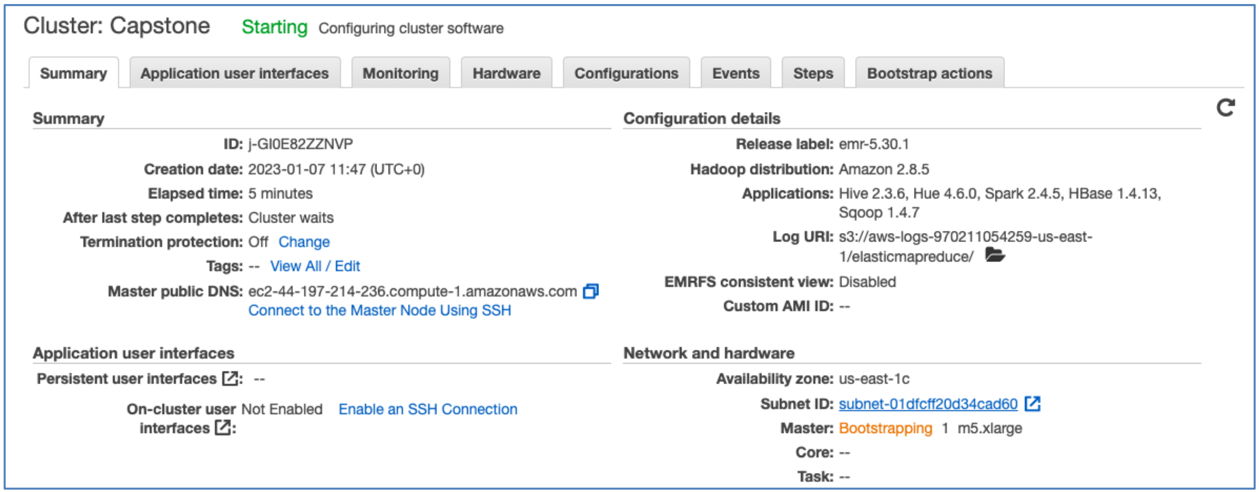
Mid-Submission LOGIC

- **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 and Spark, Root device EBS volume size as 20 GB. I have also updated the Yarn Configurations for EMR instance.

• **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:



Cluster: Capstone Starting Configuring cluster software

Summary

ID: j-GI0E82ZZNVP
Creation date: 2023-01-07 11:47 (UTC+0)
Elapsed time: 5 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS: ec2-44-197-214-236.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

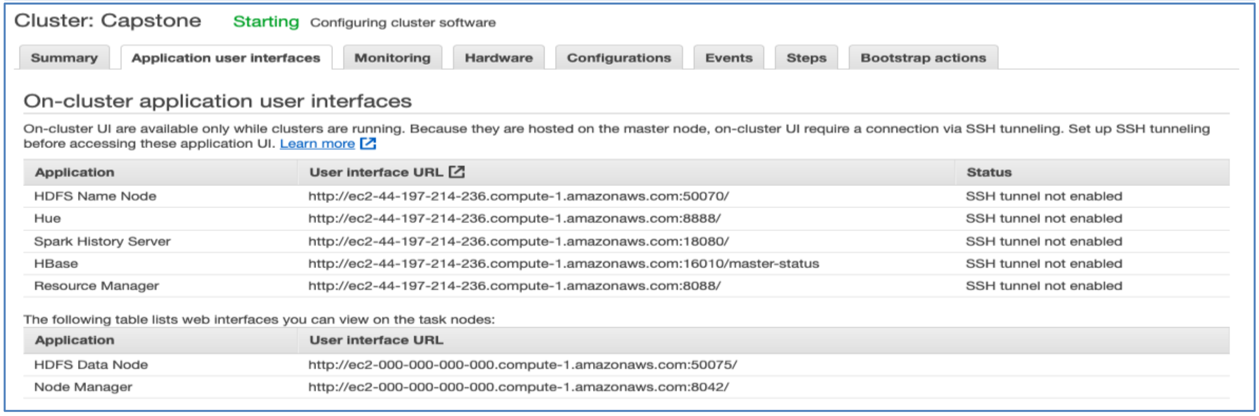
Release label: emr-5.30.1
Hadoop distribution: Amazon 2.8.5
Applications: Hive 2.3.6, Hue 4.6.0, Spark 2.4.5, HBase 1.4.13, Sqoop 1.4.7
Log URI: s3://aws-logs-970211054259-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

Persistent user interfaces: --
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Network and hardware

Availability zone: us-east-1c
Subnet ID: subnet-01dfcff20d34cad60
Master: Bootstrapping 1 m5.xlarge
Core: --
Task: --



