql://quickstart.cloudera:3306/banking \  
> --username root \  
> --password cloudera \  
> --table card_transactions \  
> --staging-table card_transactions_stage \  
> --export-dir /data/card_transactions1.csv \  
> --fields-terminated-by ','  
Warning: Please use the --verbose option to get more detailed output.  
20/02/07 02:57:28 INFO mapreduce.ExportJobBase: Transferred 4.9563 MB in 69.1563 seconds (73.394 KB/sec)  
20/02/07 02:57:28 INFO mapreduce.ExportJobBase: Exported 53292 records.  
20/02/07 02:57:28 INFO mapreduce.ExportJobBase: Starting to migrate data from staging table to destination.  
20/02/07 02:57:29 INFO manager.SqlManager: Migrated 53292 records from 'card_transactions_stage' to 'card_transactions'  
[cloudera@quickstart ~]$
```

Map-Reduce Framework
Map input records=53292
Map output records=53292
Input split bytes=636
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=650
CPU time spent (ms)=15950
Physical memory (bytes) snapshot=729100288
Virtual memory (bytes) snapshot=6250118080
Total committed heap usage (bytes)=525860864
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=0

Import back the `card_transactions1.csv` from mysql to hdfs:

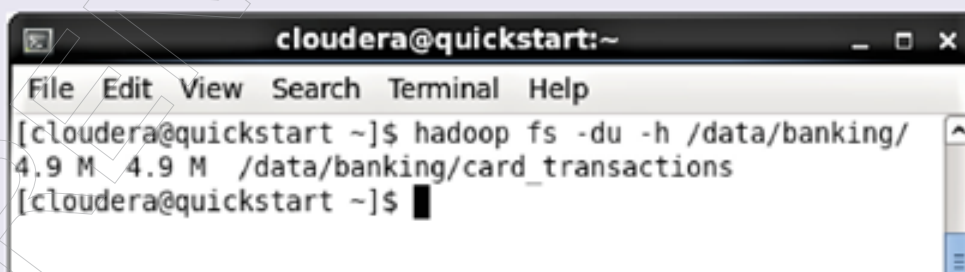
```
scoop import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root --password cloudera \  
--table card_transactions \  
--warehouse-dir /data/banking
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ scoop import \  
> --connect jdbc:mysql://quickstart.cloudera:3306/banking \  
> --username root \  
> --password cloudera \  
> --table card_transactions \  
> --warehouse-dir /data/banking  
Warning: Killed map tasks=1  
Please: Launched map tasks=5  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Other local map tasks=5  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Total time spent by all maps in occupied slots (ms)=158406  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Total time spent by all reduces in occupied slots (ms)=0  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Total time spent by all map tasks (ms)=158406  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Total vcore-milliseconds taken by all map tasks=158406  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Total megabyte-milliseconds taken by all map tasks=162287744  
Map-Reduce Framework  
Map input records=53292  
Map output records=53292  
Input split bytes=517  
Spilled Records=0  
Failed Shuffles=0  
Merged Map outputs=0  
GC time elapsed (ms)=995  
CPU time spent (ms)=21510  
Physical memory (bytes) snapshot=762652416  
Virtual memory (bytes) snapshot=46298563872  
Total committed heap usage (bytes)=533725184  
File Input Format Counters  
Bytes Read=0  
File Output Format Counters  
Bytes Written=532692  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Transferred 4.8892 MB in 79.5093 seconds (62.968 KB/sec)  
20/02/07 03:28:06 INFO mapreduce.ImportJobBase: Retrieved 53292 records.  
[cloudera@quickstart ~]$
```

To check the size of the imported data:

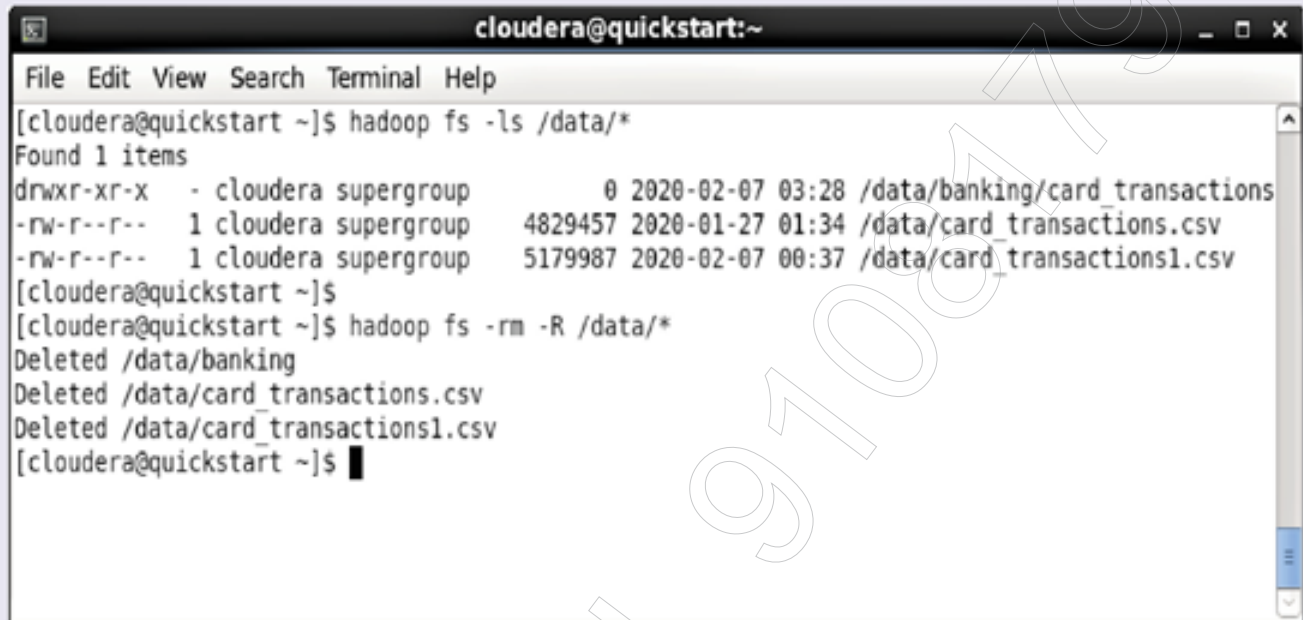
```
hadoop fs -du -h /data/banking/
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -du -h /data/banking/  
4.9 M 4.9 M /data/banking/card_transactions  
[cloudera@quickstart ~]$
```


Command to delete directories from hdfs:

