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.

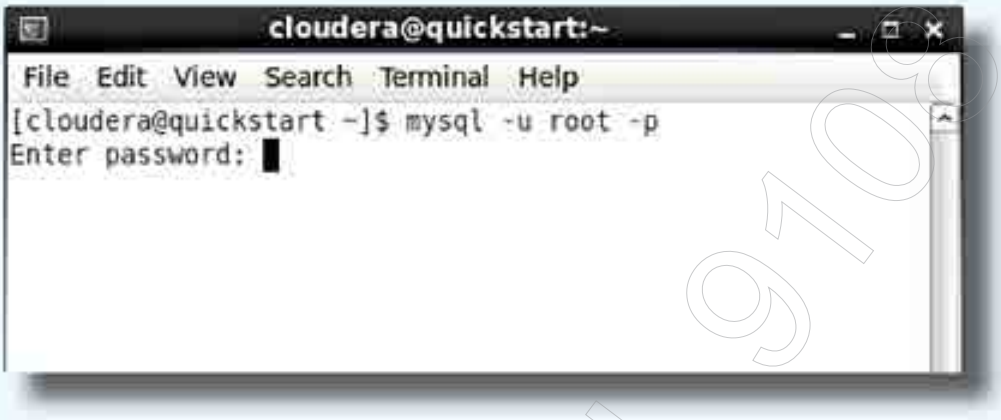
Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written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

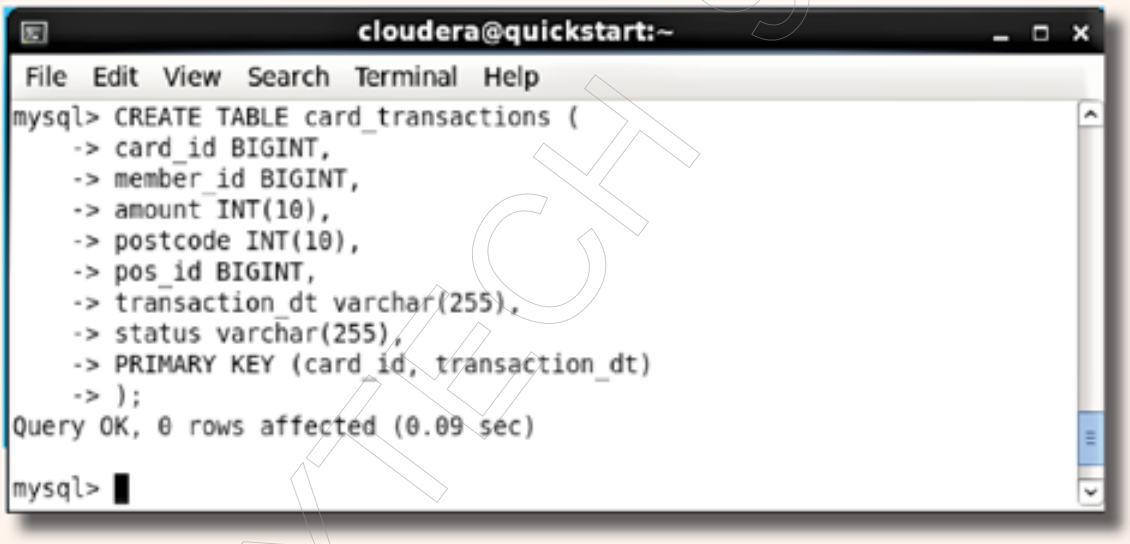
```
create database banking;

use banking;
```

TRENDYTECH 9108179578

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

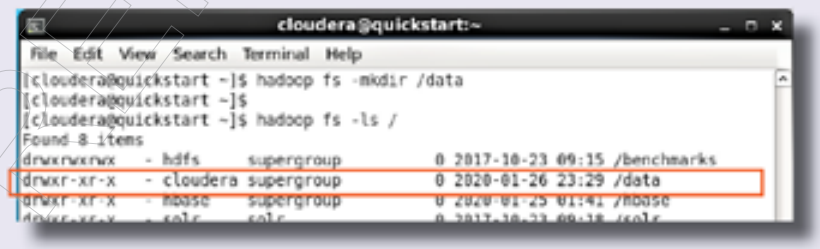
```
CREATE TABLE card_transactions (
card_id BIGINT,
member_id BIGINT,
amount INT(10),
postcode INT(10),
pos_id BIGINT,
transaction_dt varchar(255),
status varchar(255),
PRIMARY KEY (card_id, transaction_dt)
);
```

composite primary key



Create a directory inside HDFS:

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

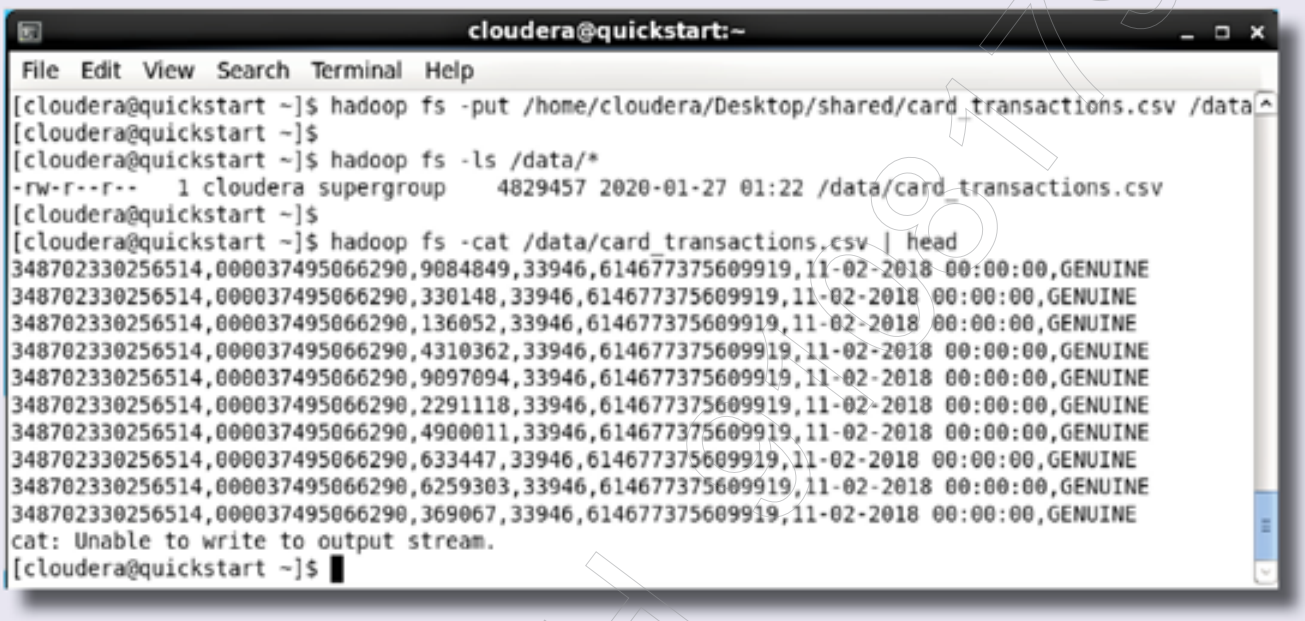


No need to create, if already available

TRENDYTECH 9108179578

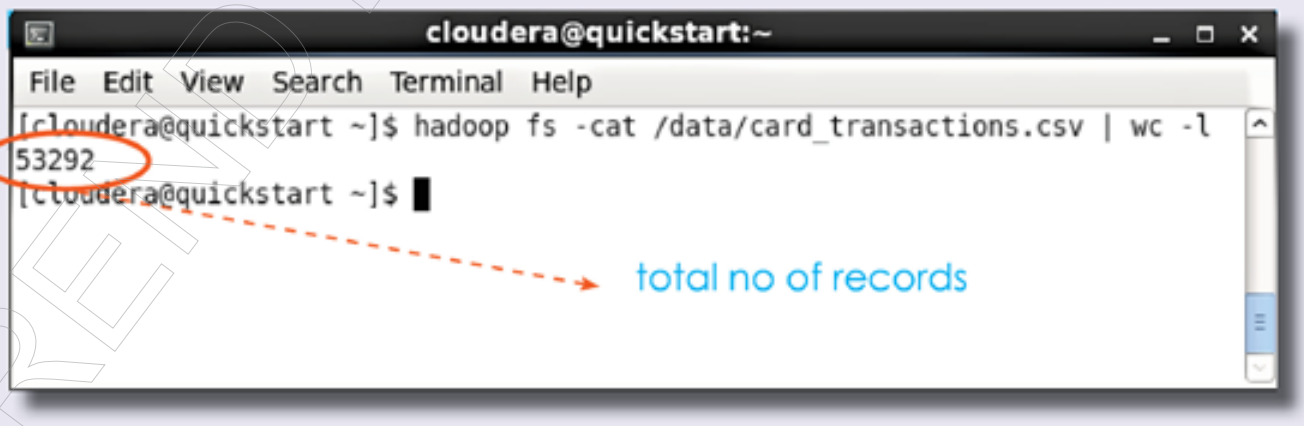
Move the csv file (*card_transactions*) to the HDFS *data* directory:

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



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

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

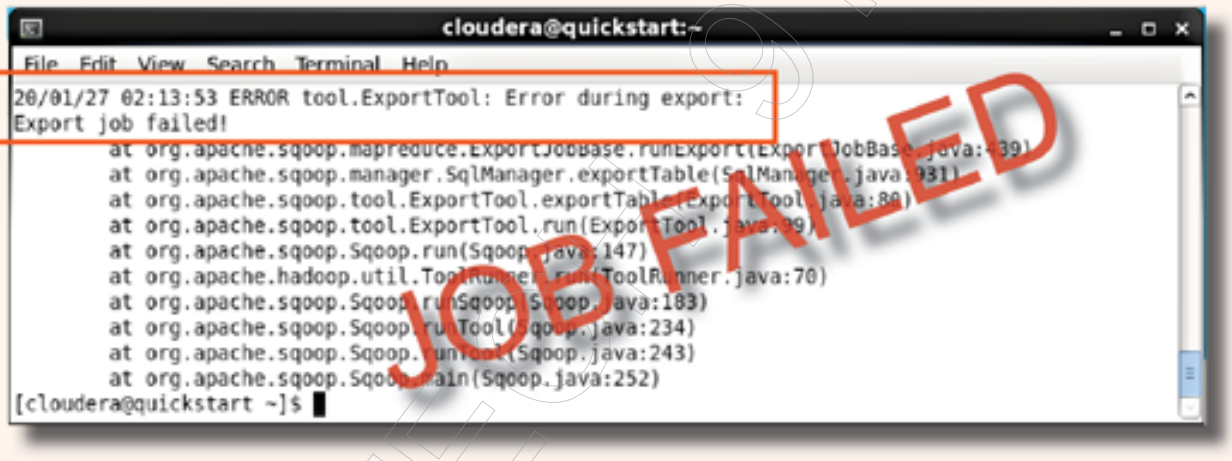


TRENDYTECH

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`.

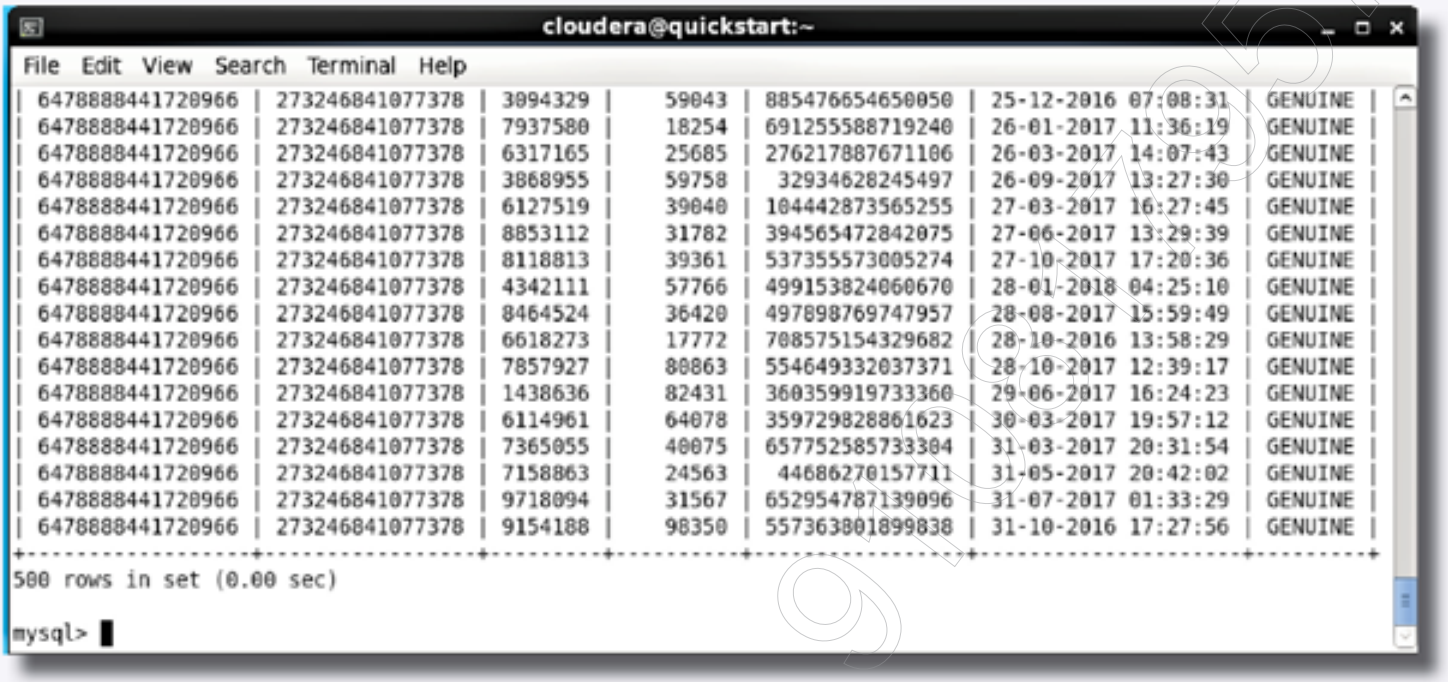
```
sqoop export \
--connect jdbc:mysql://quickstart.cloudera:3306/banking \
--username root \
--password cloudera \
--table card_transactions \
--export-dir /data/card_transactions.csv \
--fields-terminated-by ','
```



Note: The Job will fail

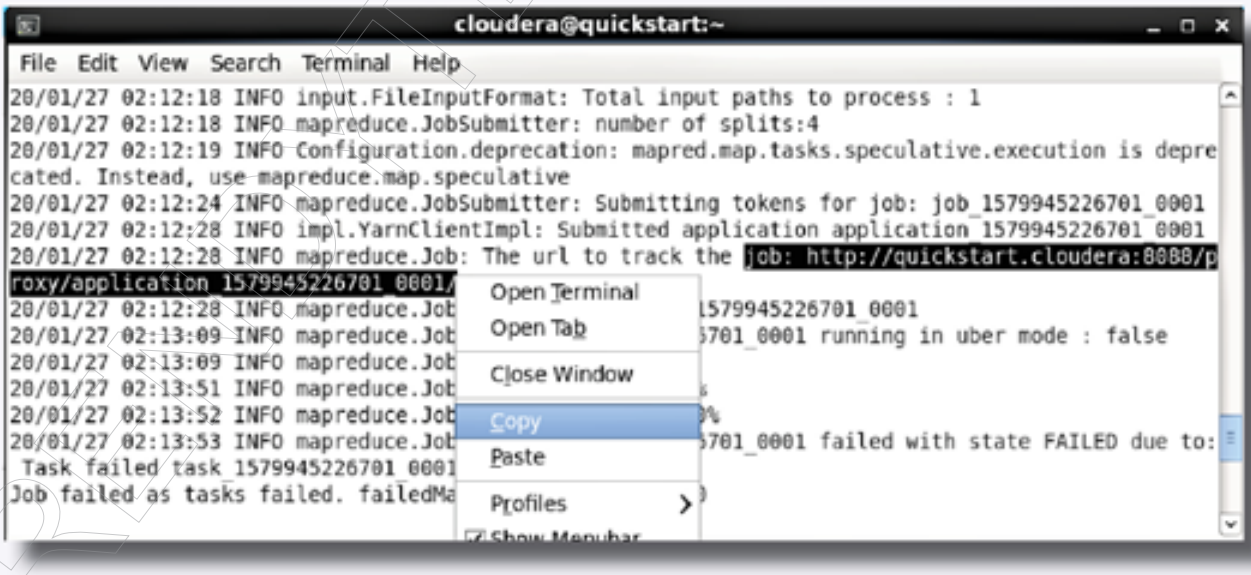
TRENDYTECH 9108179578

The *card_transactions* table data shoud look like below:



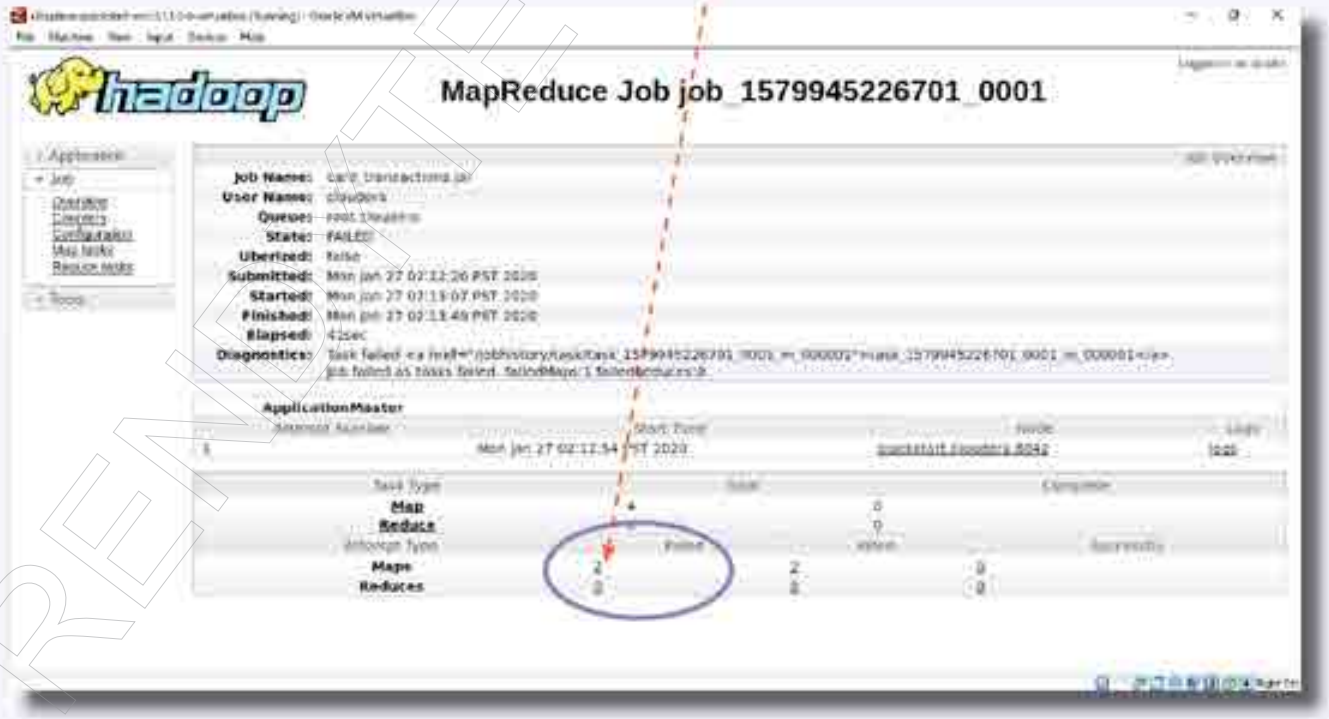
How to track the failed job:

- Go to the log and **find** The url to track the job
- **Copy** the given url



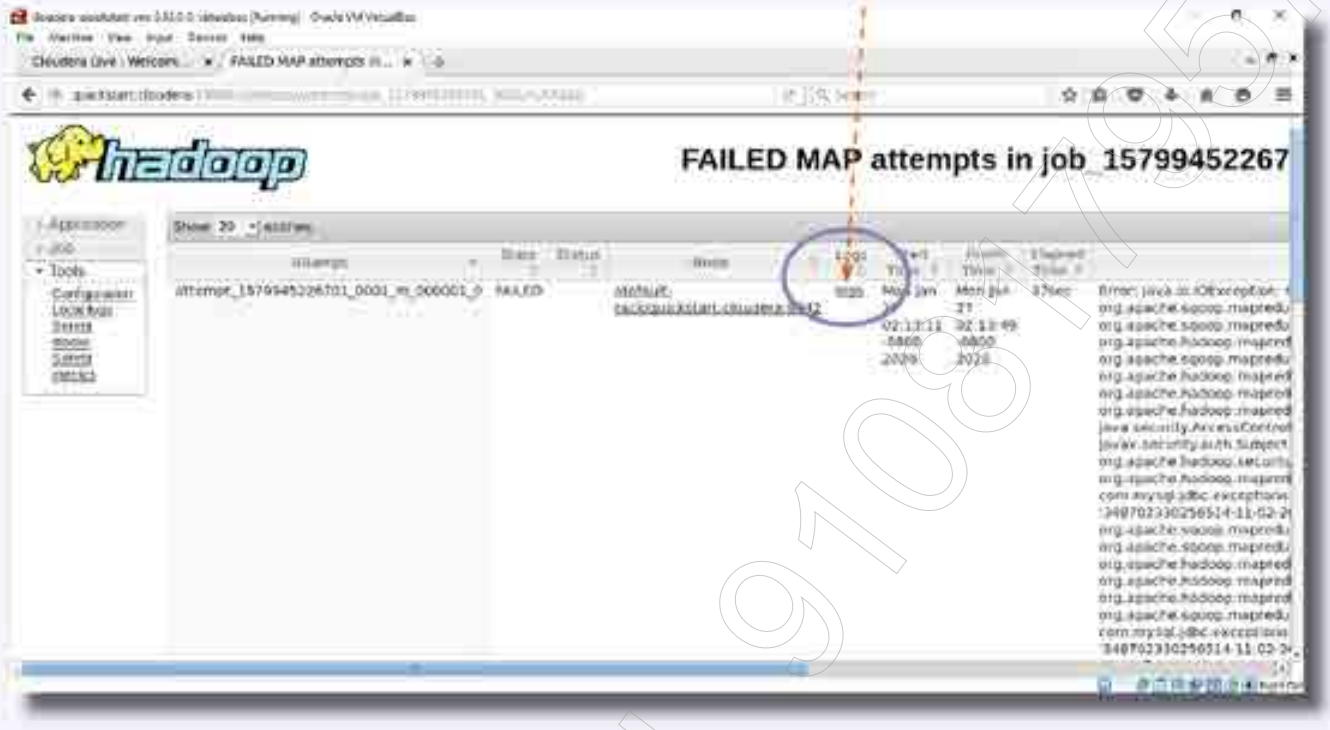
- Go to the web browser and **paste** the url and enter