Cluster: Capstone Starting Configuring cluster software

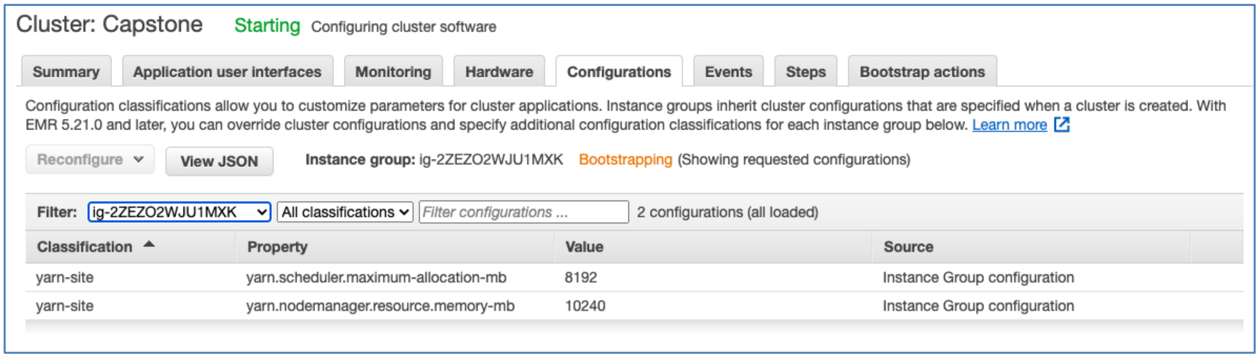
On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. Learn more

Application	User interface URL	Status
HDFS Name Node	http://ec2-44-197-214-236.compute-1.amazonaws.com:50070/	SSH tunnel not enabled
Hue	http://ec2-44-197-214-236.compute-1.amazonaws.com:8888/	SSH tunnel not enabled
Spark History Server	http://ec2-44-197-214-236.compute-1.amazonaws.com:18080/	SSH tunnel not enabled
HBase	http://ec2-44-197-214-236.compute-1.amazonaws.com:16010/master-status	SSH tunnel not enabled
Resource Manager	http://ec2-44-197-214-236.compute-1.amazonaws.com:8088/	SSH tunnel not enabled

The following table lists web interfaces you can view on the task nodes:

Application	User interface URL
HDFS Data Node	http://ec2-000-000-000-000.compute-1.amazonaws.com:50075/
Node Manager	http://ec2-000-000-000-000.compute-1.amazonaws.com:8042/



Cluster: Capstone Starting Configuring cluster software

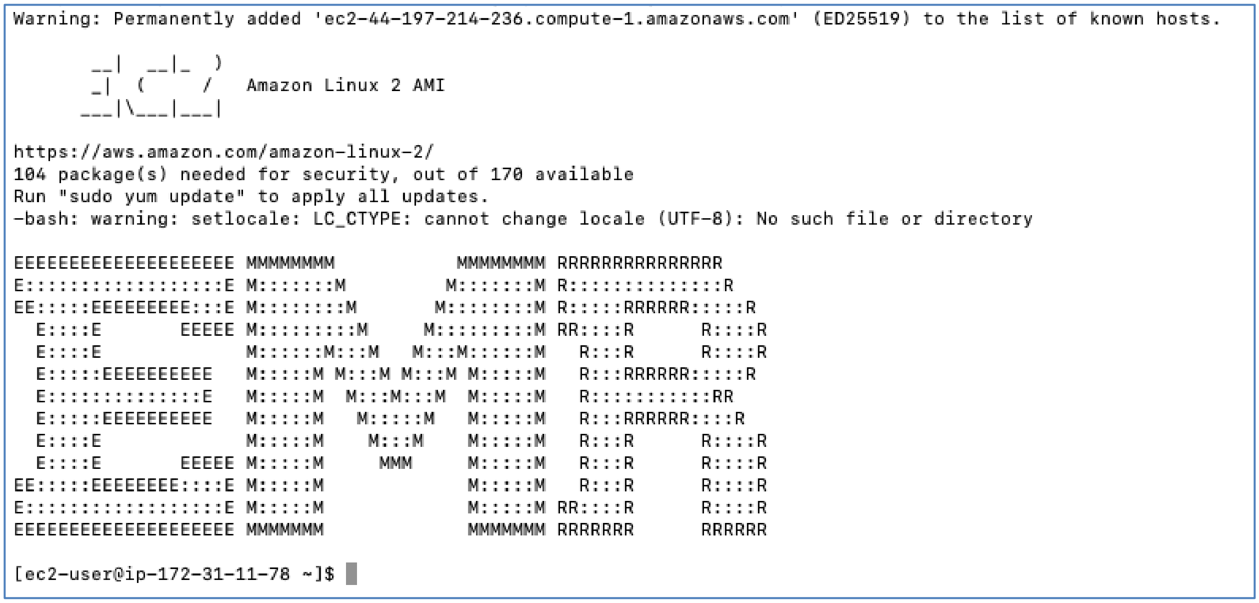
Configuration classifications allow you to customize parameters for cluster applications. Instance groups inherit cluster configurations that are specified when a cluster is created. With EMR 5.21.0 and later, you can override cluster configurations and specify additional configuration classifications for each instance group below. Learn more

Instance group: ig-2ZEZO2WJU1MXK Bootstrapping (Showing requested configurations)

Filter: ig-2ZEZO2WJU1MXK All classifications Filter configurations ... 2 configurations (all loaded)

Classification	Property	Value	Source
yarn-site	yarn.scheduler.maximum-allocation-mb	8192	Instance Group configuration
yarn-site	yarn.nodemanager.resource.memory-mb	10240	Instance Group configuration

