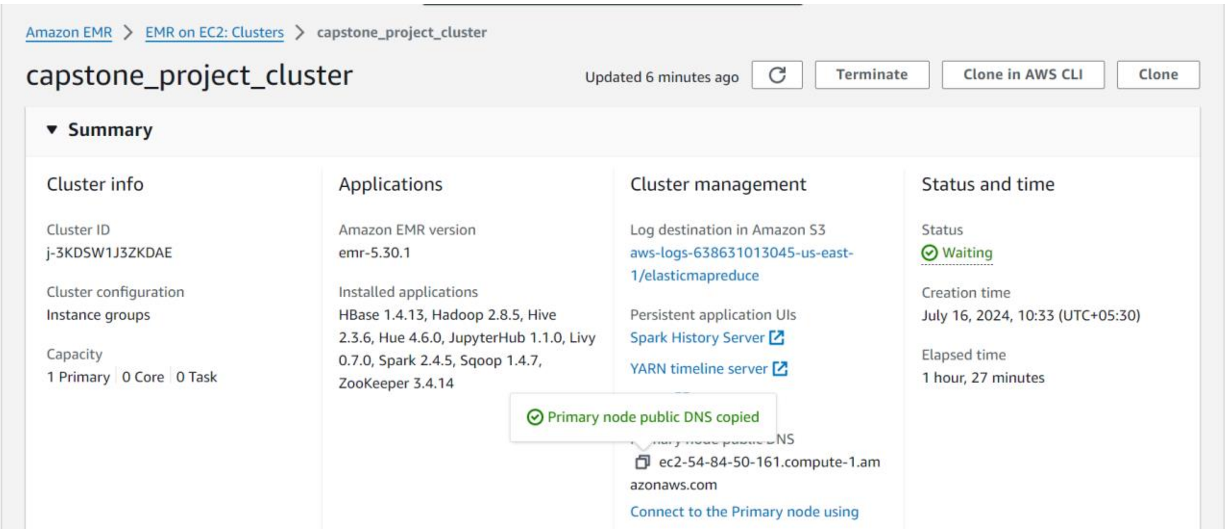
Mid-Submission – Logic Explanation

Explanation of the solution to the batch layer problem

1. In order to complete below tasks, I have created EMR cluster with Hadoop, Sqoop, Hive, Hbase, Hue , Jupyterhub, Livy, Spark and Zookeeper Root device EBS volume size as 20 GB
   • Task 1: Load the transactions history data (card_transactions.csv) in a NoSQL database.
   • Task 2: Ingest the relevant data from AWS RDS to Hadoop.
   • Task 3: Create a look-up table with columns specified earlier in the problem statement.
   • Task 4: After creating the table, you need to load the relevant data in the lookup table.

EMR Cluster Configuration:



Amazon EMR > EMR on EC2: Clusters > capstone_project_cluster
Updated 6 minutes ago
Terminate
Clone in AWS CLI
Clone
▼ Summary

capstone_project_cluster

Cluster info

Cluster ID
j-3KDSW1J3ZKDAE

Cluster configuration
Instance groups

Capacity
1 Primary | 0 Core | 0 Task

Applications

Amazon EMR version
emr-5.30.1

Installed applications
HBase 1.4.13, Hadoop 2.8.5, Hive 2.3.6, Hue 4.6.0, JupyterHub 1.1.0, Livy 0.7.0, Spark 2.4.5, Sqoop 1.4.7, ZooKeeper 3.4.14

Cluster management

Log destination in Amazon S3
aws-logs-638631013045-us-east-1/elasticmapreduce

Persistent application UIs
Spark History Server
YARN timeline server

Primary node public DNS copied

Primary node public DNS
ec2-54-84-50-161.compute-1.amazonaws.com
Connect to the Primary node using

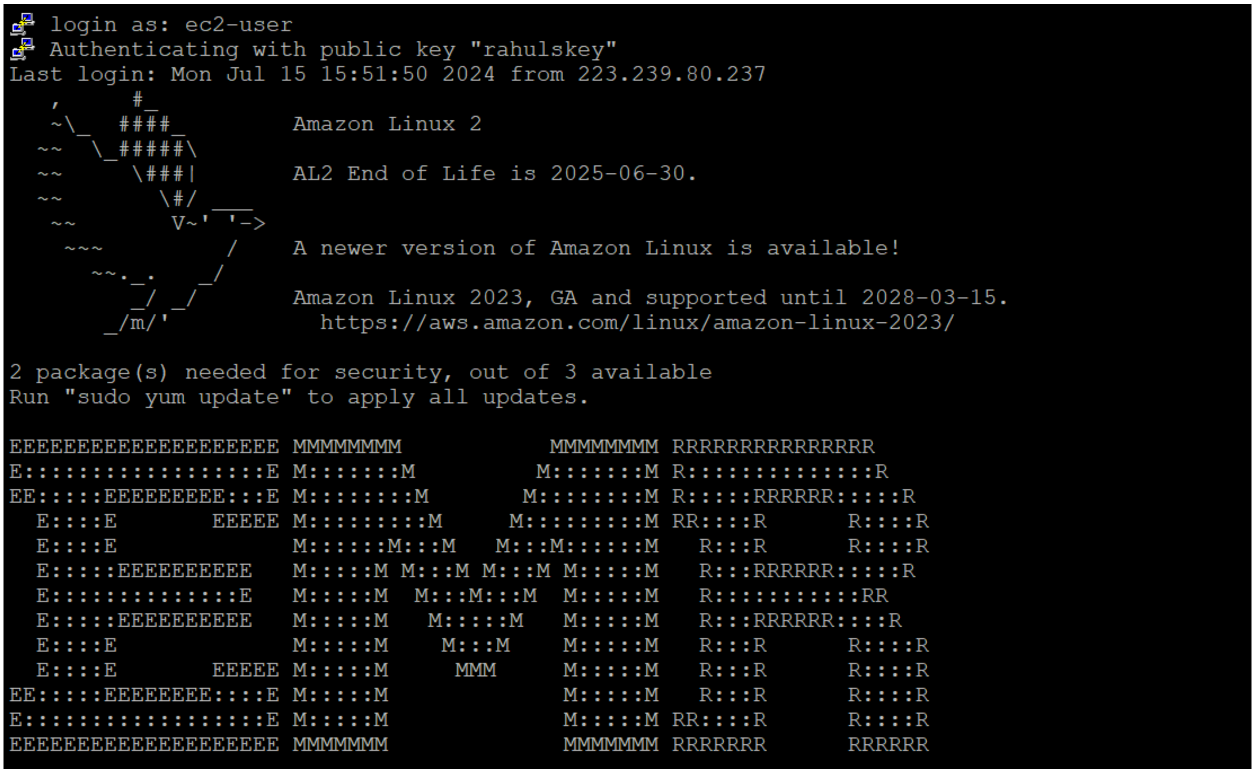
Status and time

Status
Waiting

Creation time
July 16, 2024, 10:33 (UTC+05:30)

Elapsed time
1 hour, 27 minutes

2. **Logged into EMR instance as “ec2-user”**



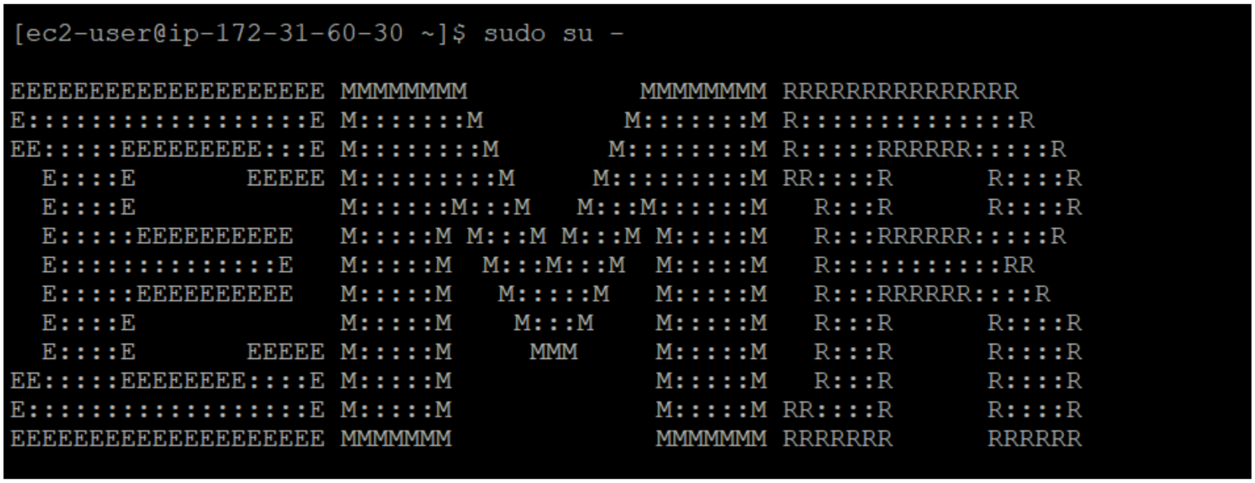
```
login as: ec2-user
Authenticating with public key "rahulskey"
Last login: Mon Jul 15 15:51:50 2024 from 223.239.80.237
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

2 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

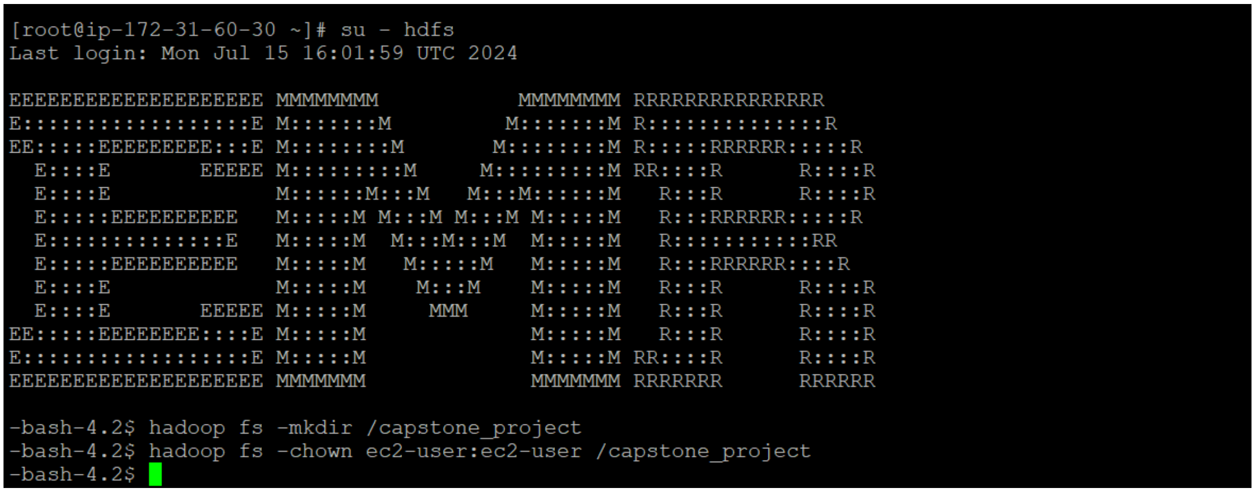
3. **Switch to root user and then to hdfs user. Create directory and change its ownership -> exit from hdfs user -> exit from root user back to ec2-user**.