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



TRENDYTECH 9098179578

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

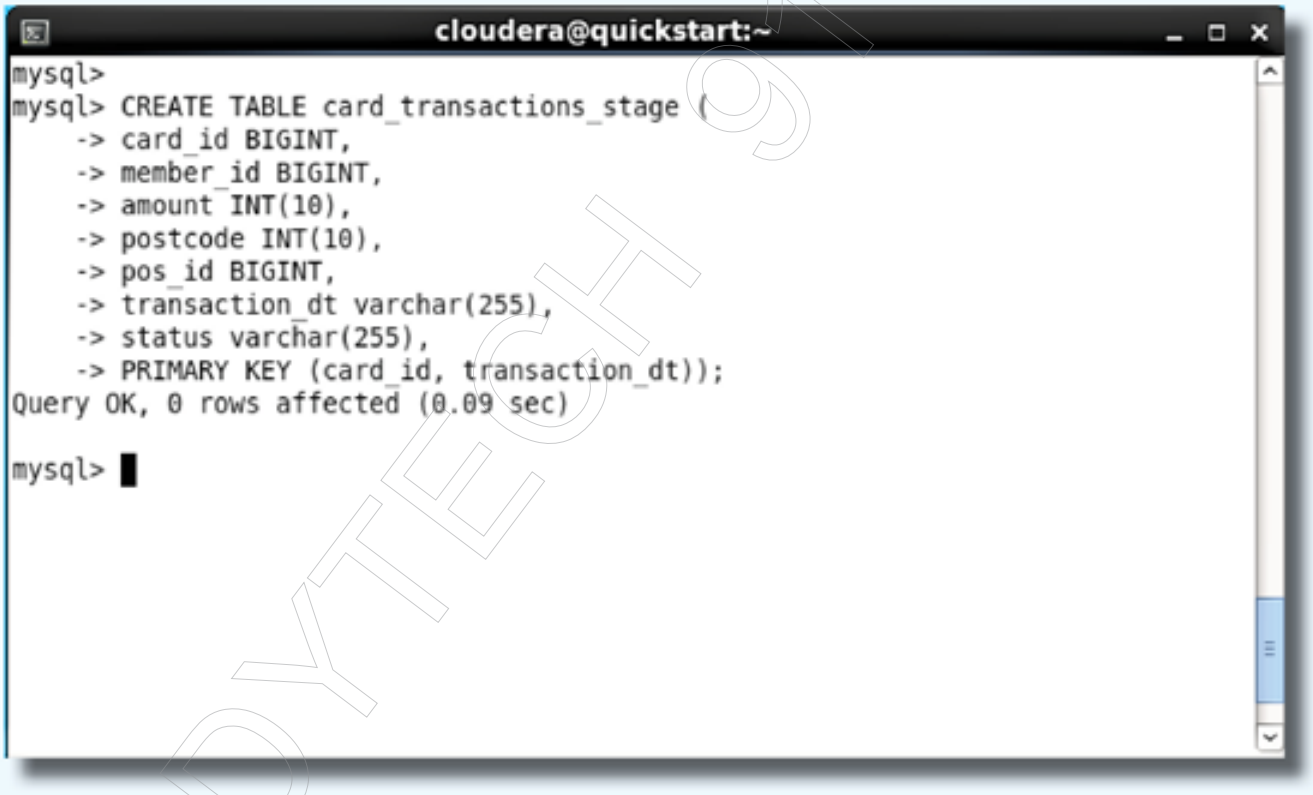


- In the Log window, you will find the actual **errors**

TRENDYTECH 9108179548

Create a staging table *card_transactions_stage* in mysql:

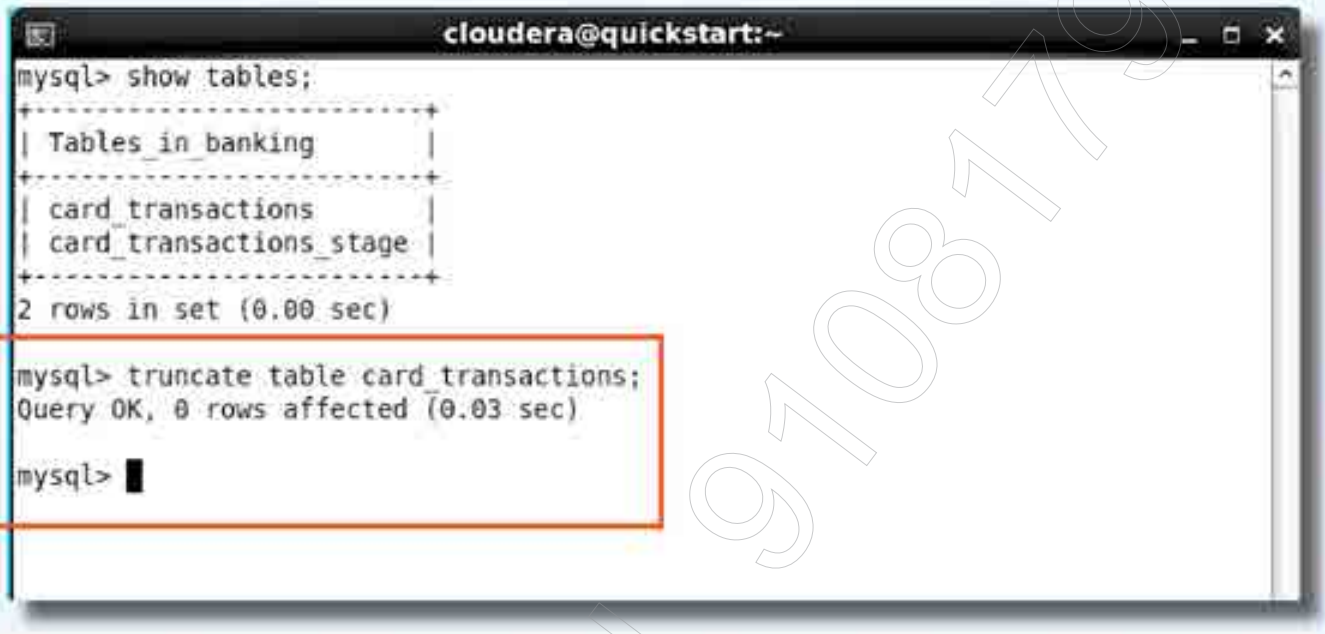
```
CREATE TABLE card_transactions_stage (
card_id BIGINT,
member_id BIGINT,
amount INT(10),
postcode INT(10),
pos_id BIGINT,
transaction_dt varchar(255),
status varchar(255),
PRIMARY KEY (card_id, transaction_dt));
```



TRENDYTECH 9108179578

Truncate *card_transactions* table from mysql:

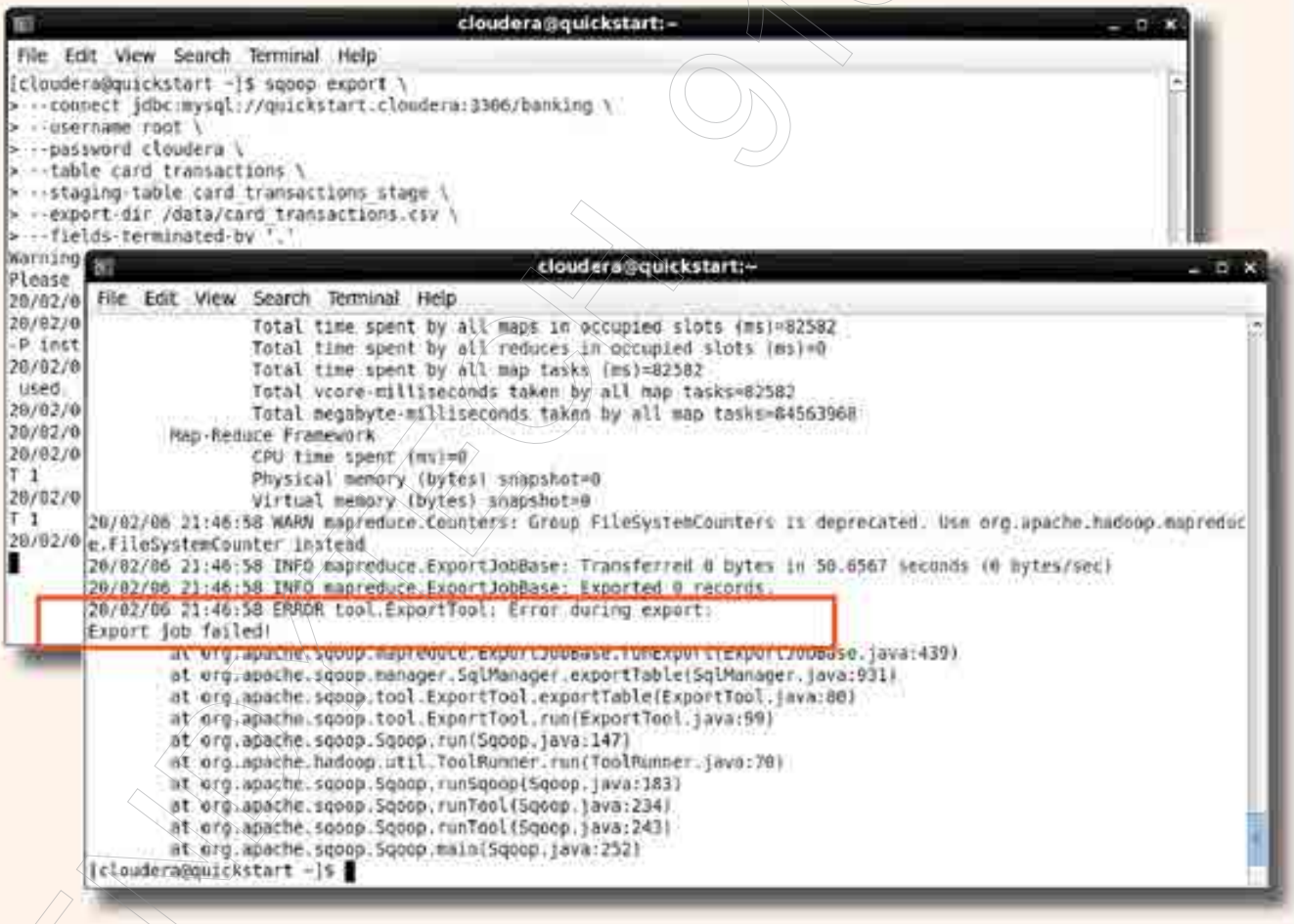
```
truncate table card_transactions;
```