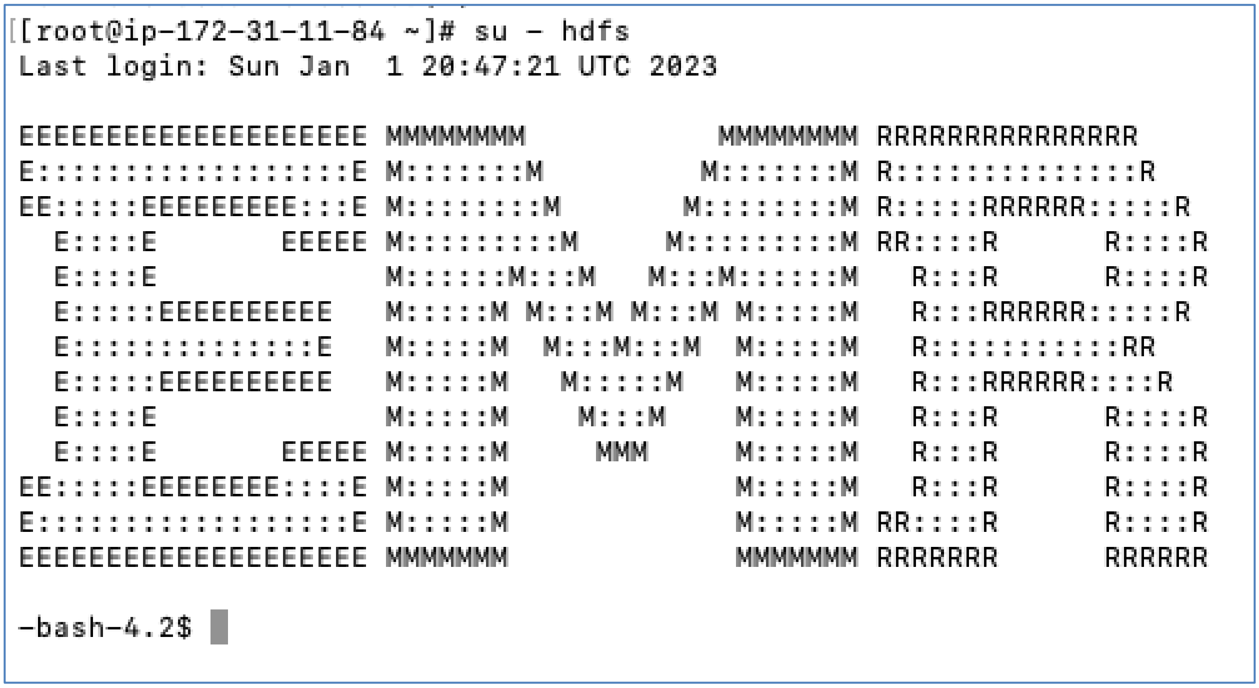
2. Logged into EMR instance as “**ec2-user**”



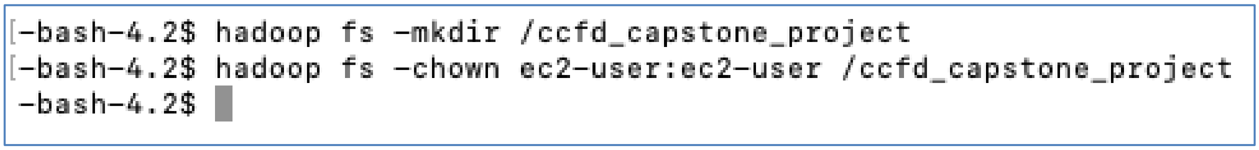
```
Warning: Permanently added 'ec2-44-197-214-236.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
104 package(s) needed for security, out of 170 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory

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

[ec2-user@ip-172-31-11-78 ~]$
```

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 /ccfd_capstone_project
hadoop fs -chown ec2-user:ec2-user /ccfd_capstone_project

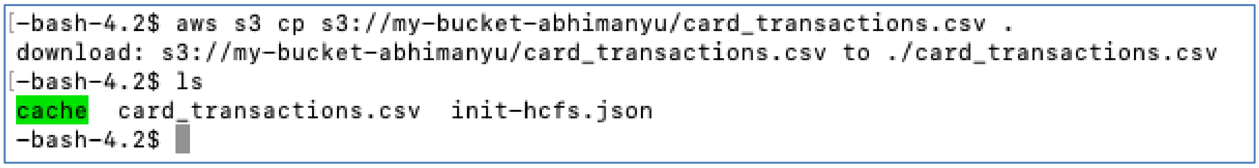


```
[[root@ip-172-31-11-84 ~]# su - hdfs
Last login: Sun Jan  1 20:47:21 UTC 2023

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$
```



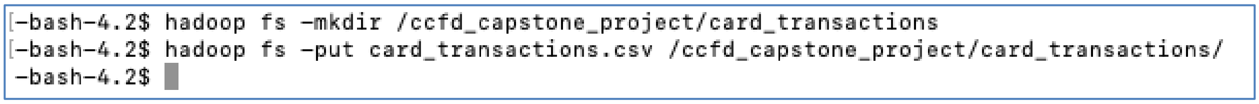
```
[-bash-4.2$ hadoop fs -mkdir /ccfd_capstone_project
[-bash-4.2$ hadoop fs -chown ec2-user:ec2-user /ccfd_capstone_project
-bash-4.2$
```