```
sudo su –
su – hdfs
hadoop fs -mkdir /capstone_project
hadoop fs -chown ec2-user:ec2-user /capstone_project
```



```
[ec2-user@ip-172-31-60-30 ~]$ sudo su -

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

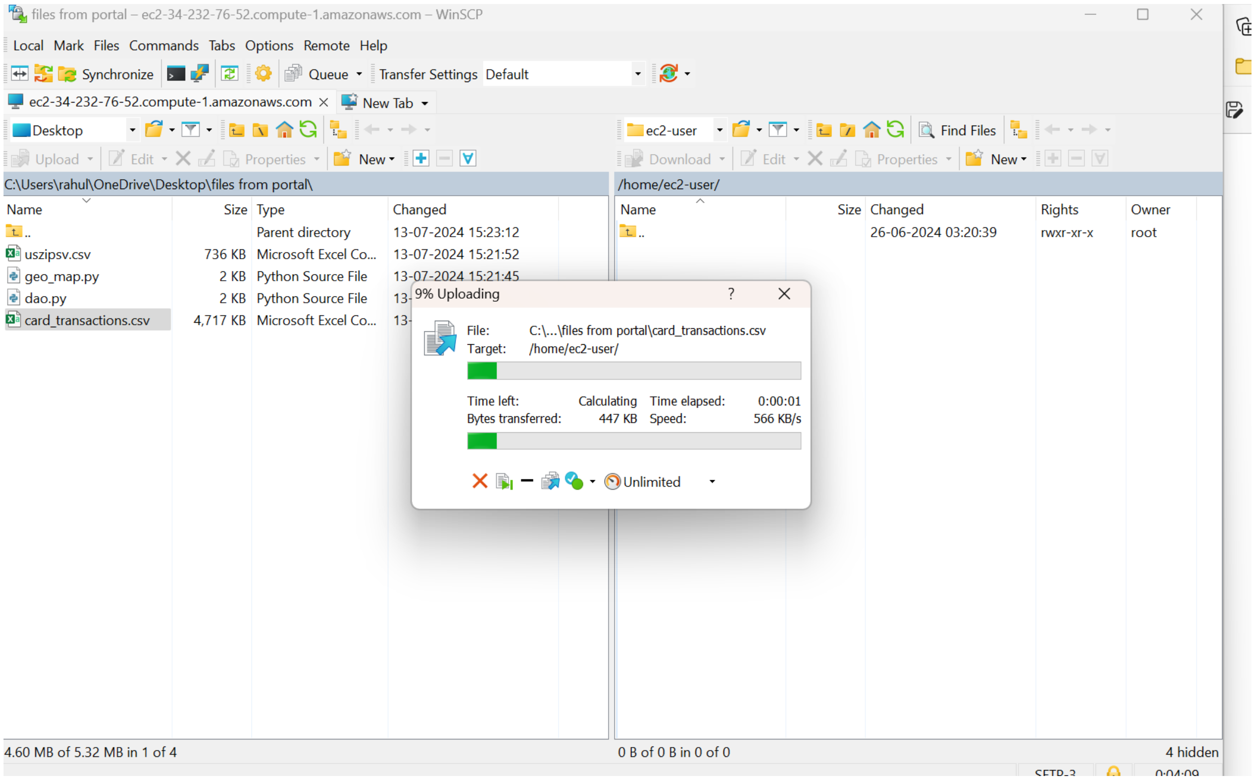


```
[root@ip-172-31-60-30 ~]# su - hdfs
Last login: Mon Jul 15 16:01:59 UTC 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

-bash-4.2$ hadoop fs -mkdir /capstone_project
-bash-4.2$ hadoop fs -chown ec2-user:ec2-user /capstone_project
-bash-4.2$
```

4. Downloaded **card_transactions.csv** from the resource section of the capstone project from the learning platform and transfer it to ec2 instance via WinSCP.



files from portal – ec2-34-232-76-52.compute-1.amazonaws.com – WinSCP

Local Mark Files Commands Tabs Options Remote Help
Synchronize
Queue
Transfer Settings Default
ec2-34-232-76-52.compute-1.amazonaws.com
New Tab
Desktop
Upload
Edit
Properties
New
ec2-user
Find Files
Download
Edit
Properties
New

C:\Users\rahul\OneDrive\Desktop\files from portal\

```
Name                    Size  Type                 Changed
..                            Parent directory     13-07-2024 15:23:12
uszipsv.csv           736 KB  Microsoft Excel Co... 13-07-2024 15:21:52
geo_map.py              2 KB  Python Source File   13-07-2024 15:21:45
dao.py                  2 KB  Python Source File   13-
card_transactions.csv 4,717 KB Microsoft Excel Co... 13-
```

/home/ec2-user/

```
Name    Size  Changed              Rights     Owner
..            26-06-2024 03:20:39  rwxr-xr-x  root
```

9% Uploading

```
File:    C:\...\files from portal\card_transactions.csv
Target:  /home/ec2-user/

Time left:          Calculating   Time elapsed:   0:00:01
Bytes transferred:  447 KB        Speed:          566 KB/s
```

4.60 MB of 5.32 MB in 1 of 4

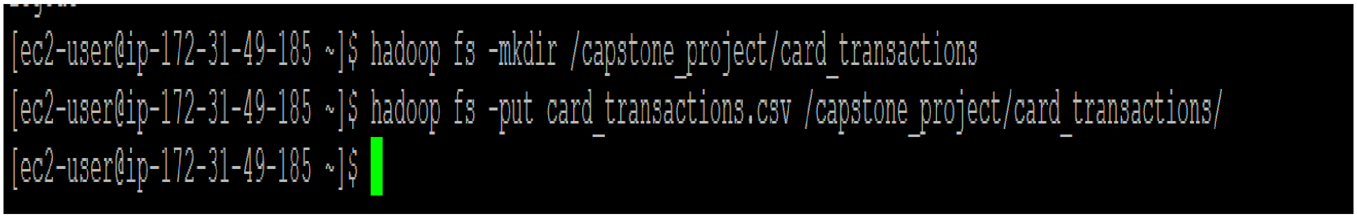
0 B of 0 B in 0 of 0

4 hidden

5. Create a directory in HDFS and copy card_transactions.csv in that location.

hadoop fs -mkdir/capstone_project/card_transactions

hadoop fs -put card_transactions.csv /capstone_project/card_transactions/



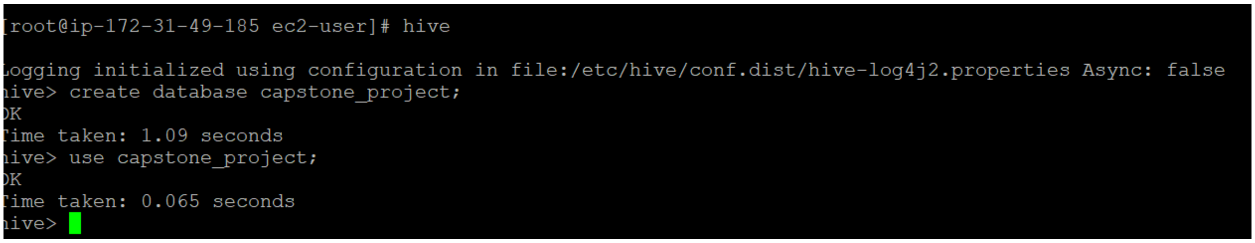
```
[ec2-user@ip-172-31-49-185 ~]$ hadoop fs -mkdir /capstone_project/card_transactions
[ec2-user@ip-172-31-49-185 ~]$ hadoop fs -put card_transactions.csv /capstone_project/card_transactions/
[ec2-user@ip-172-31-49-185 ~]$
```

Now our basic setup is ready for the project. We can now start with completing desired tasks

Task 1: Load the transactions history data (card_transactions.csv) in a NoSQL database.

--------- Hive Operations: Starts Here ----------

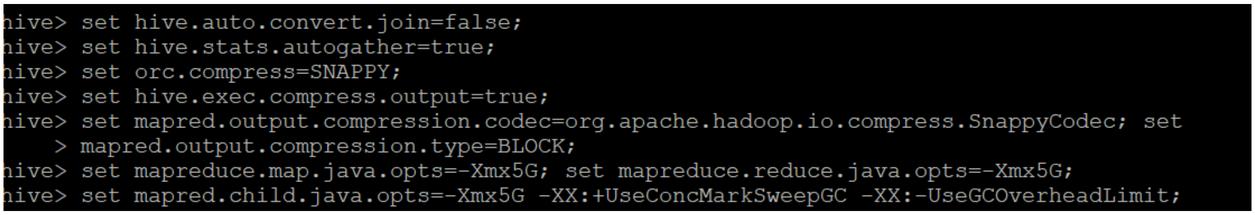
1. Start hive and create new database named ccfd_capstone_project -> switch to ccfd_capstone_project database
   create database capstone_project;
   use capstone_project;



```
[root@ip-172-31-49-185 ec2-user]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> create database capstone_project;
OK
Time taken: 1.09 seconds
hive> use capstone_project;
OK
Time taken: 0.065 seconds
hive>
```

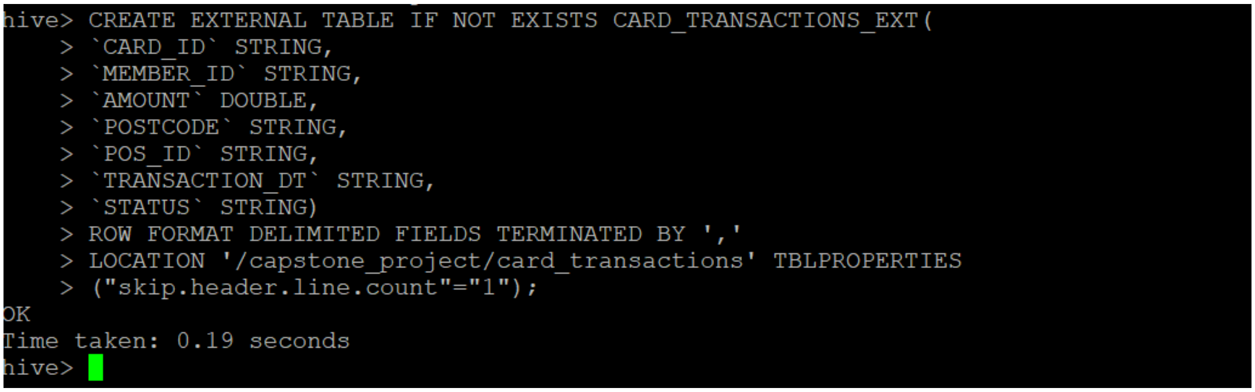
2. **Set below parameters for the hive session**
   set hive.auto.convert.join=false;
   set hive.stats.autogather=true;
   set orc.compress=SNAPPY;
   set hive.exec.compress.output=true;
   set mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
   set mapred.output.compression.type=BLOCK;
   set mapreduce.map.java.opts=-Xmx5G; set mapreduce.reduce.java.opts=-Xmx5G;
   set mapred.child.java.opts=-Xmx5G -XX:+UseConcMarkSweepGC -XX:-UseGCOverheadLimit;



```
hive> set hive.auto.convert.join=false;
hive> set hive.stats.autogather=true;
hive> set orc.compress=SNAPPY;
hive> set hive.exec.compress.output=true;
hive> set mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec; set
    > mapred.output.compression.type=BLOCK;
hive> set mapreduce.map.java.opts=-Xmx5G; set mapreduce.reduce.java.opts=-Xmx5G;
hive> set mapred.child.java.opts=-Xmx5G -XX:+UseConcMarkSweepGC -XX:-UseGCOverheadLimit;
```