TRENDYTECH 9108179578

Export data from hdfs to mysql using staging table:

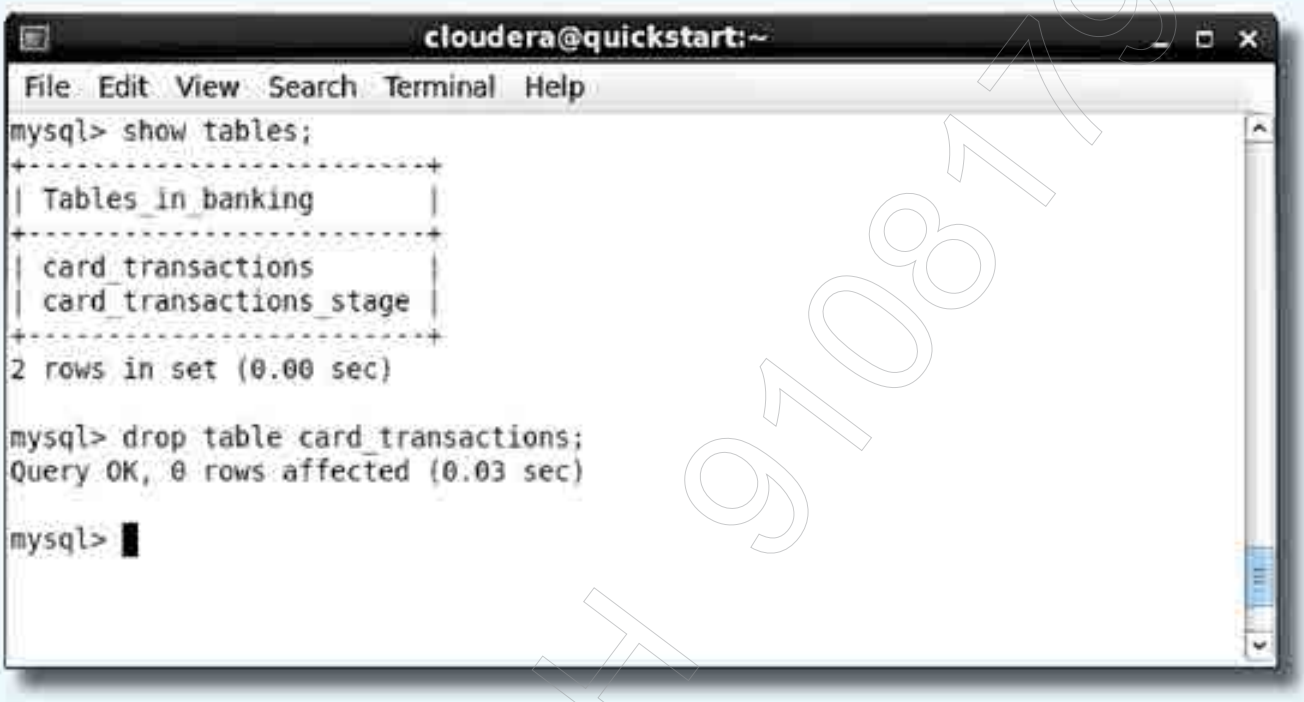
```
sqoop export \
--connect jdbc:mysql://quickstart.cloudera:3306/banking \
--username root \
--password cloudera \
--table card_transactions \
--staging-table card_transactions_stage \
--export-dir /data/card_transactions.csv \
--fields-terminated-by ','
```



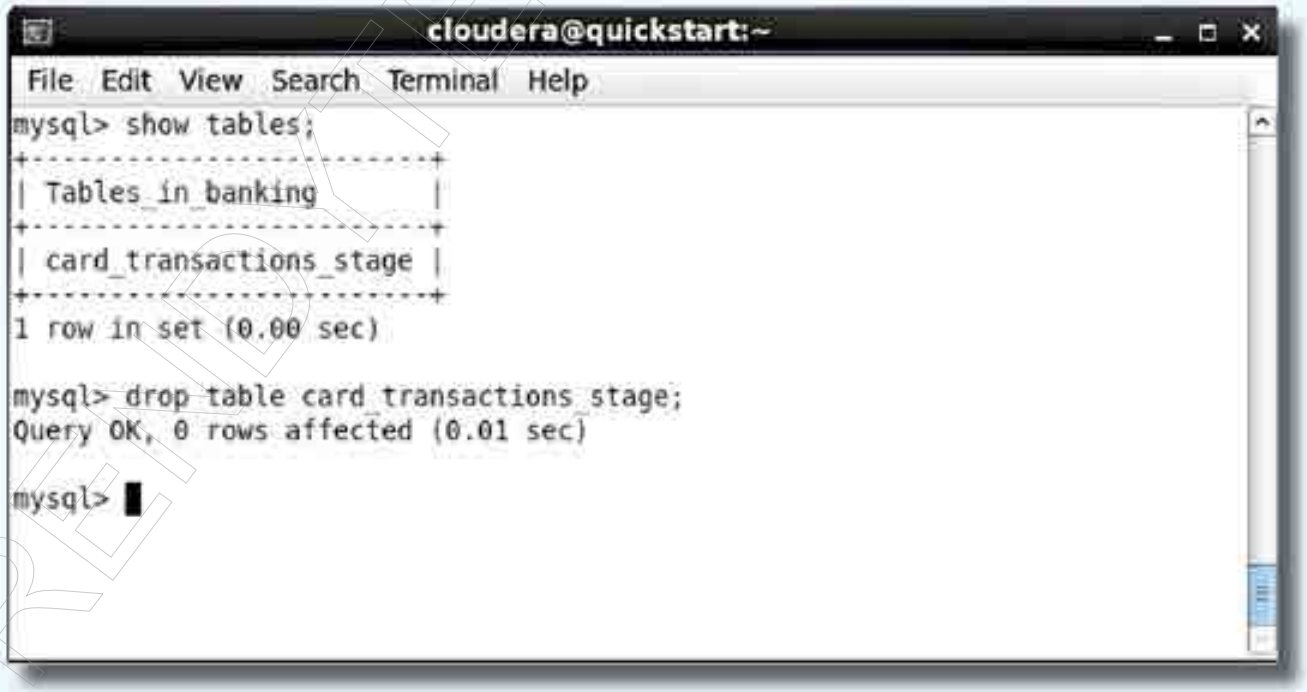
TRENDYTECH 9108179578

Drop both the staging and the actual table from mysql:

```
drop table card_transactions;
```




```
drop table card_transactions_stage;
```