4. Upload the **card_transaction.csv** provided in S3 and import it to the EMR instance.



```
[-bash-4.2$ aws s3 cp s3://my-bucket-abhimanyu/card_transactions.csv .
download: s3://my-bucket-abhimanyu/card_transactions.csv to ./card_transactions.csv
[-bash-4.2$ ls
cache  card_transactions.csv  init-hcfs.json
-bash-4.2$
```

5. Create a directory in HDFS and copy **card_transactions.csv** in that location.
   hadoop fs -mkdir/ccfd_capstone_project/card_transactions
   hadoop fs -put card_transactions.csv /ccfd_capstone_project/card_transactions/



```
-bash-4.2$ hadoop fs -mkdir /ccfd_capstone_project/card_transactions
-bash-4.2$ hadoop fs -put card_transactions.csv /ccfd_capstone_project/card_transactions/
-bash-4.2$ 
```

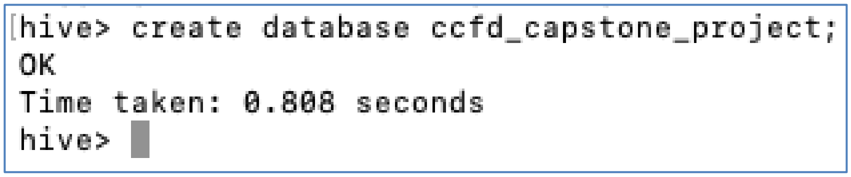
Now as our environment for project is ready. We are starting with completing desired tasks.

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

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

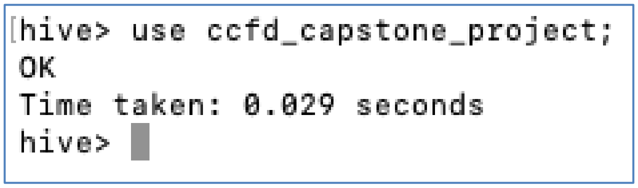
1. Start hive and create new database named **ccfd_capstone_project** -> switch to ccfd_capstone_project database

   create database ccfd_capstone_project;



```
hive> create database ccfd_capstone_project;
OK
Time taken: 0.808 seconds
hive> 
```

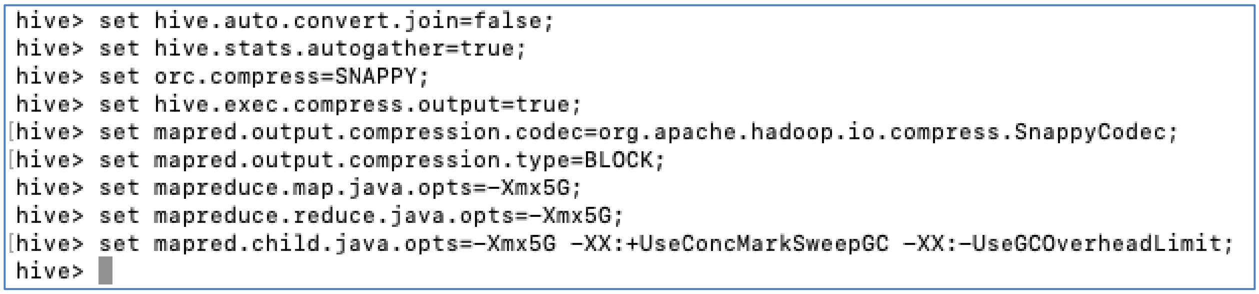
use ccfd_capstone_project;



```
hive> use ccfd_capstone_project;
OK
Time taken: 0.029 seconds
hive> 
```