3. **Create an external table “card_transactions_ext”**

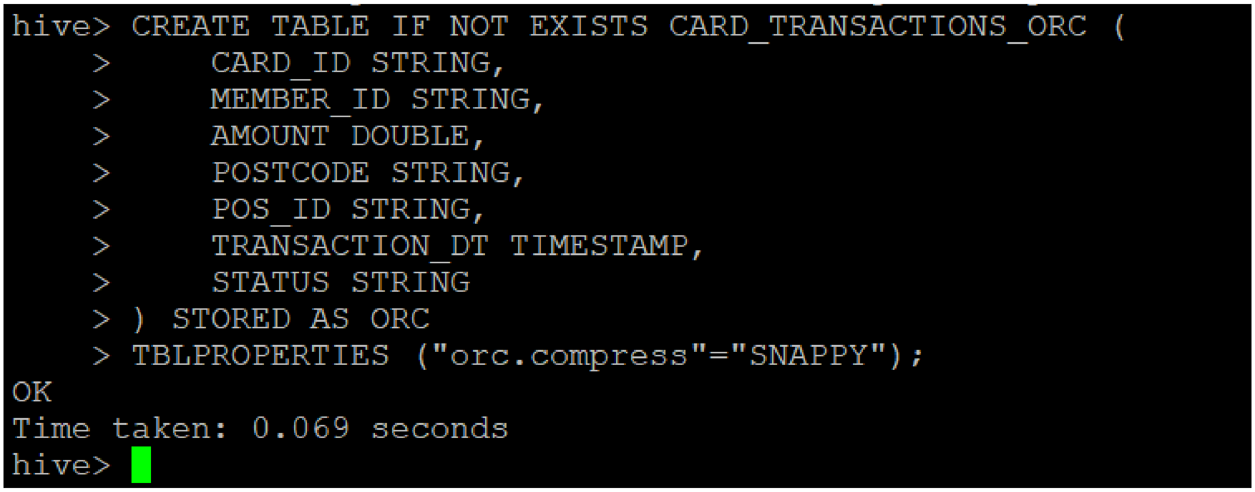
```
CREATE EXTERNAL TABLE IF NOT EXISTS CARD_TRANSACTIONS_EXT(
 `CARD_ID` STRING,
`MEMBER_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`POS_ID` STRING,
`TRANSACTION_DT` STRING,
`STATUS` STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/capstone_project/card_transactions' TBLPROPERTIES
("skip.header.line.count"="1");
```



```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS CARD_TRANSACTIONS_EXT(
    > `CARD_ID` STRING,
    > `MEMBER_ID` STRING,
    > `AMOUNT` DOUBLE,
    > `POSTCODE` STRING,
    > `POS_ID` STRING,
    > `TRANSACTION_DT` STRING,
    > `STATUS` STRING)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LOCATION '/capstone_project/card_transactions' TBLPROPERTIES
    > ("skip.header.line.count"="1");
OK
Time taken: 0.19 seconds
hive>
```

4. Create table **"card_transactions_orc"** in ORC format for better performance.

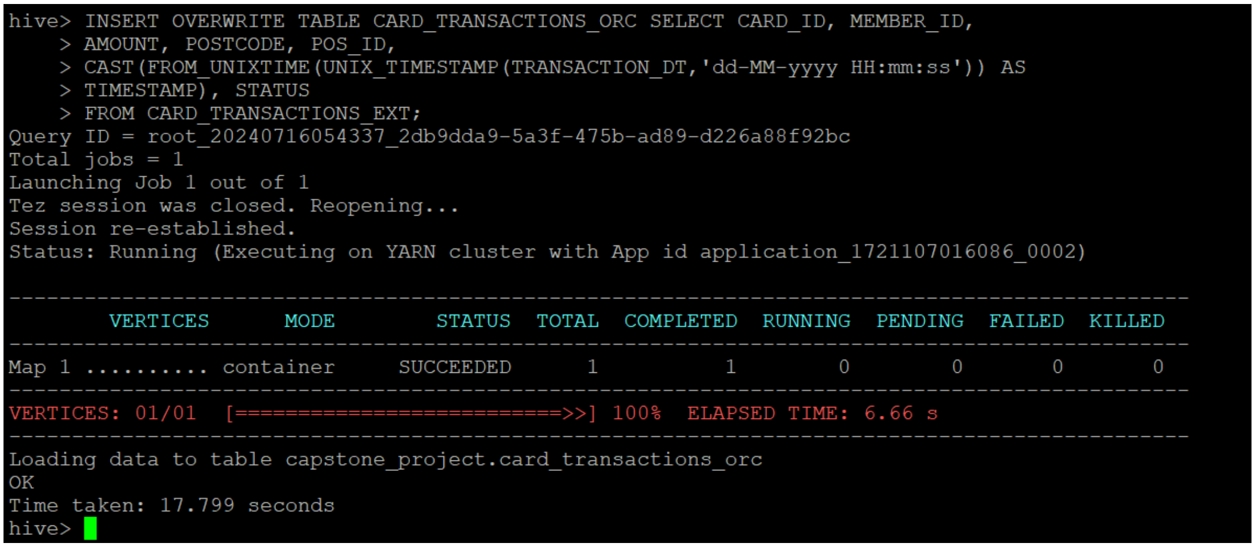
```
CREATE TABLE IF NOT EXISTS CARD_TRANSACTIONS_ORC(`CARD_ID`
STRING,`MEMBER_ID` STRING,`AMOUNT` DOUBLE,`POSTCODE` STRING,`POS_ID`
STRING,`TRANSACTION_DT` TIMESTAMP,`STATUS` STRING) STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");
```



```
hive> CREATE TABLE IF NOT EXISTS CARD_TRANSACTIONS_ORC (
    >     CARD_ID STRING,
    >     MEMBER_ID STRING,
    >     AMOUNT DOUBLE,
    >     POSTCODE STRING,
    >     POS_ID STRING,
    >     TRANSACTION_DT TIMESTAMP,
    >     STATUS STRING
    > ) STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.069 seconds
hive>
```

5. Load data in **“card_transactions_orc”** table and type cast **transaction_dt** column in timestamp format

INSERT OVERWRITE TABLE CARD_TRANSACTIONS_ORC SELECT CARD_ID, MEMBER_ID, AMOUNT, POSTCODE, POS_ID, CAST(FROM_UNIXTIME(UNIX_TIMESTAMP(TRANSACTION_DT,'dd-MM-yyyy HH:mm:ss')) AS TIMESTAMP), STATUS FROM CARD_TRANSACTIONS_EXT;

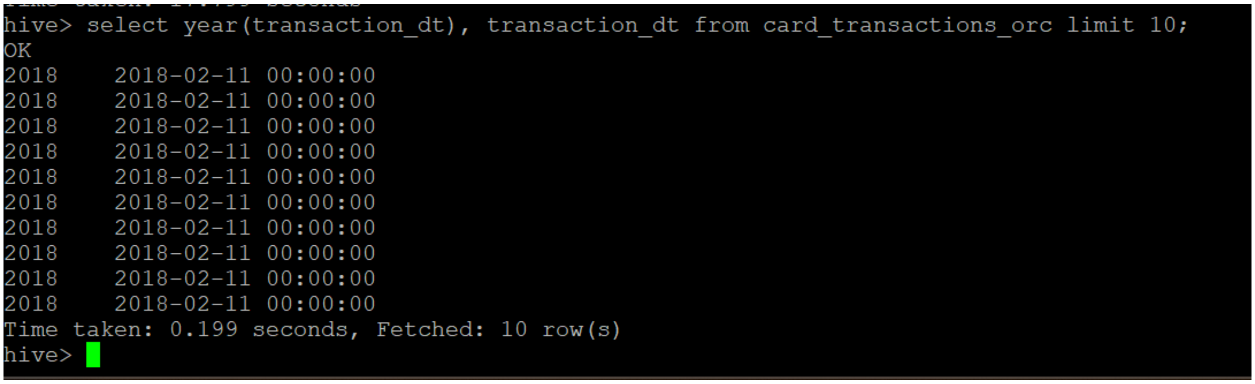


```
hive> INSERT OVERWRITE TABLE CARD_TRANSACTIONS_ORC SELECT CARD_ID, MEMBER_ID,
    > AMOUNT, POSTCODE, POS_ID,
    > CAST(FROM_UNIXTIME(UNIX_TIMESTAMP(TRANSACTION_DT,'dd-MM-yyyy HH:mm:ss')) AS
    > TIMESTAMP), STATUS
    > FROM CARD_TRANSACTIONS_EXT;
Query ID = root_20240716054337_2db9dda9-5a3f-475b-ad89-d226a88f92bc
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1721107016086_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 6.66 s
----------------------------------------------------------------------------------------------
Loading data to table capstone_project.card_transactions_orc
OK
Time taken: 17.799 seconds
hive>
```

6. Verify **transaction_dt** and year columns in **"card_transactions_orc"** table.

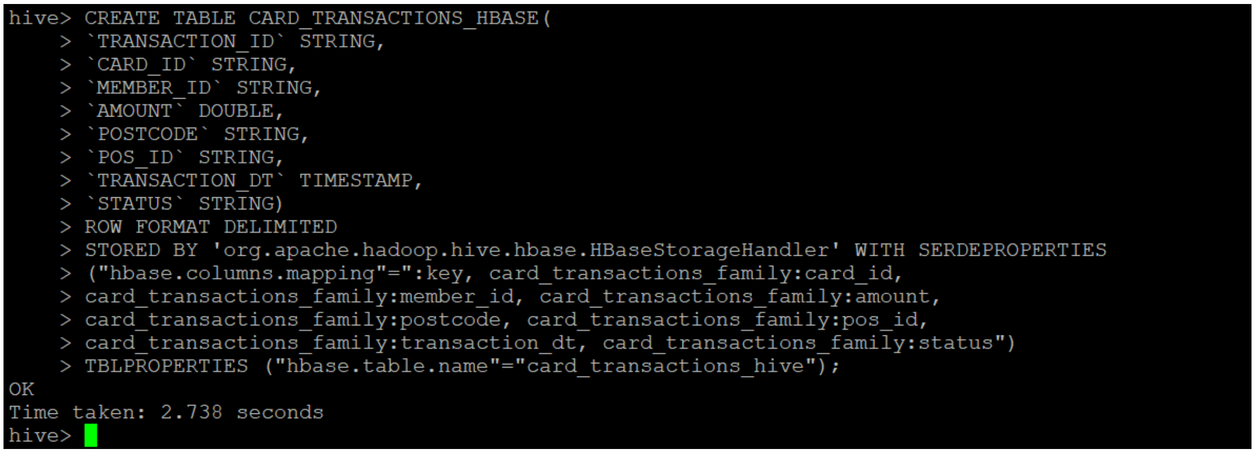
select year(transaction_dt), transaction_dt from card_transactions_orc limit 10;



```
hive> select year(transaction_dt), transaction_dt from card_transactions_orc limit 10;
OK
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
2018    2018-02-11 00:00:00
Time taken: 0.199 seconds, Fetched: 10 row(s)
hive>
```

7. Create hive-hbase integrated table which will be visible in HBase as well. "**card_transactions_hbase**" table

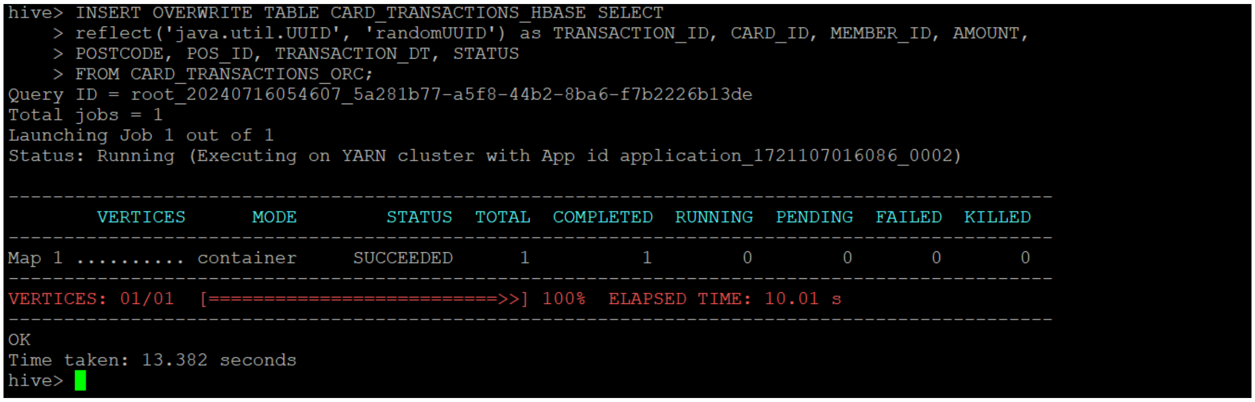
CREATE TABLE CARD_TRANSACTIONS_HBASE(`TRANSACTION_ID` STRING, `CARD_ID` STRING, `MEMBER_ID` STRING, `AMOUNT` DOUBLE, `POSTCODE` STRING, `POS_ID` STRING, `TRANSACTION_DT` TIMESTAMP, `STATUS` STRING)
ROW FORMAT DELIMITED
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping"=":key, card_transactions_family:card_id, card_transactions_family:member_id, card_transactions_family:amount, card_transactions_family:postcode, card_transactions_family:pos_id, card_transactions_family:transaction_dt, card_transactions_family:status") TBLPROPERTIES ("hbase.table.name"="card_transactions_hive");



```
hive> CREATE TABLE CARD_TRANSACTIONS_HBASE(
    > `TRANSACTION_ID` STRING,
    > `CARD_ID` STRING,
    > `MEMBER_ID` STRING,
    > `AMOUNT` DOUBLE,
    > `POSTCODE` STRING,
    > `POS_ID` STRING,
    > `TRANSACTION_DT` TIMESTAMP,
    > `STATUS` STRING)
    > ROW FORMAT DELIMITED
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES
    > ("hbase.columns.mapping"=":key, card_transactions_family:card_id,
    > card_transactions_family:member_id, card_transactions_family:amount,
    > card_transactions_family:postcode, card_transactions_family:pos_id,
    > card_transactions_family:transaction_dt, card_transactions_family:status")
    > TBLPROPERTIES ("hbase.table.name"="card_transactions_hive");
OK
Time taken: 2.738 seconds
hive>
```

8. Load data in "card_transactions_hbase" table which will be visible in HBase as well with table name as "card_transactions_hive".Using randomUUID to populate TRANSACTION_ID field (row key).