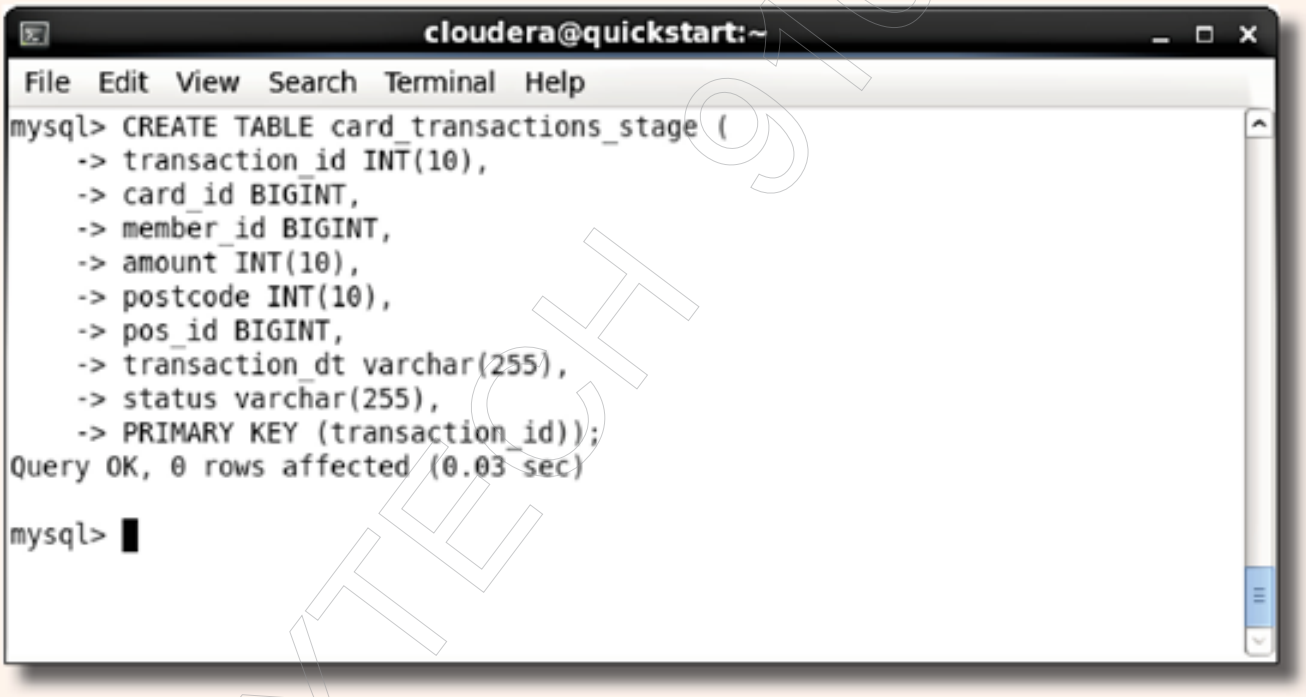
TRENDYTECH 9108179578

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));
```

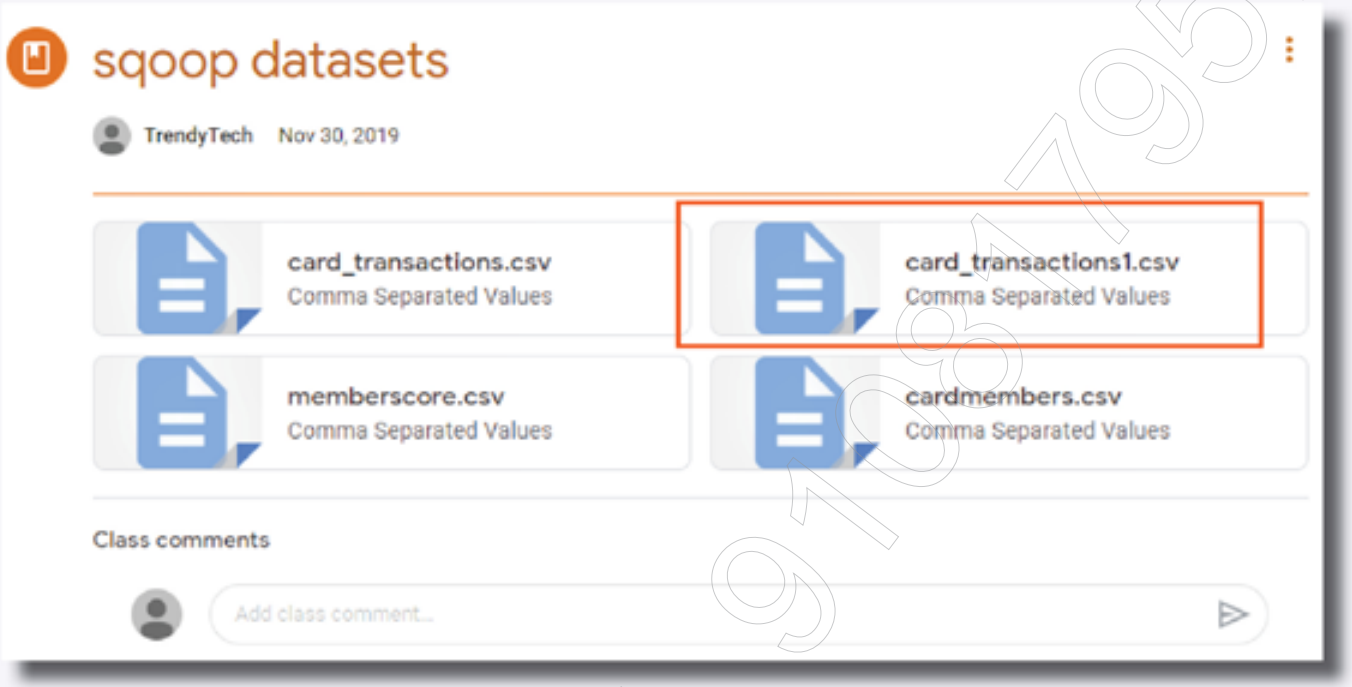
TRENDYTECH 9108179578