2. Set below parameters for the hive session

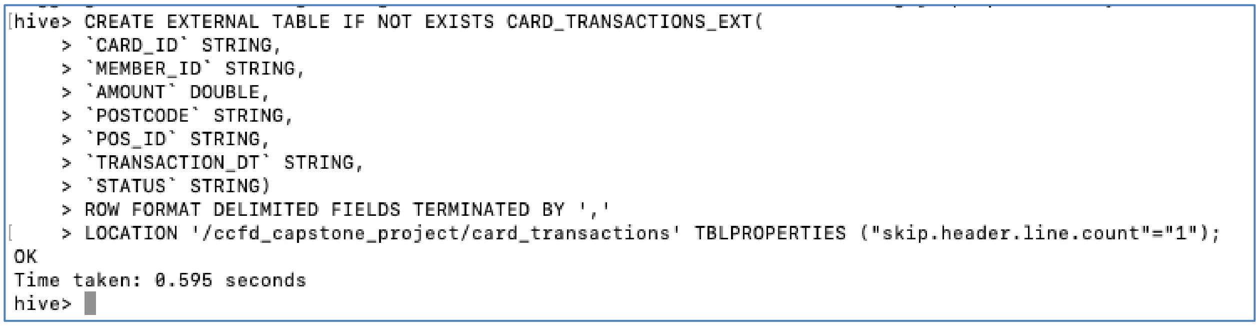
```
set hive.auto.convert.join=false;
set hive.stats.autogather=true;
set orc.compress=SNAPPY;
set hive.exec.compress.output=true;
set mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
set mapred.output.compression.type=BLOCK;
set mapreduce.map.java.opts=-Xmx5G;
set mapreduce.reduce.java.opts=-Xmx5G;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

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

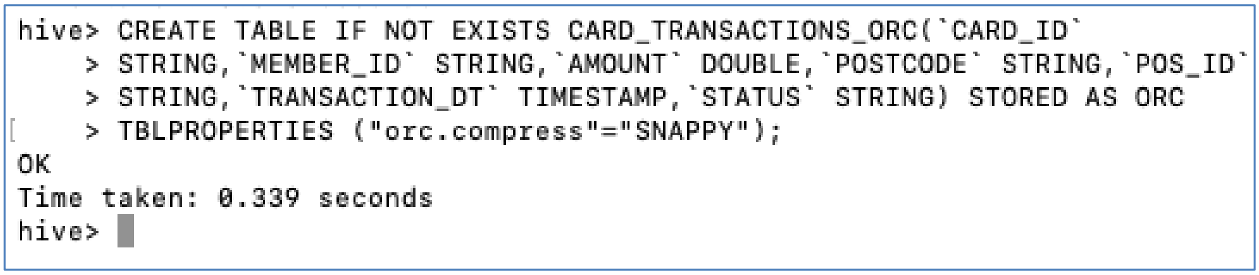
```
CREATE EXTERNAL TABLE IF NOT EXISTS CARD_TRANSACTIONS_EXT(
`CARD_ID` STRING,
`MEMBER_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`POS_ID` STRING,
`TRANSACTION_DT` STRING,
`STATUS` STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/ccfd_capstone_project/card_transactions'
TBLPROPERTIES ("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 '/ccfd_capstone_project/card_transactions' TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 0.595 seconds
hive> 
```

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

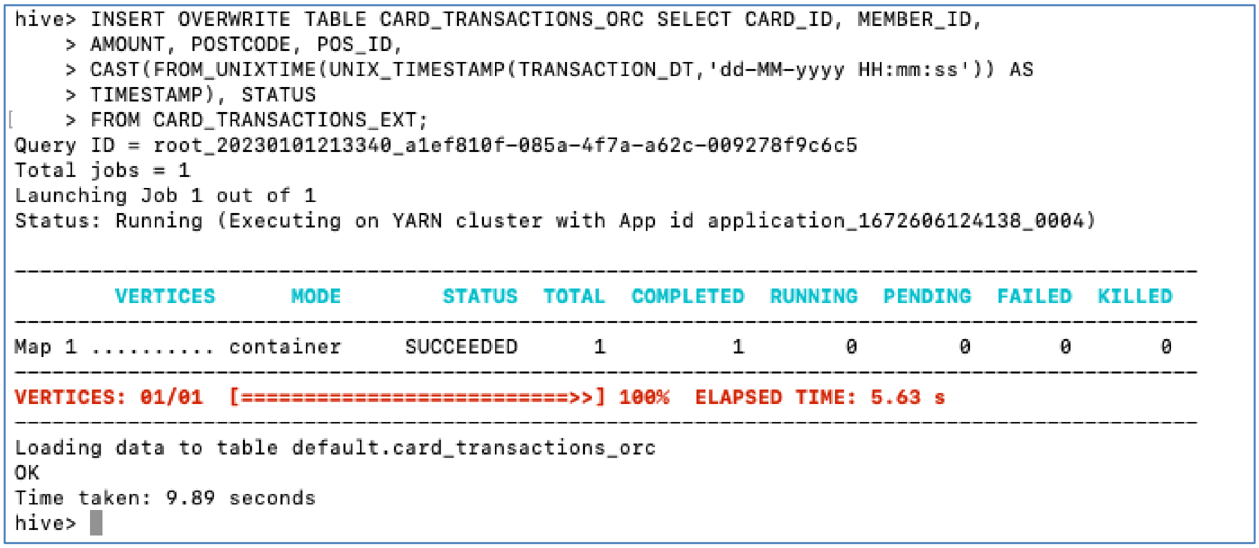
   CREATE TABLE IF NOT EXISTS CARD_TRANSACTIONS_ORC(
   `CARD_ID` STRING,
   `MEMBER_ID` STRING,
   `AMOUNT` DOUBLE,
   `POSTCODE` STRING,
   `POS_ID` STRING,
   `TRANSACTION_DT` TIMESTAMP,
   `STATUS` STRING)
   STORED AS ORC
   TBLPROPERTIES ("orc.compress"="SNAPPY");