INSERT OVERWRITE TABLE CARD_TRANSACTIONS_HBASE SELECT reflect('java.util.UUID', 'randomUUID') as TRANSACTION_ID, CARD_ID, MEMBER_ID, AMOUNT, POSTCODE, POS_ID, TRANSACTION_DT, STATUS FROM CARD_TRANSACTIONS_ORC;

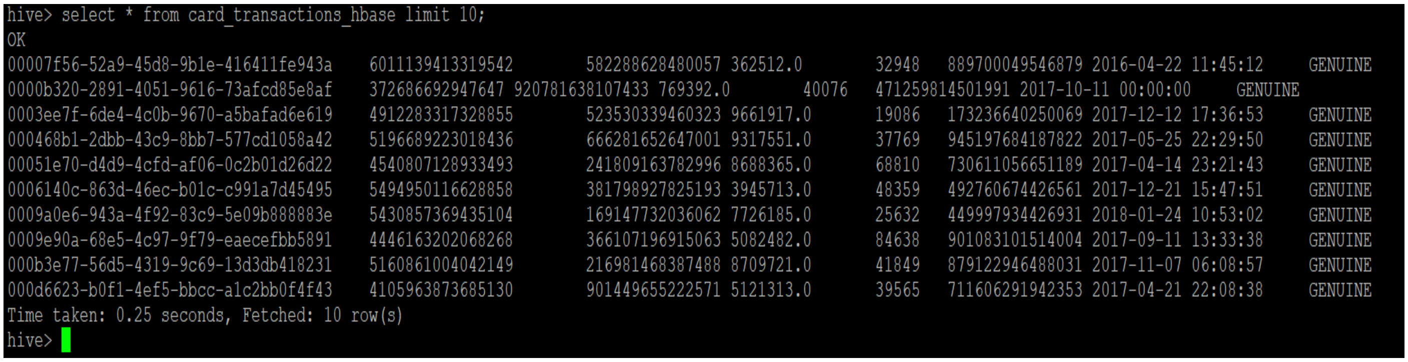


```
hive> INSERT OVERWRITE TABLE CARD_TRANSACTIONS_HBASE SELECT
    > reflect('java.util.UUID', 'randomUUID') as TRANSACTION_ID, CARD_ID, MEMBER_ID, AMOUNT,
    > POSTCODE, POS_ID, TRANSACTION_DT, STATUS
    > FROM CARD_TRANSACTIONS_ORC;
Query ID = root_20240716054607_5a281b77-a5f8-44b2-8ba6-f7b2226b13de
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1721107016086_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 10.01 s
----------------------------------------------------------------------------------------------
OK
Time taken: 13.382 seconds
hive>
```

9. Verify data in **"card_transactions_hbase**" table.

select * from card_transactions_hbase limit 10;



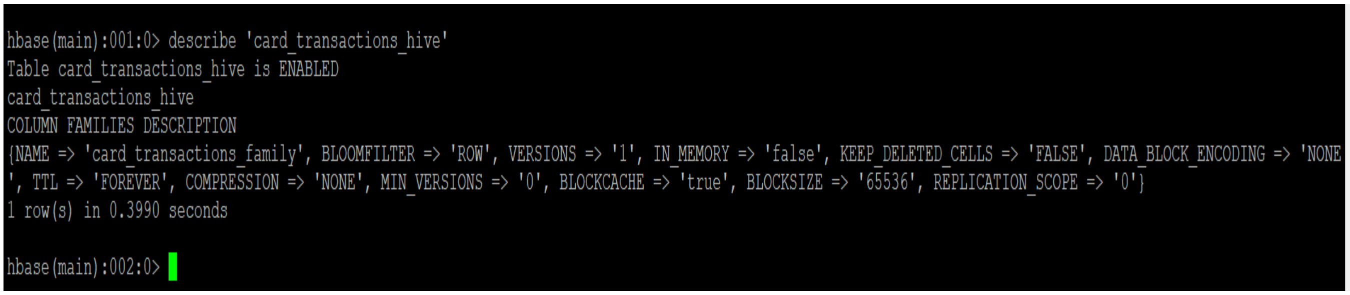
```
hive> select * from card_transactions_hbase limit 10;
OK
00007f56-52a9-45d8-9b1e-416411fe943a    6011139413319542        582288628480057 362512.0        32948   889700049546879 2016-04-22 11:45:12     GENUINE
0000b320-2891-4051-9616-73afcd85e8af    372686692947647 920781638107433 769392.0        40076   471259814501991 2017-10-11 00:00:00     GENUINE
0003ee7f-6de4-4c0b-9670-a5bafad6e619    4912283317328855        523530339460323 9661917.0       19086   173236640250069 2017-12-12 17:36:53     GENUINE
000468b1-2dbb-43c9-8bb7-577cd1058a42    5196689223018436        666281652647001 9317551.0       37769   945197684187822 2017-05-25 22:29:50     GENUINE
00051e70-d4d9-4cfd-af06-0c2b01d26d22    4540807128933493        241809163782996 8688365.0       68810   730611056651189 2017-04-14 23:21:43     GENUINE
0006140c-863d-46ec-b01c-c991a7d45495    5494950116628858        381798927825193 3945713.0       48359   492760674426561 2017-12-21 15:47:51     GENUINE
0009a0e6-943a-4f92-83c9-5e09b888883e    5430857369435104        169147732036062 7726185.0       25632   449997934426931 2018-01-24 10:53:02     GENUINE
0009e90a-68e5-4c97-9f79-eaecefbb5891    4446163202068268        366107196915063 5082482.0       84638   901083101514004 2017-09-11 13:33:38     GENUINE
000b3e77-56d5-4319-9c69-13d3db418231    5160861004042149        216981468387488 8709721.0       41849   879122946488031 2017-11-07 06:08:57     GENUINE
000d6623-b0f1-4ef5-bbcc-a1c2bb0f4f43    4105963873685130        901449655222571 5121313.0       39565   711606291942353 2017-04-21 22:08:38     GENUINE
Time taken: 0.25 seconds, Fetched: 10 row(s)
hive>
```

---------- **Hive Operations: Ends Here** ----------

---------- Hbase Operations: Starts Here ----------

1. Start HBase and verify details of "**card_transactions_hive**" table (hive-hbase integrated table).

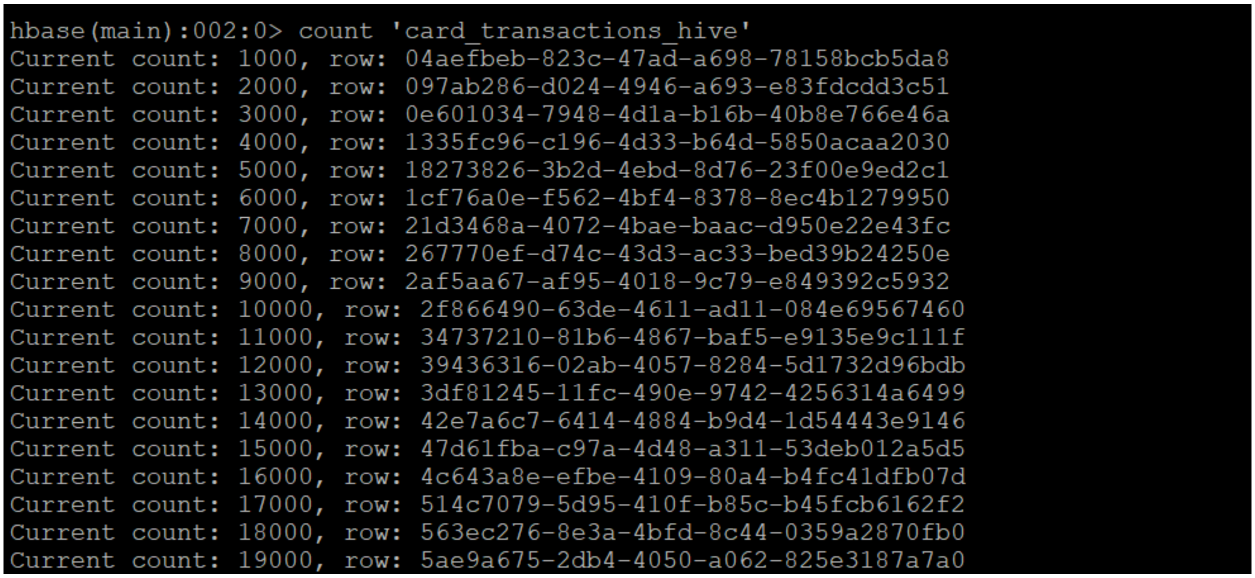
   describe 'card_transactions_hive'



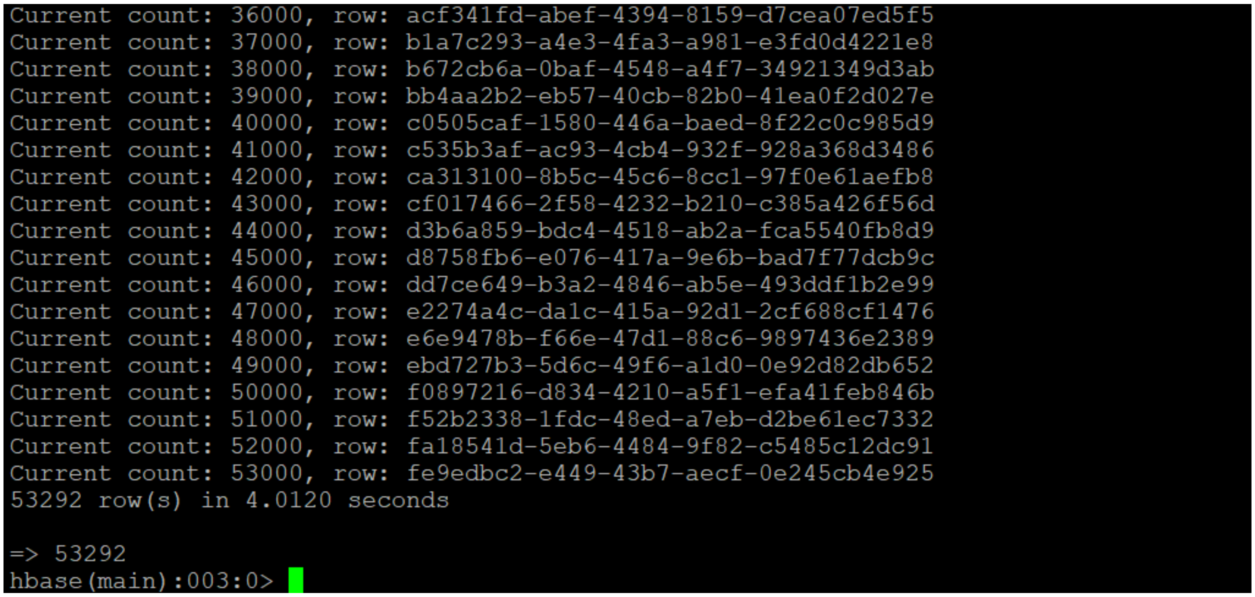
```
hbase(main):001:0> describe 'card_transactions_hive'
Table card_transactions_hive is ENABLED
card_transactions_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'card_transactions_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE
', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.3990 seconds

hbase(main):002:0>
```