```
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));
```

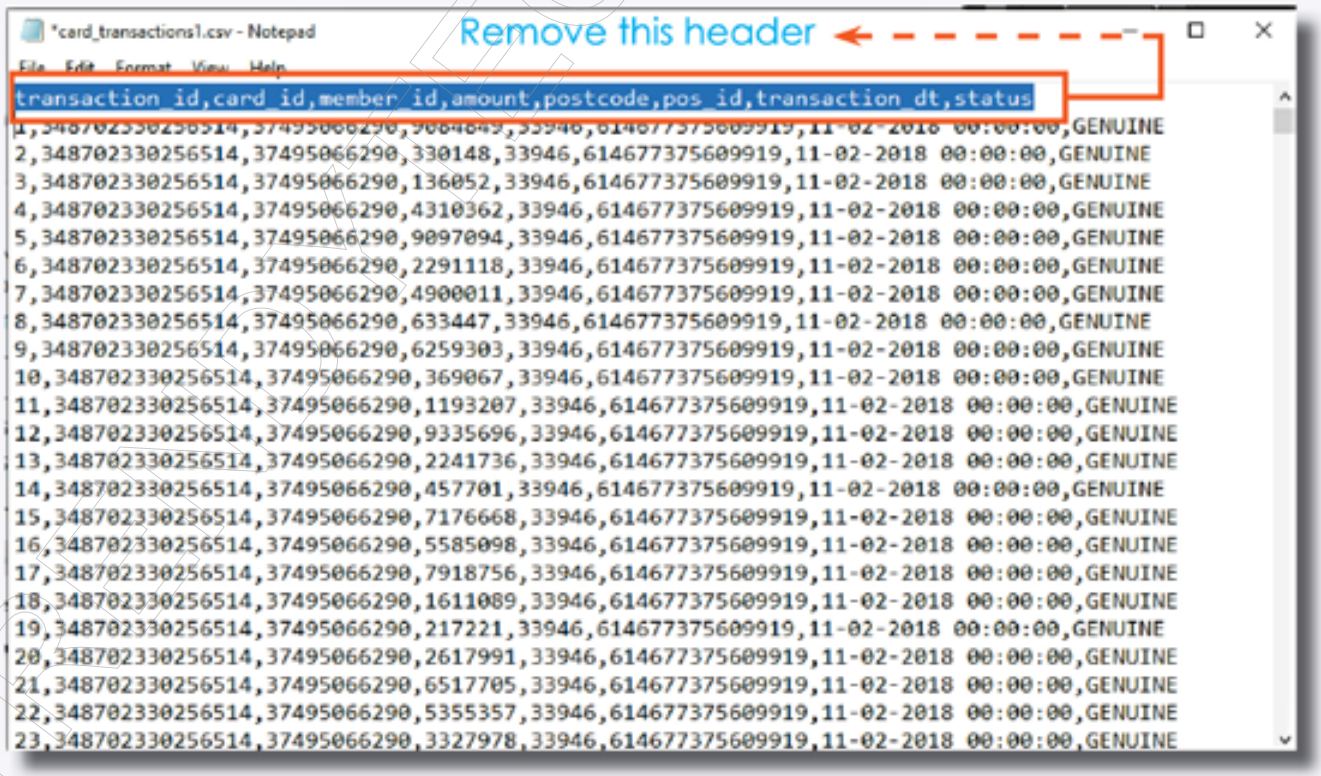


TRENDYTECH 9108179578

Go to Google classroom and download the *card_transactions1.csv*:



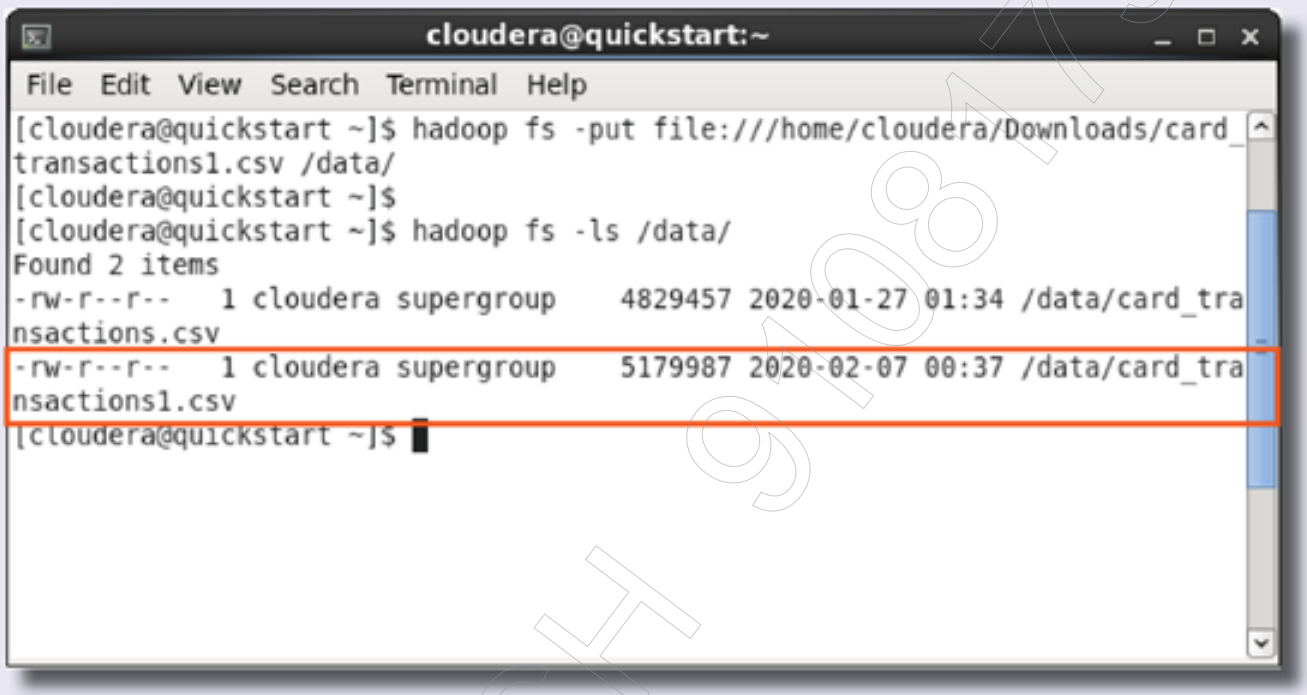
Remove the header from the *card_transactions1.csv* file and save:



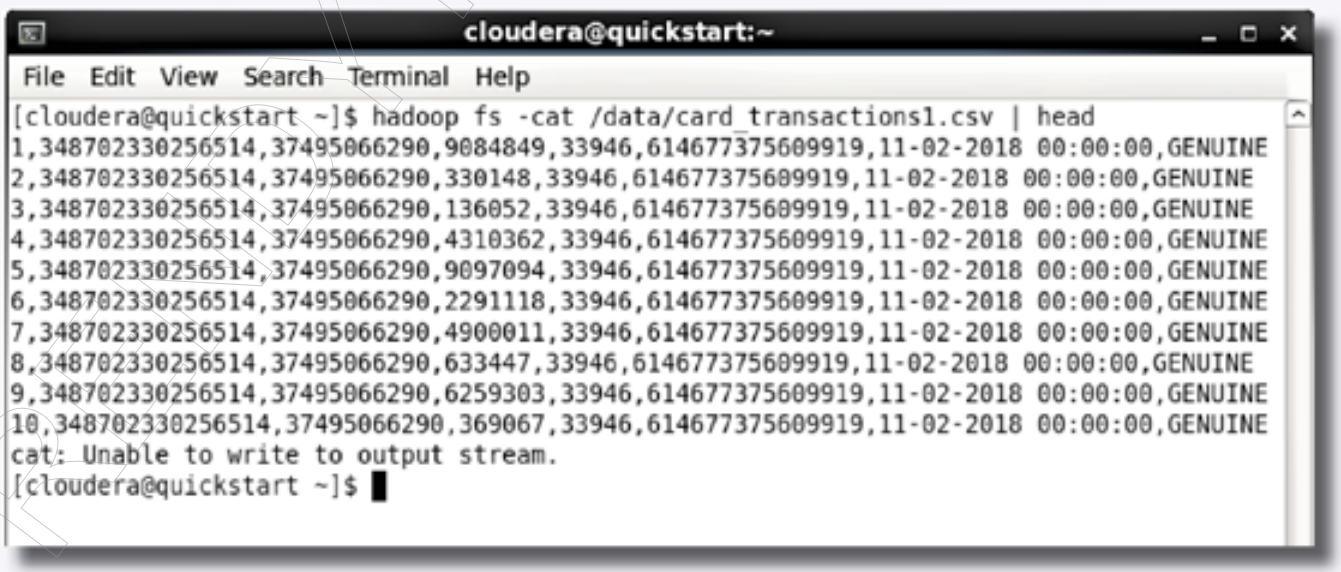
TRENDYTECH 9108179578

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