```
hadoop fs -rm -R /data/*
```



The screenshot shows a terminal window titled 'cloudera@quickstart:~'. The user runs the command 'hadoop fs -ls /data/*'. The output shows three items: a directory '/data/banking/card_transactions' and two files '/data/card_transactions.csv' and '/data/card_transactions1.csv'. Then, the user runs 'hadoop fs -rm -R /data/*'. The output shows three lines of 'Deleted' messages for the same paths. The terminal window has a menu bar with 'File Edit View Search Terminal Help' and a scrollbar on the right.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -ls /data/*  
Found 1 items  
drwxr-xr-x  - cloudera supergroup          0 2020-02-07 03:28 /data/banking/card_transactions  
-rw-r--r--  1 cloudera supergroup    4829457 2020-01-27 01:34 /data/card_transactions.csv  
-rw-r--r--  1 cloudera supergroup    5179987 2020-02-07 00:37 /data/card_transactions1.csv  
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ hadoop fs -rm -R /data/*  
Deleted /data/banking  
Deleted /data/card_transactions.csv  
Deleted /data/card_transactions1.csv  
[cloudera@quickstart ~]$
```

Import data from mysql to hdfs in compress form:

```
sqoop import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table card_transactions \  
--warehouse-dir /data/banking \  
--compress
```


Check again the size of the imported data in compress form:

```
hadoop fs -du -h /data/banking/
```



The screenshot shows a terminal window titled 'cloudera@quickstart:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the command '[cloudera@quickstart ~]\$ hadoop fs -du -h /data/banking/' being executed, followed by the output '1.3 M 1.3 M /data/banking/card_transactions'. The prompt '[cloudera@quickstart ~]\$' is shown again with a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop fs -du -h /data/banking/  
1.3 M 1.3 M /data/banking/card_transactions  
[cloudera@quickstart ~]$
```

Sqoop Incremental

Create a table in mysql under banking database:

```
create table member_details(  
card_id BIGINT,  
member_id BIGINT,  
member_joining_dt timestamp,  
card_purchase_dt varchar(255),  
country varchar(255),  
city varchar(255),  
PRIMARY KEY (card_id));
```

Create a staging table in mysql under banking database:

```
create table member_details_stage as  
select * from member_details where 1=0;
```

Create a directory in hdfs and put the *cardmembers.csv* data in it:

```
hadoop fs -mkdir /data/banking/card_member
```

```
hadoop fs -put file:///home/cloudera/Downloads/  
cardmembers.csv /data/banking/card_member/
```

Export the data to the created table (*member_details*) in mysql through staging table of banking database from hdfs:

```
sqoop export \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table member_details \  
--staging-table member_details_stage \  
--export-dir /data/banking/card_member \  
--fields-terminated-by ','
```

Create a Sqoop job to import data from mysql to hdfs:

```
sqoop job \  
--create job_banking_member_details \  
--import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table member_details \  
--warehouse-dir /data/banking \  
--incremental append \  
--check-column card_id \  
--last-value 0
```

List available Sqoop jobs:

```
sqoop job --list
```

Execute a Sqoop job:

```
sqoop job --exec job_banking_member_details
```

To show last incremental parameter:

```
sqoop job --show job_banking_member_details | grep incremental
```

Delete a Sqoop job:

```
sqoop job --delete job_banking_member_details
```

To create a password file:

```
echo -n "cloudera" >> password-file
```

Create a Sqoop job with password file:

```
sqoop job \  
--create job_banking_member_details \  
-- import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password-file file:///home/cloudera/password-file \  
--table member_details \  
--warehouse-dir /data/banking \  
--incremental append \  
--check-column card_id \  
--last-value 0
```


Execute a Sqoop job:

```
sqoop job --exec job_banking_member_details
```

To check the output files:

```
hadoop fs -cat /data/banking/member_score/* | wc -l
```

```
CREATE TABLE member_score(  
member_id BIGINT,  
score INT(3),  
PRIMARY KEY (member_id));
```

```
CREATE TABLE member_score_stage(  
member_id BIGINT,  
score INT(3),  
PRIMARY KEY (member_id));
```

```
sqoop export \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table member_score \  
--staging-table member_score_stage \  
--export-dir /data/banking/member_score \  
--fields-terminated-by ','
```

```
sqoop import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table member_score \  
--delete-target-dir \  
--warehouse-dir /data/banking
```

java cryptography encryption key store

reference link: <https://www.ericlin.me/2015/06/securely-managing-passwords-in-sqoop/>

<https://www.pixelstech.net/article/1420439432-Different-types-of-keystore-in-Java----JCEKS>

```
hadoop credential create mysql.banking.password -provider  
jceks://hdfs/user/sumitm/mysql.password.jceks
```

```
hadoop fs -cat /user/sumitm/mysql.password.jceks
```

```
sqoop eval \  
-Dhadoop.security.credential.provider.path=jceks://hdfs/  
user/sumitm/mysql.password.jceks \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password-alias mysql.banking.password \  
--query "select count(1) from member_score"
```

```
sqoop import \  
--connect jdbc:mysql://quickstart.cloudera:3306/banking \  
--username root \  
--password cloudera \  
--table card_transactions \  
--hive-import \  
--hive-database banking \  
--compress
```




5 Star Google Rated Big Data Course

LEARN FROM THE EXPERT



9108179578

Call for more details

Follow US

Trainer Mr. Sumit Mittal

LinkedIn <https://www.linkedin.com/in/bigdatabysumit/>

Website <https://trendytech.in/courses/big-data-online-training/>

Phone 9108179578

Email trendytech.sumit@gmail.com

Youtube TrendyTech

Twitter @BigdataBySumit

Instagram bigdatabysumit

Facebook <https://www.facebook.com/trendytech.in/>