2. Verify count of **"card_transactions_hive"** table
   Command : count 'card_transactions_hive'



```
hbase(main):002:0> count 'card_transactions_hive'
Current count: 1000, row: 04aefbeb-823c-47ad-a698-78158bcb5da8
Current count: 2000, row: 097ab286-d024-4946-a693-e83fdcdd3c51
Current count: 3000, row: 0e601034-7948-4d1a-b16b-40b8e766e46a
Current count: 4000, row: 1335fc96-c196-4d33-b64d-5850acaa2030
Current count: 5000, row: 18273826-3b2d-4ebd-8d76-23f00e9ed2c1
Current count: 6000, row: 1cf76a0e-f562-4bf4-8378-8ec4b1279950
Current count: 7000, row: 21d3468a-4072-4bae-baac-d950e22e43fc
Current count: 8000, row: 267770ef-d74c-43d3-ac33-bed39b24250e
Current count: 9000, row: 2af5aa67-af95-4018-9c79-e849392c5932
Current count: 10000, row: 2f866490-63de-4611-ad11-084e69567460
Current count: 11000, row: 34737210-81b6-4867-baf5-e9135e9c111f
Current count: 12000, row: 39436316-02ab-4057-8284-5d1732d96bdb
Current count: 13000, row: 3df81245-11fc-490e-9742-4256314a6499
Current count: 14000, row: 42e7a6c7-6414-4884-b9d4-1d54443e9146
Current count: 15000, row: 47d61fba-c97a-4d48-a311-53deb012a5d5
Current count: 16000, row: 4c643a8e-efbe-4109-80a4-b4fc41dfb07d
Current count: 17000, row: 514c7079-5d95-410f-b85c-b45fcb6162f2
Current count: 18000, row: 563ec276-8e3a-4bfd-8c44-0359a2870fb0
Current count: 19000, row: 5ae9a675-2db4-4050-a062-825e3187a7a0
```



```
Current count: 36000, row: acf341fd-abef-4394-8159-d7cea07ed5f5
Current count: 37000, row: b1a7c293-a4e3-4fa3-a981-e3fd0d4221e8
Current count: 38000, row: b672cb6a-0baf-4548-a4f7-34921349d3ab
Current count: 39000, row: bb4aa2b2-eb57-40cb-82b0-41ea0f2d027e
Current count: 40000, row: c0505caf-1580-446a-baed-8f22c0c985d9
Current count: 41000, row: c535b3af-ac93-4cb4-932f-928a368d3486
Current count: 42000, row: ca313100-8b5c-45c6-8cc1-97f0e61aefb8
Current count: 43000, row: cf017466-2f58-4232-b210-c385a426f56d
Current count: 44000, row: d3b6a859-bdc4-4518-ab2a-fca5540fb8d9
Current count: 45000, row: d8758fb6-e076-417a-9e6b-bad7f77dcb9c
Current count: 46000, row: dd7ce649-b3a2-4846-ab5e-493ddf1b2e99
Current count: 47000, row: e2274a4c-da1c-415a-92d1-2cf688cf1476
Current count: 48000, row: e6e9478b-f66e-47d1-88c6-9897436e2389
Current count: 49000, row: ebd727b3-5d6c-49f6-a1d0-0e92d82db652
Current count: 50000, row: f0897216-d834-4210-a5f1-efa41feb846b
Current count: 51000, row: f52b2338-1fdc-48ed-a7eb-d2be61ec7332
Current count: 52000, row: fa18541d-5eb6-4484-9f82-c5485c12dc91
Current count: 53000, row: fe9edbc2-e449-43b7-aecf-0e245cb4e925
53292 row(s) in 4.0120 seconds

=> 53292
hbase(main):003:0>
```

---------- **Hbase Operations: Ends Here** ----------

Count of the **"card_transactions_hive**" table is **53292** which is matching with given requirement

Task 2: Ingest the relevant data from AWS RDS to Hadoop.

---------- **Sqoop Operations: Starts Here**----------

1. Run Sqoop command to import "member_score" table from RDS to HDFS.

```
sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data \
--username upgraduser \
--password upgraduser \
--table member_score \
--null-string 'NA' \
--null-non-string '\\N' \
--delete-target-dir \
--target-dir '/capstone_project/member_score' \
-m 1
```

2. Run Sqoop command to import “card_member” table from RDS to HDFS.

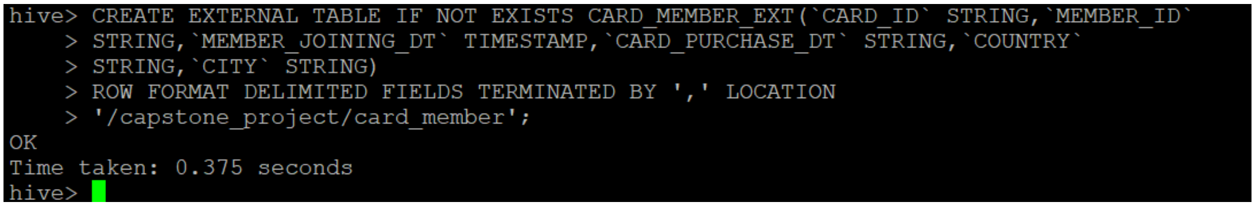
```
sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data \
--username upgraduser \
--password upgraduser \
--table card_member \
--null-string 'NA' \
--null-non-string '\\N' \
--delete-target-dir \
--target-dir '/capstone_project/card_member' \
-m 1
```

---------- Sqoop Operations: Ends Here----------

---------- Hive Operations: Starts Here----------

1. Start hive and Create external table **"card_member_ext"** to hold data from card_member table in RDS.

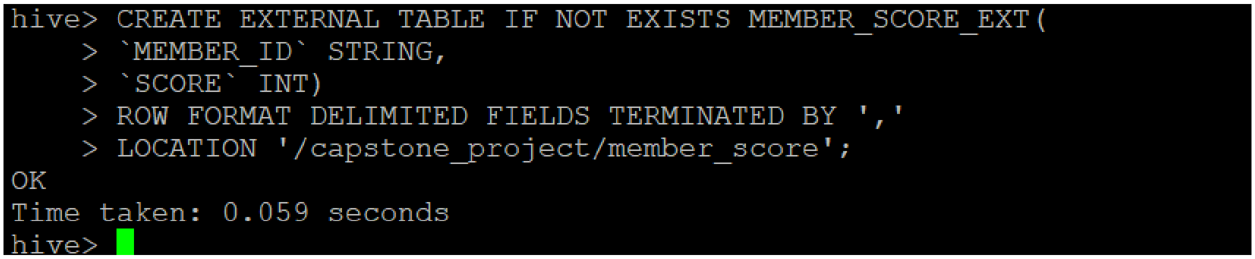
```
CREATE EXTERNAL TABLE IF NOT EXISTS CARD_MEMBER_EXT(`CARD_ID`
STRING,`MEMBER_ID`
STRING,`MEMBER_JOINING_DT` TIMESTAMP,`CARD_PURCHASE_DT`
STRING,`COUNTRY`
STRING,`CITY` STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION
'/capstone_project/card_member';
```



```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS CARD_MEMBER_EXT(`CARD_ID` STRING,`MEMBER_ID`
    > STRING,`MEMBER_JOINING_DT` TIMESTAMP,`CARD_PURCHASE_DT` STRING,`COUNTRY`
    > STRING,`CITY` STRING)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION
    > '/capstone_project/card_member';
OK
Time taken: 0.375 seconds
hive>
```

2. Create external table **"member_score_ext**" to hold data from member_score table in RDS.

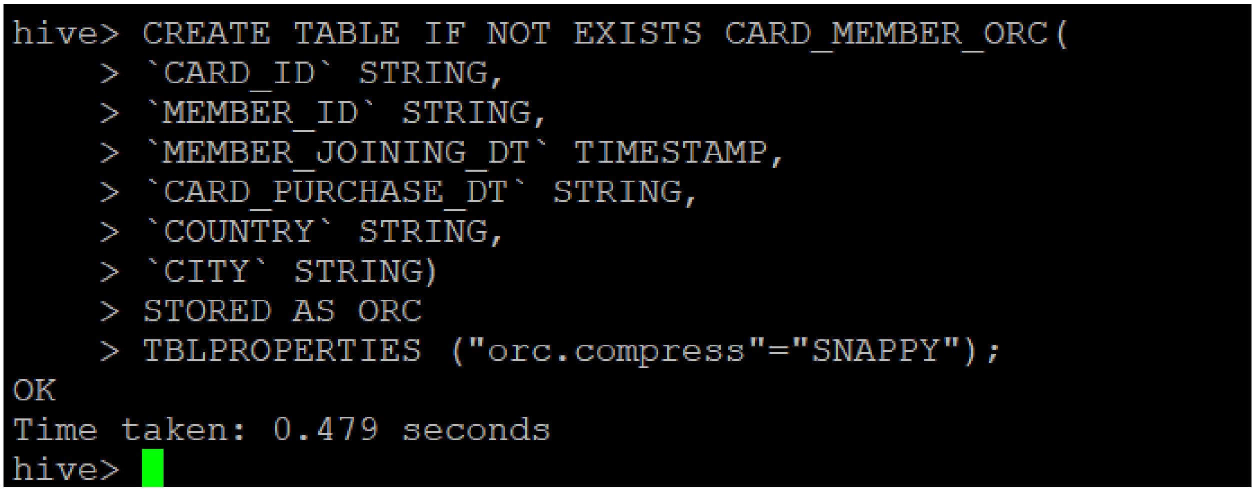
CREATE EXTERNAL TABLE IF NOT EXISTS MEMBER_SCORE_EXT(
`MEMBER_ID` STRING,
`SCORE` INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/capstone_project/member_score';



```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS MEMBER_SCORE_EXT(
    > `MEMBER_ID` STRING,
    > `SCORE` INT)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LOCATION '/capstone_project/member_score';
OK
Time taken: 0.059 seconds
hive>
```

3. Create "**card_member_orc**" table. For better performance.

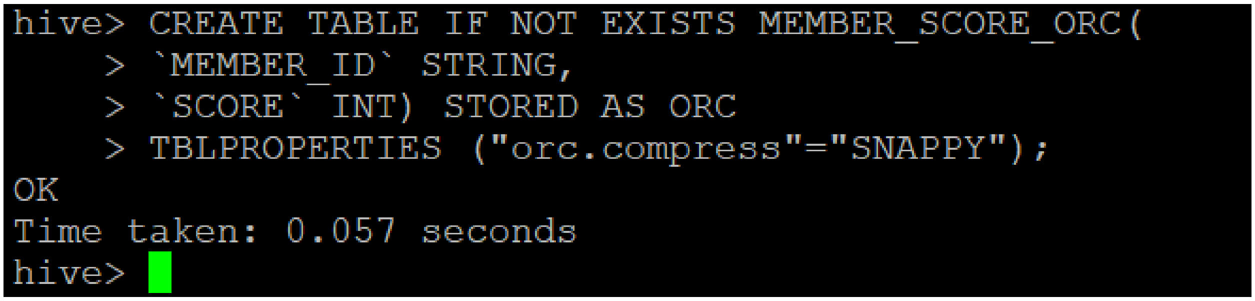
CREATE TABLE IF NOT EXISTS CARD_MEMBER_ORC(
`CARD_ID` STRING,
`MEMBER_ID` STRING,
`MEMBER_JOINING_DT` TIMESTAMP,
`CARD_PURCHASE_DT` STRING,
`COUNTRY` STRING,
`CITY` STRING)
STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");



```
hive> CREATE TABLE IF NOT EXISTS CARD_MEMBER_ORC(
    > `CARD_ID` STRING,
    > `MEMBER_ID` STRING,
    > `MEMBER_JOINING_DT` TIMESTAMP,
    > `CARD_PURCHASE_DT` STRING,
    > `COUNTRY` STRING,
    > `CITY` STRING)
    > STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.479 seconds
hive>
```