```
hadoop fs -put file:///home/cloudera/Downloads/card_
transactions1.csv /data/
```



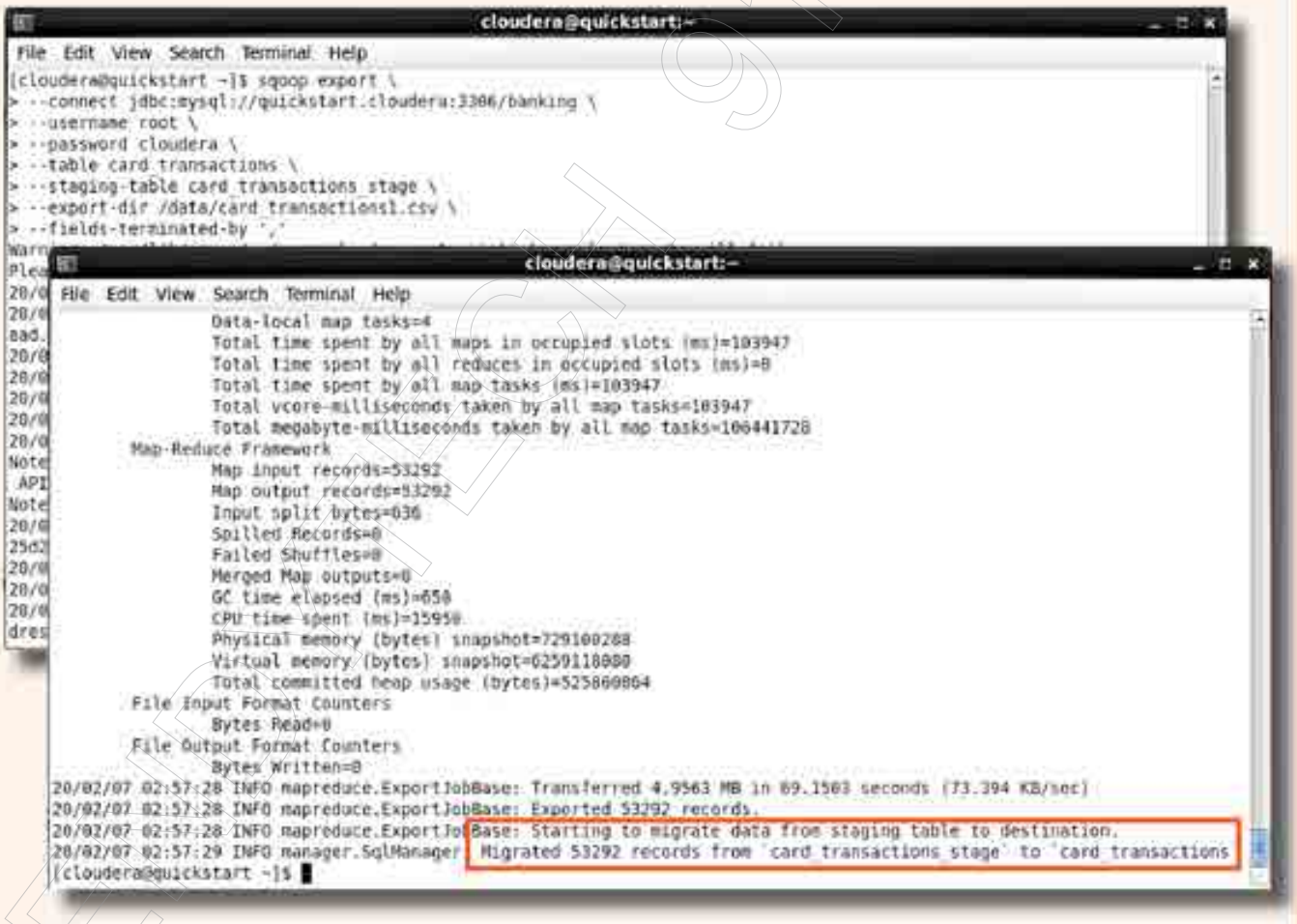
The *card_transactions* tabele data shoud look like below:



TRENDYTECH 9108179578

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

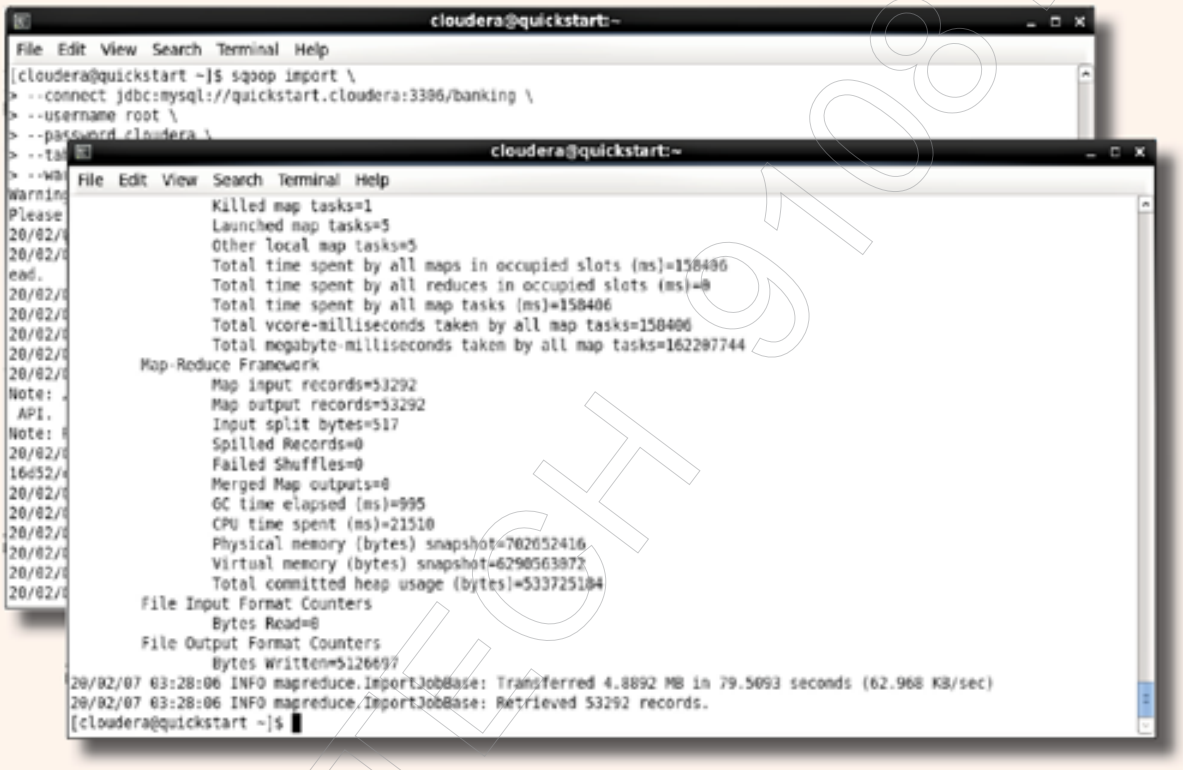
```
sqoop export \
--connect jdbc:mysql://quickstart.cloudera:3306/banking \
--username root \
--password cloudera \
--table card_transactions \
--staging-table card_transactions_stage \
--export-dir /data/card_transactions1.csv \
--fields-terminated-by ','
```



TRENDYTECH 9108179578

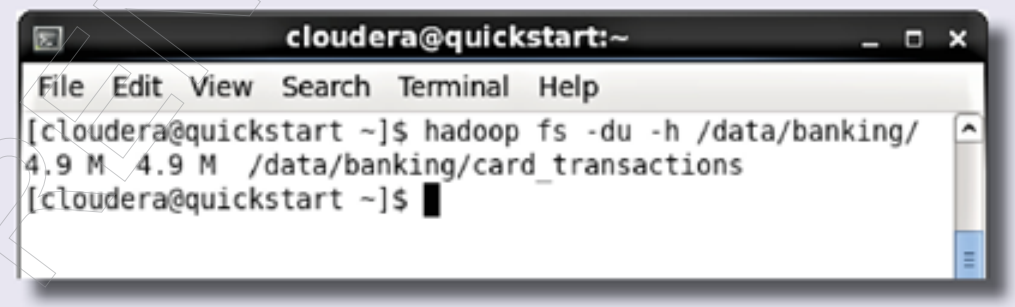
Import back the *card_transactions1.csv* from mysql to hdfs:

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



To check the size of the imported data:

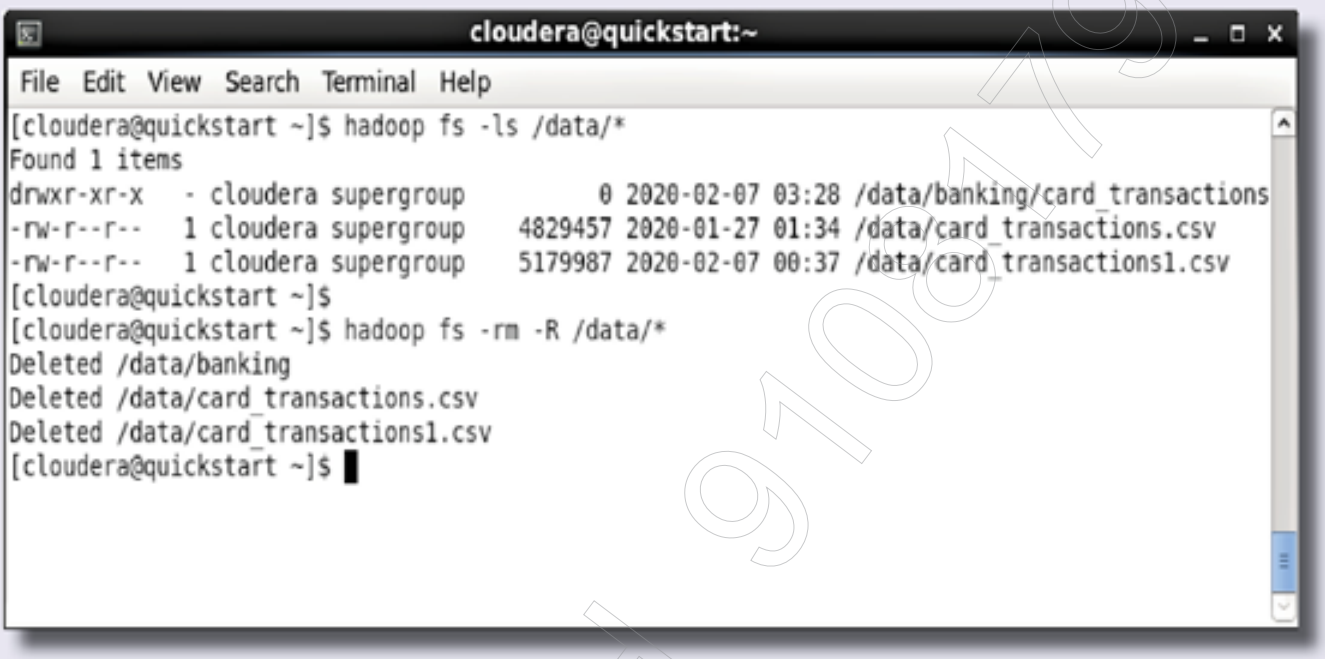
```
hadoop fs -du -h /data/banking/
```



TRENDYTECH 9108179578

Command to delete directories from hdfs:

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



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
```

TRENDYTECH 9108179578

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

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

TRENDYTECH 9108179578

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/
```

TRENDYTECH 9108179578

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
```

TRENDYTECH 9108179578

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
```

TRENDYTECH 9108179578

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 ','
```

TRENDYTECH 9108179578

```
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"
```

TRENDYTECH 9108179578

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

TRENDYTECH 9108179578

Google REVIEWS

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/