```
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: 0.339 seconds
hive>
```

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

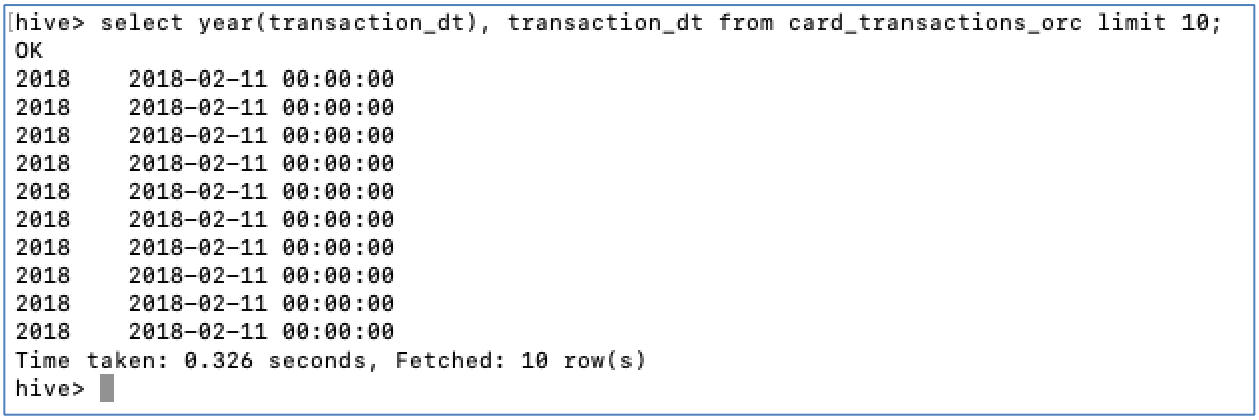
   INSERT OVERWRITE TABLE CARD_TRANSACTIONS_ORC
   SELECT CARD_ID,
   MEMBER_ID,
   AMOUNT,
   POSTCODE,
   POS_ID,
   CAST(FROM_UNIXTIME(UNIX_TIMESTAMP(TRANSACTION_DT,'dd-MM-yyyy HH:mm:ss')) AS TIMESTAMP),
   STATUS FROM CARD_TRANSACTIONS_EXT;



```
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_20230101213340_a1ef810f-085a-4f7a-a62c-009278f9c6c5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1672606124138_0004)

----------------------------------------------------------------------------------------------
        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: 5.63 s
----------------------------------------------------------------------------------------------
Loading data to table default.card_transactions_orc
OK
Time taken: 9.89 seconds
hive>
```

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

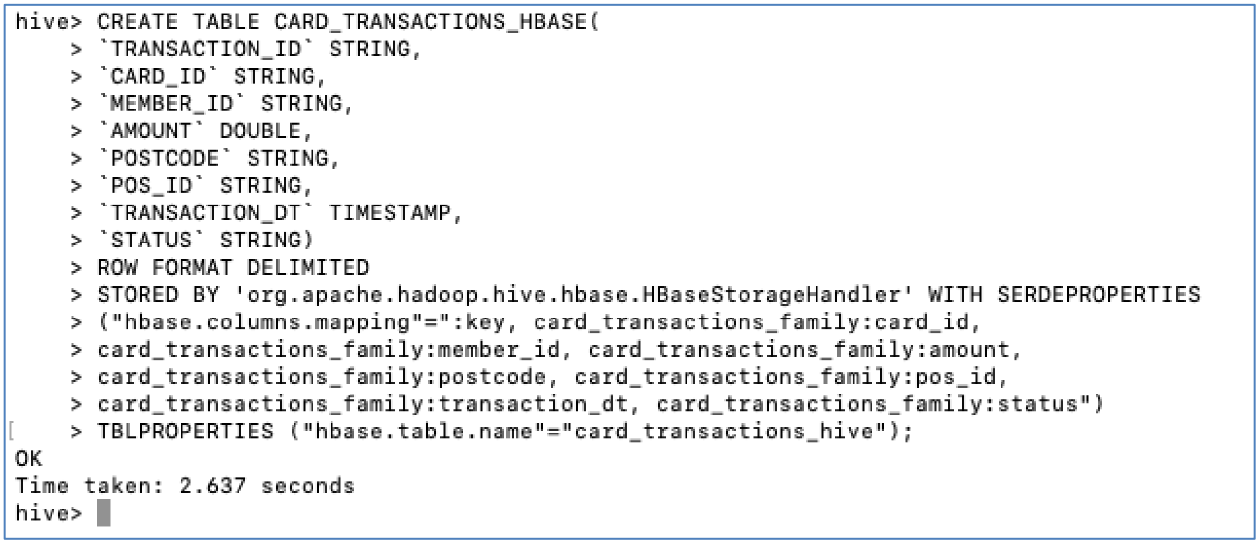
   select year(transaction_dt), transaction_dt from card_transactions_orc limit 10;