4. Create "**member_score_orc**" table. For better performance.

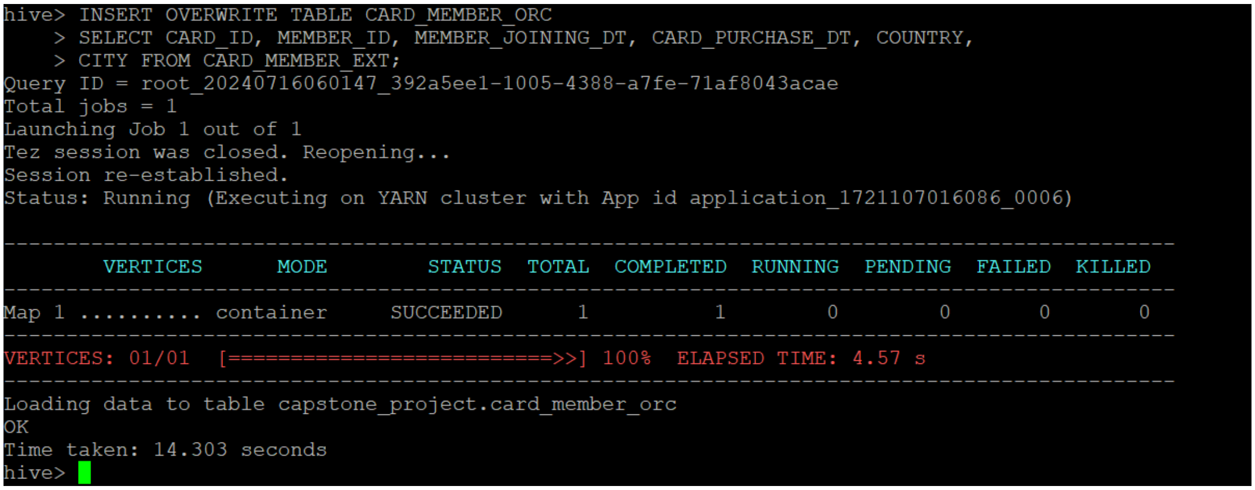
CREATE TABLE IF NOT EXISTS MEMBER_SCORE_ORC(
`MEMBER_ID` STRING,
`SCORE` INT) STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");



```
hive> CREATE TABLE IF NOT EXISTS MEMBER_SCORE_ORC(
    > `MEMBER_ID` STRING,
    > `SCORE` INT) STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.057 seconds
hive>
```

5. Load data into "**card_member_orc**" table from "**card_member_ext**" table.

INSERT OVERWRITE TABLE CARD_MEMBER_ORC
SELECT CARD_ID, MEMBER_ID, MEMBER_JOINING_DT, CARD_PURCHASE_DT,
COUNTRY,
CITY FROM CARD_MEMBER_EXT;

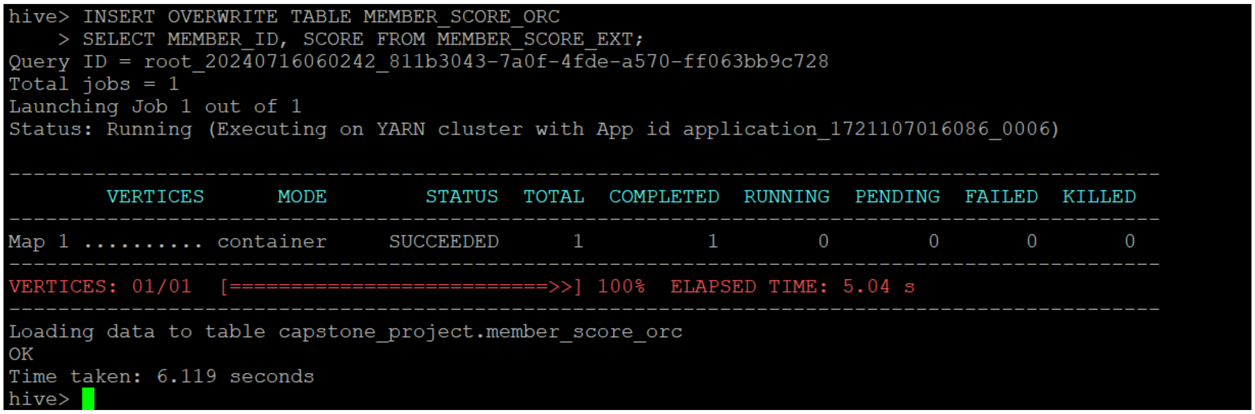


```
hive> INSERT OVERWRITE TABLE CARD_MEMBER_ORC
    > SELECT CARD_ID, MEMBER_ID, MEMBER_JOINING_DT, CARD_PURCHASE_DT, COUNTRY,
    > CITY FROM CARD_MEMBER_EXT;
Query ID = root_20240716060147_392a5ee1-1005-4388-a7fe-71af8043acae
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1721107016086_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 4.57 s
----------------------------------------------------------------------------------------------
Loading data to table capstone_project.card_member_orc
OK
Time taken: 14.303 seconds
hive>
```

6. Load data into “**member_score_orc**” table from “**member_score_ext**” table.

INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
SELECT MEMBER_ID, SCORE FROM MEMBER_SCORE_EXT;

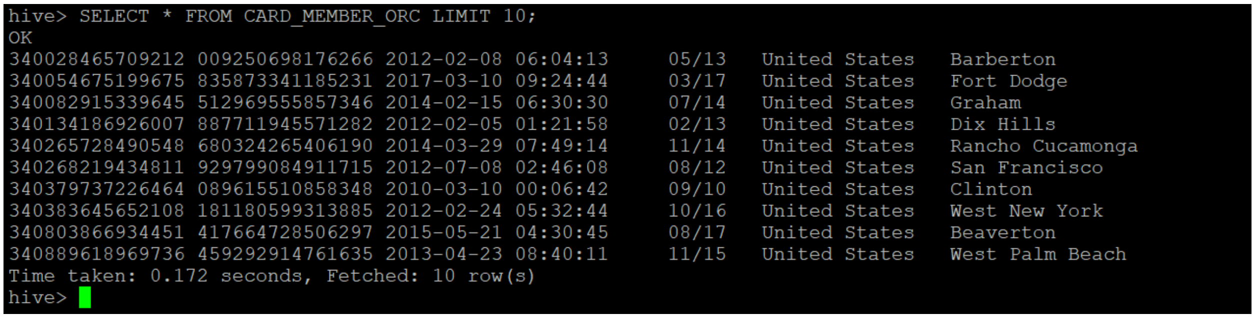


```
hive> INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
    > SELECT MEMBER_ID, SCORE FROM MEMBER_SCORE_EXT;
Query ID = root_20240716060242_811b3043-7a0f-4fde-a570-ff063bb9c728
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1721107016086_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 5.04 s
----------------------------------------------------------------------------------------------
Loading data to table capstone_project.member_score_orc
OK
Time taken: 6.119 seconds
hive>
```

7. Verify data in “**card_member_orc**" table.

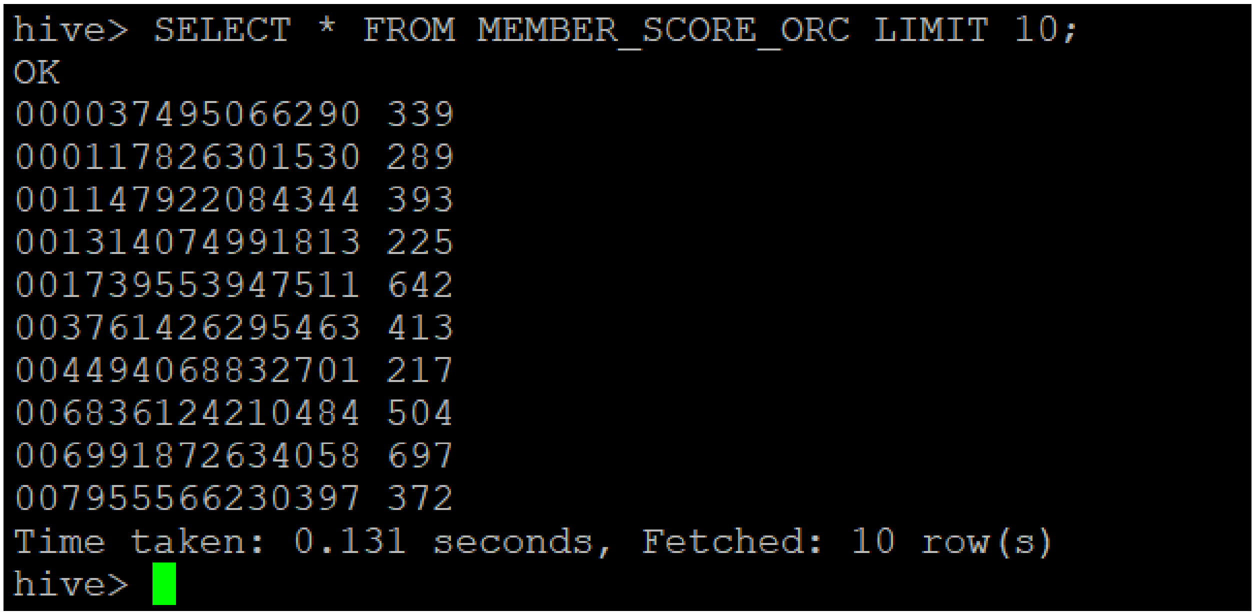
SELECT * FROM CARD_MEMBER_ORC LIMIT 10;



```
hive> SELECT * FROM CARD_MEMBER_ORC LIMIT 10;
OK
340028465709212 009250698176266 2012-02-08 06:04:13     05/13   United States   Barberton
340054675199675 835873341185231 2017-03-10 09:24:44     03/17   United States   Fort Dodge
340082915339645 512969555857346 2014-02-15 06:30:30     07/14   United States   Graham
340134186926007 887711945571282 2012-02-05 01:21:58     02/13   United States   Dix Hills
340265728490548 680324265406190 2014-03-29 07:49:14     11/14   United States   Rancho Cucamonga
340268219434811 929799084911715 2012-07-08 02:46:08     08/12   United States   San Francisco
340379737226464 089615510858348 2010-03-10 00:06:42     09/10   United States   Clinton
340383645652108 181180599313885 2012-02-24 05:32:44     10/16   United States   West New York
340803866934451 417664728506297 2015-05-21 04:30:45     08/17   United States   Beaverton
340889618969736 459292914761635 2013-04-23 08:40:11     11/15   United States   West Palm Beach
Time taken: 0.172 seconds, Fetched: 10 row(s)
hive>
```

8. Verify data in "**member_score_orc**" table.

SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;



```
hive> SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;
OK
000037495066290 339
000117826301530 289
001147922084344 393
001314074991813 225
001739553947511 642
003761426295463 413
004494068832701 217
006836124210484 504
006991872634058 697
007955566230397 372
Time taken: 0.131 seconds, Fetched: 10 row(s)
hive>
```

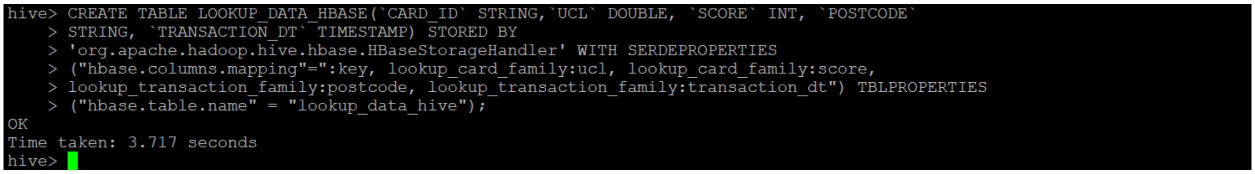
---------- Hive Operations: Ends Here----------

Task 3: Create a look-up table with columns specified earlier in the problem statement.

Create “lookup_data_hbase” table (hive-hbase integrated table) which will be visible in HBase (lookup_data_hive).

---------- **Hive Operations: Starts Here**----------

CREATE TABLE LOOKUP_DATA_HBASE(`CARD_ID` STRING,`UCL` DOUBLE, `SCORE` INT, `POSTCODE`
STRING, `TRANSACTION_DT` TIMESTAMP) STORED BY
'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES
("hbase.columns.mapping"=":key, lookup_card_family:ucl, lookup_card_family:score,
lookup_transaction_family:postcode, lookup_transaction_family:transaction_dt")
TBLPROPERTIES
("hbase.table.name" = "lookup_data_hive");



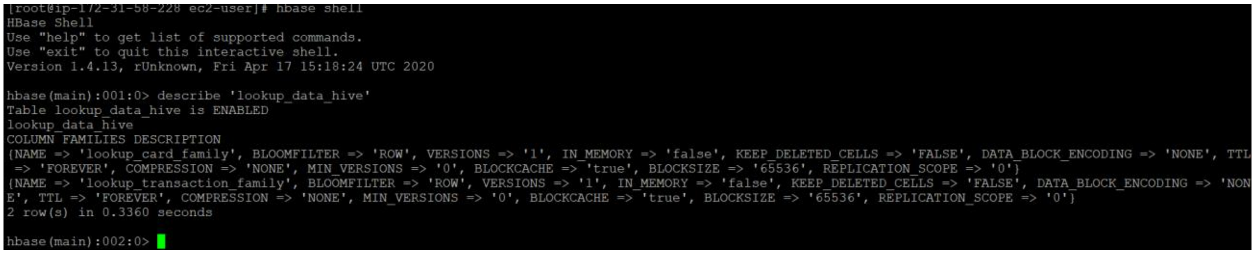
```
hive> CREATE TABLE LOOKUP_DATA_HBASE(`CARD_ID` STRING,`UCL` DOUBLE, `SCORE` INT, `POSTCODE`
    > STRING, `TRANSACTION_DT` TIMESTAMP) STORED BY
    > 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES
    > ("hbase.columns.mapping"=":key, lookup_card_family:ucl, lookup_card_family:score,
    > lookup_transaction_family:postcode, lookup_transaction_family:transaction_dt") TBLPROPERTIES
    > ("hbase.table.name" = "lookup_data_hive");
OK
Time taken: 3.717 seconds
hive>
```

---------- **Hive Operations: Ends Here**----------

---------- **Hbase Operations: Starts Here**----------

- Verify details of **lookup_data_hive** (hive-hbase integrated) table :

describe 'lookup_data_hive'



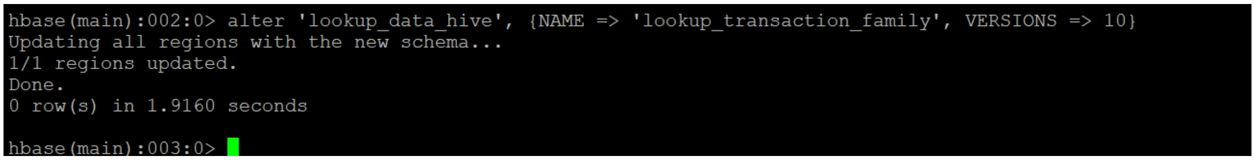
```
[root@ip-172-31-58-228 ec2-user]# hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.13, rUnknown, Fri Apr 17 15:18:24 UTC 2020

hbase(main):001:0> describe 'lookup_data_hive'
Table lookup_data_hive is ENABLED
lookup_data_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'lookup_card_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL
 => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'lookup_transaction_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NON
E', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.3360 seconds

hbase(main):002:0>
```

- Alter "**lookup_data_hive**" table and set VERSIONS to 10 for lookup_transaction_family. We are supposed to store last 10 transactions in lookup table so altering VERSIONS to 10.

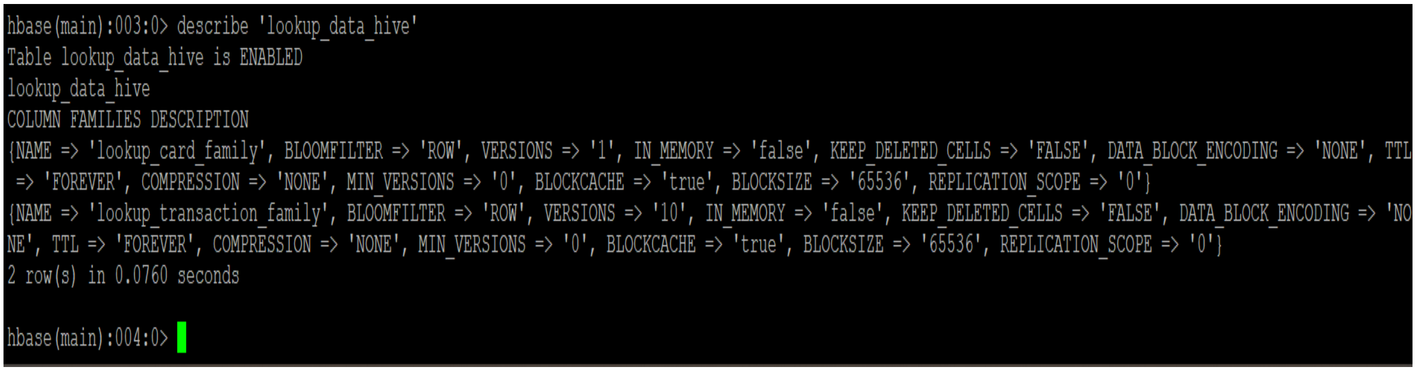
  alter 'lookup_data_hive', {NAME => 'lookup_transaction_family', VERSIONS => 10}



```
hbase(main):002:0> alter 'lookup_data_hive', {NAME => 'lookup_transaction_family', VERSIONS => 10}
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9160 seconds

hbase(main):003:0>
```

- Verify details of "lookup_data_hive" (hive-hbase integrated) table after altering version to 10 : describe 'lookup_data_hive'



```
hbase(main):003:0> describe 'lookup_data_hive'
Table lookup_data_hive is ENABLED
lookup_data_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'lookup_card_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL
 => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'lookup_transaction_family', BLOOMFILTER => 'ROW', VERSIONS => '10', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NO
NE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0760 seconds

hbase(main):004:0>
```

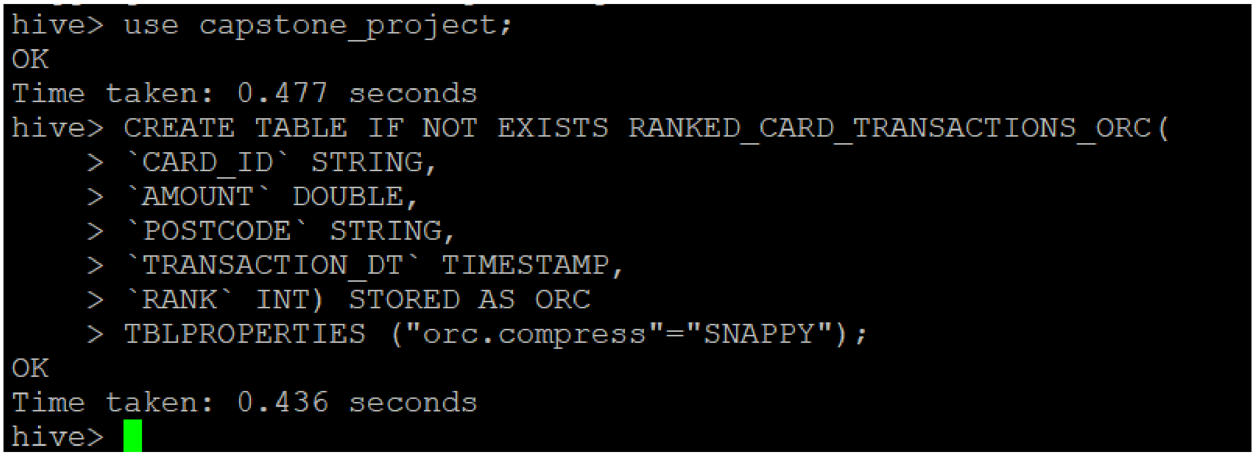
---------- Hbase Operations: Starts Here----------

❖ Task 4: After creating the table, you need to load the relevant data in the lookup table.

---------- **Hive Operations: Starts Here** ----------

1. Start hive and Create table “**ranked_card_transactions_orc**” to store last 10 transactions for each card_id. For better performance.

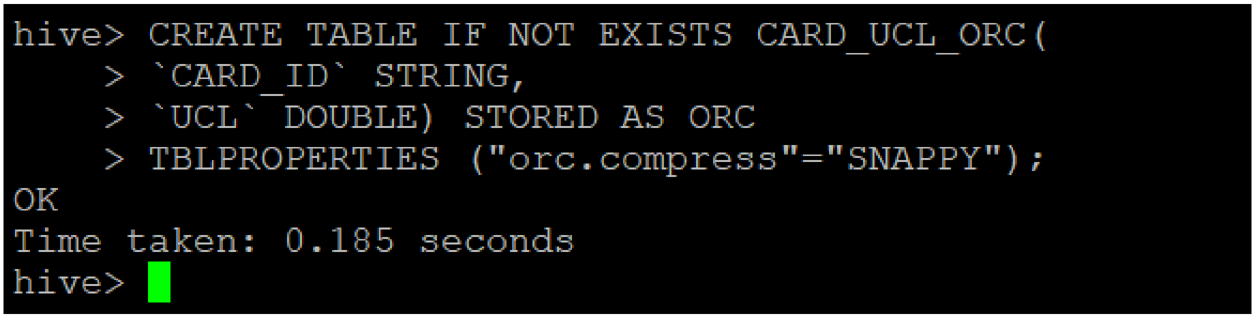
CREATE TABLE IF NOT EXISTS RANKED_CARD_TRANSACTIONS_ORC(
`CARD_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`TRANSACTION_DT` TIMESTAMP,
`RANK` INT) STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");



```
hive> use capstone_project;
OK
Time taken: 0.477 seconds
hive> CREATE TABLE IF NOT EXISTS RANKED_CARD_TRANSACTIONS_ORC(
    > `CARD_ID` STRING,
    > `AMOUNT` DOUBLE,
    > `POSTCODE` STRING,
    > `TRANSACTION_DT` TIMESTAMP,
    > `RANK` INT) STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.436 seconds
hive>
```