```
[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: 0.326 seconds, Fetched: 10 row(s)
hive>
```

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

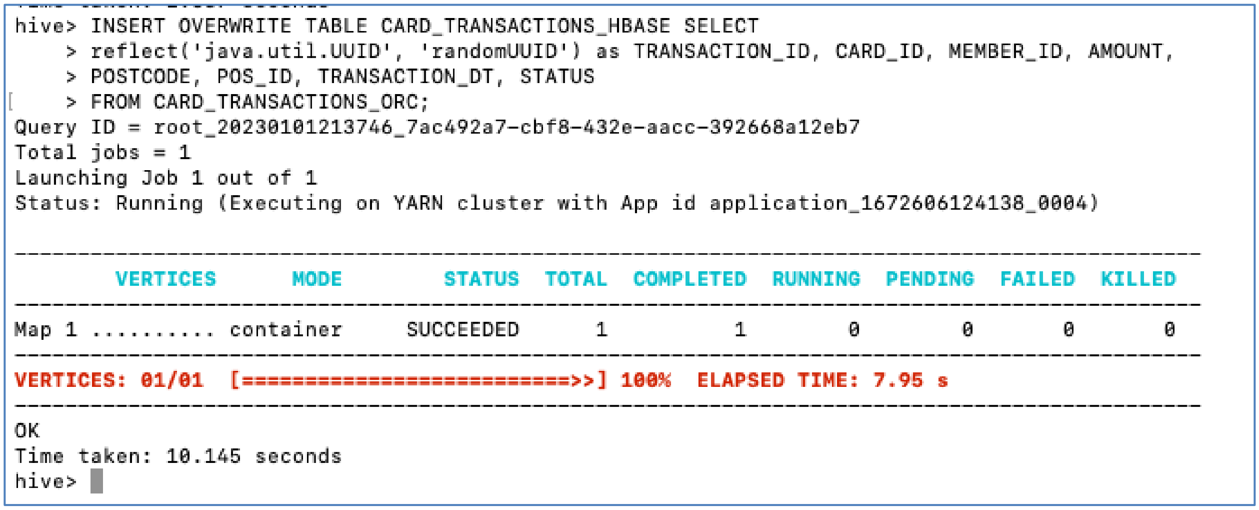
```
CREATE TABLE CARD_TRANSACTIONS_HBASE(
`TRANSACTION_ID` STRING,
`CARD_ID` STRING,
`MEMBER_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`POS_ID` STRING,
`TRANSACTION_DT` TIMESTAMP,
`STATUS` STRING)
ROW FORMAT DELIMITED
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("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: 2.637 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).

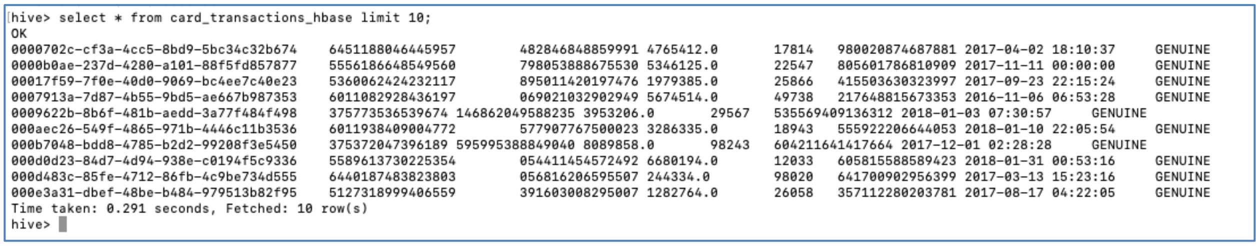
```
INSERT OVERWRITE TABLE CARD_TRANSACTIONS_HBASE
SELECT reflect('java.util.UUID', 'randomUUID') as TRANSACTION_ID,
CARD_ID,
MEMBER_ID,
AMOUNT,
POSTCODE,
POS_ID,
TRANSACTION_DT,
STATUS
FROM CARD_TRANSACTIONS_ORC;
```



```
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_20230101213746_7ac492a7-cbf8-432e-aacc-392668a12eb7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1672606124138_0004)

----------------------------------------------------------------------------------------------
        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: 7.95 s
----------------------------------------------------------------------------------------------
OK
Time taken: 10.145 seconds
hive>
```

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

```
select * from card_transactions_hbase limit 10;
```