2. Create table "**card_ucl_orc**" to store UCL values for each card_id. For better performance.

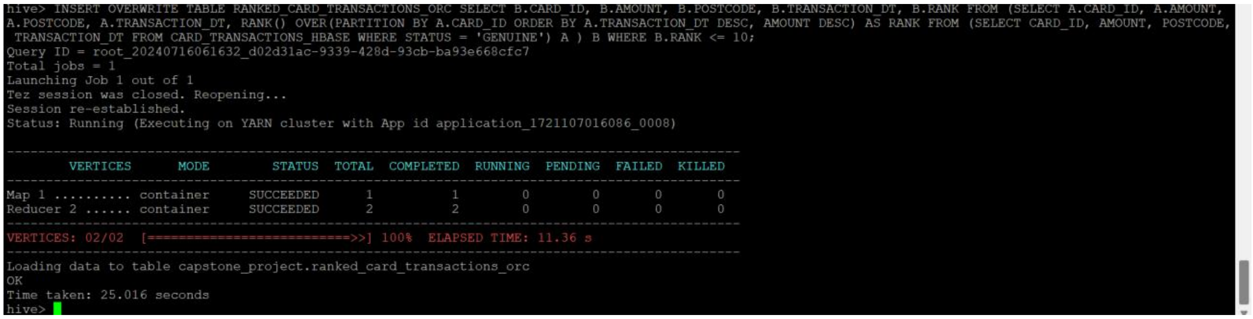
CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
`CARD_ID` STRING,
`UCL` DOUBLE) STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");



```
hive> CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
    > `CARD_ID` STRING,
    > `UCL` DOUBLE) STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.185 seconds
hive>
```

3. Load data in “**ranked_card_transactions_orc**” table

```
INSERT OVERWRITE TABLE RANKED_CARD_TRANSACTIONS_ORC
SELECT B.CARD_ID, B.AMOUNT, B.POSTCODE, B.TRANSACTION_DT, B.RANK
FROM
(SELECT A.CARD_ID, A.AMOUNT, A.POSTCODE, A.TRANSACTION_DT, RANK()
OVER(PARTITION
BY A.CARD_IDORDER BY A.TRANSACTION_DT DESC, AMOUNT DESC) AS RANK
FROM
(SELECT CARD_ID, AMOUNT, POSTCODE, TRANSACTION_DT FROM
CARD_TRANSACTIONS_HBASE WHERESTATUS = 'GENUINE') A ) B WHERE
B.RANK <= 10;
```

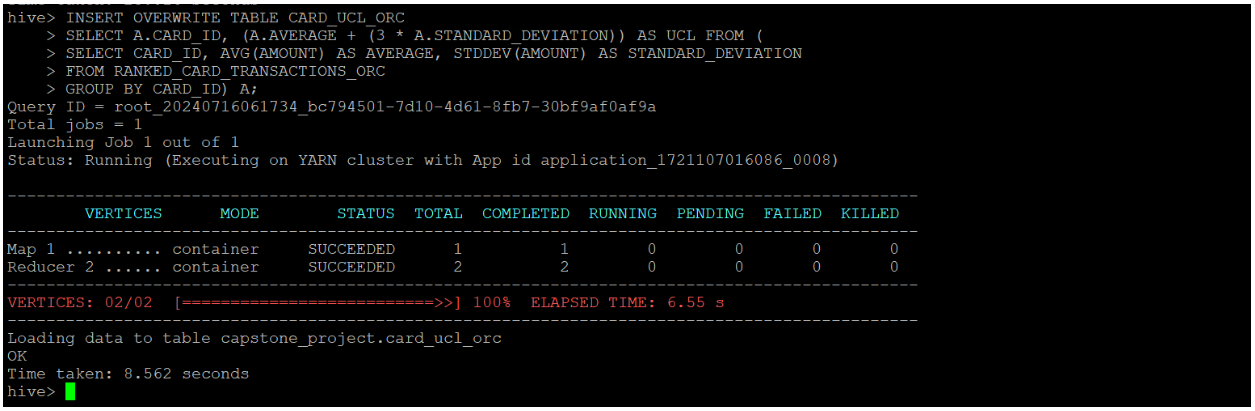


```
hive> INSERT OVERWRITE TABLE RANKED_CARD_TRANSACTIONS_ORC SELECT B.CARD_ID, B.AMOUNT, B.POSTCODE, B.TRANSACTION_DT, B.RANK FROM (SELECT A.CARD_ID, A.AMOUNT,
A.POSTCODE, A.TRANSACTION_DT, RANK() OVER(PARTITION BY A.CARD_ID ORDER BY A.TRANSACTION_DT DESC, AMOUNT DESC) AS RANK FROM (SELECT CARD_ID, AMOUNT, POSTCODE,
 TRANSACTION_DT FROM CARD_TRANSACTIONS_HBASE WHERE STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;
Query ID = root_20240716061632_d02d31ac-9339-428d-93cb-ba93e668cfc7
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1721107016086_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.36 s
----------------------------------------------------------------------------------------------
Loading data to table capstone_project.ranked_card_transactions_orc
OK
Time taken: 25.016 seconds
hive> 
```

4. Load data in "**card_ucl_orc**" table. In innermost query, select card_id, average of amount and standard deviation of amount from card_transactions_orc. In outermost query, select card_id and compute UCL using average and standard deviation with formula (avg + (3 * stddev)). Insert all this data in card_ucl_orc.

```
INSERT OVERWRITE TABLE CARD_UCL_ORC
SELECT A.CARD_ID, (A.AVERAGE + (3 * A.STANDARD_DEVIATION)) AS UCL FROM (
SELECT CARD_ID, AVG(AMOUNT) AS AVERAGE, STDDEV(AMOUNT) AS
STANDARD_DEVIATION
FROM RANKED_CARD_TRANSACTIONS_ORC
GROUP BY CARD_ID) A;
```

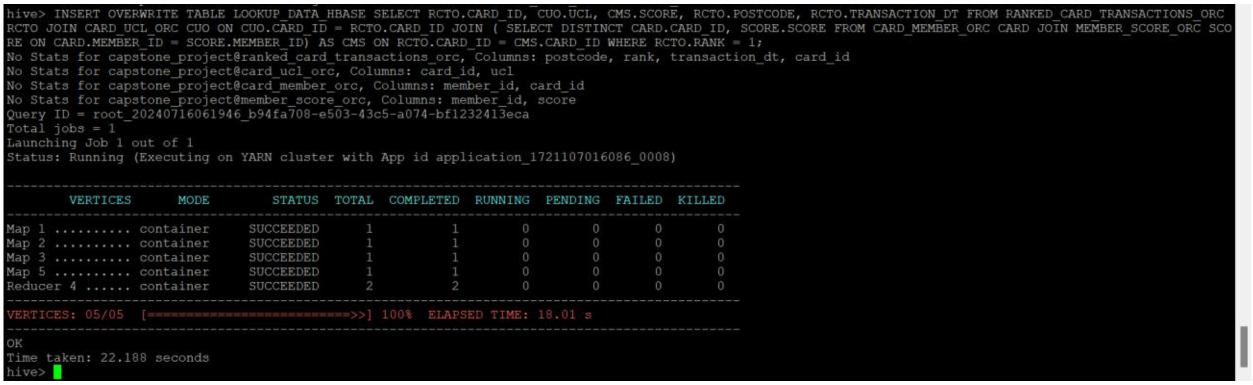


```
hive> INSERT OVERWRITE TABLE CARD_UCL_ORC
    > SELECT A.CARD_ID, (A.AVERAGE + (3 * A.STANDARD_DEVIATION)) AS UCL FROM (
    > SELECT CARD_ID, AVG(AMOUNT) AS AVERAGE, STDDEV(AMOUNT) AS STANDARD_DEVIATION
    > FROM RANKED_CARD_TRANSACTIONS_ORC
    > GROUP BY CARD_ID) A;
Query ID = root_20240716061734_bc794501-7d10-4d61-8fb7-30bf9af0af9a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1721107016086_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.55 s
----------------------------------------------------------------------------------------------
Loading data to table capstone_project.card_ucl_orc
OK
Time taken: 8.562 seconds
hive> 
```

5. Load data in **lookup_data_hbase** table.

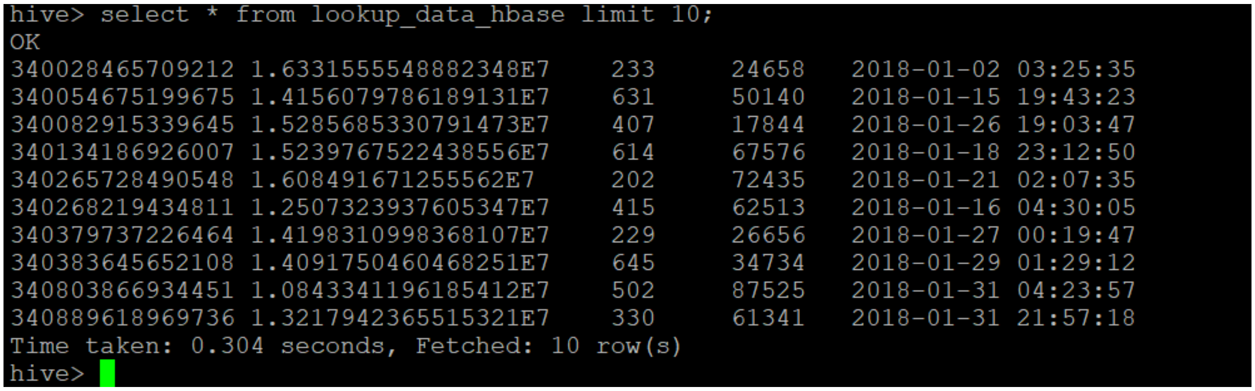
```
INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE
SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE,
RCTO.TRANSACTION_DTFROM RANKED_CARD_TRANSACTIONS_ORC RCTO
JOIN CARD_UCL_ORC CUO
ON CUO.CARD_ID =
RCTO.CARD_IDJOIN (
SELECT DISTINCT CARD.CARD_ID,
SCORE.SCOREFROM
CARD_MEMBER_ORC CARD
JOIN MEMBER_SCORE_ORC SCORE
ON CARD.MEMBER_ID =
SCORE.MEMBER_ID) AS CMSON
RCTO.CARD_ID = CMS.CARD_ID
WHERE RCTO.RANK = 1;
```



```
hive> INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE, RCTO.TRANSACTION_DT FROM RANKED_CARD_TRANSACTIONS_ORC
RCTO JOIN CARD_UCL_ORC CUO ON CUO.CARD_ID = RCTO.CARD_ID JOIN ( SELECT DISTINCT CARD.CARD_ID, SCORE.SCORE FROM CARD_MEMBER_ORC CARD JOIN MEMBER_SCORE_ORC SCO
RE ON CARD.MEMBER_ID = SCORE.MEMBER_ID) AS CMS ON RCTO.CARD_ID = CMS.CARD_ID WHERE RCTO.RANK = 1;
No Stats for capstone_project@ranked_card_transactions_orc, Columns: postcode, rank, transaction_dt, card_id
No Stats for capstone_project@card_ucl_orc, Columns: card_id, ucl
No Stats for capstone_project@card_member_orc, Columns: member_id, card_id
No Stats for capstone_project@member_score_orc, Columns: member_id, score
Query ID = root_20240716061946_b94fa708-e503-43c5-a074-bf1232413eca
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1721107016086_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 2 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 18.01 s
----------------------------------------------------------------------------------------------
OK
Time taken: 22.188 seconds
hive>
```

6. Verify count in “**lookup_data_hbase**” table.
   select count(*) from lookup_data_hbase limit 10;



```
hive> select * from lookup_data_hbase limit 10;
OK
340028465709212 1.6331555548882348E7    233     24658   2018-01-02 03:25:35
340054675199675 1.4156079786189131E7    631     50140   2018-01-15 19:43:23
340082915339645 1.5285685330791473E7    407     17844   2018-01-26 19:03:47
340134186926007 1.5239767522438556E7    614     67576   2018-01-18 23:12:50
340265728490548 1.608491671255562E7     202     72435   2018-01-21 02:07:35
340268219434811 1.2507323937605347E7    415     62513   2018-01-16 04:30:05
340379737226464 1.4198310998368107E7    229     26656   2018-01-27 00:19:47
340383645652108 1.4091750460468251E7    645     34734   2018-01-29 01:29:12
340803866934451 1.0843341196185412E7    502     87525   2018-01-31 04:23:57
340889618969736 1.3217942365515321E7    330     61341   2018-01-31 21:57:18
Time taken: 0.304 seconds, Fetched: 10 row(s)
hive>
```