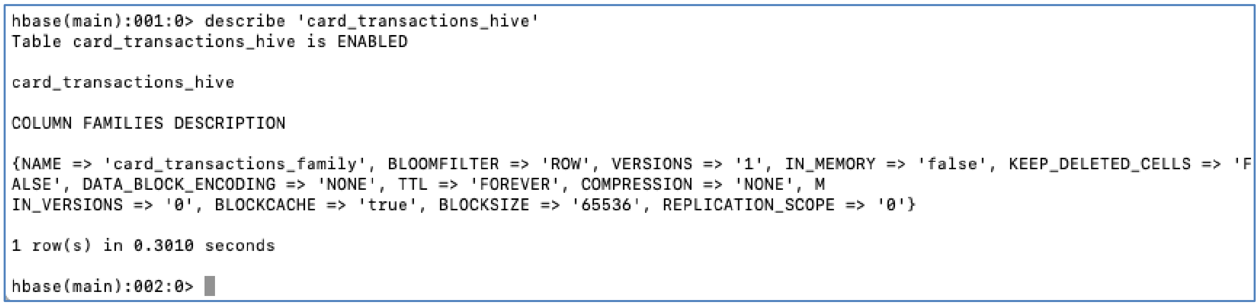
```
hive> select * from card_transactions_hbase limit 10;
OK
0000702c-cf3a-4cc5-8bd9-5bc34c32b674    6451188046445957        482846848859991 4765412.0       17814   980020874687881 2017-04-02 18:10:37     GENUINE
0000b0ae-237d-4280-a101-88f5fd857877    5556186648549560        798053888675530 5346125.0       22547   805601786810909 2017-11-11 00:00:00     GENUINE
00017f59-7f0e-40d0-9069-bc4ee7c40e23    5360062424232117        895011420197476 1979385.0       25866   415503630323997 2017-09-23 22:15:24     GENUINE
0007913a-7d87-4b55-9bd5-ae667b987353    6011082928436197        069021032902949 5674514.0       49738   217648815673353 2016-11-06 06:53:28     GENUINE
0009622b-8b6f-481b-aedd-3a77f484f498    375773536539674 146862049588235 3953206.0       29567   535569409136312 2018-01-03 07:30:57     GENUINE
000aec26-549f-4865-971b-4446c11b3536    6011938409004772        577907767500023 3286335.0       18943   555922206644053 2018-01-10 22:05:54     GENUINE
000b7048-bdd8-4785-b2d2-99208f3e5450    375372047396189 595995388849040 8089858.0       98243   604211641417664 2017-12-01 02:28:28     GENUINE
000d0d23-84d7-4d94-938e-c0194f5c9336    5589613730225354        054411454572492 6680194.0       12033   605815588589423 2018-01-31 00:53:16     GENUINE
000d483c-85fe-4712-86fb-4c9be734d555    6440187483823803        056816206595507 244334.0        98020   641700902956399 2017-03-13 15:23:16     GENUINE
000e3a31-dbef-48be-b484-979513b82f95    5127318999406559        391603008295007 1282764.0       26058   357112280203781 2017-08-17 04:22:05     GENUINE
Time taken: 0.291 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).

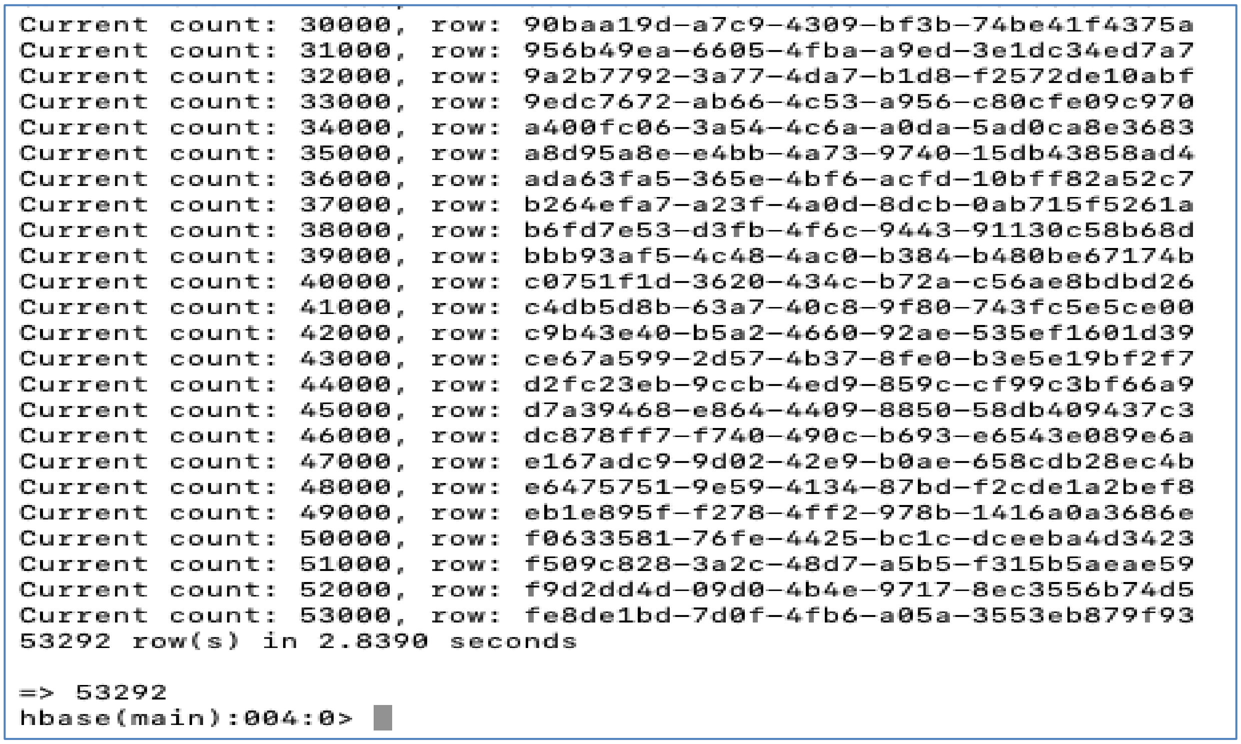
   describe 'card_transactions_hive'



```
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 => 'F
ALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', M
IN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}

1 row(s) in 0.3010 seconds

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

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

   count 'card_transactions_hive'



```
Current count: 30000, row: 90baa19d-a7c9-4309-bf3b-74be41f4375a
Current count: 31000, row: 956b49ea-6605-4fba-a9ed-3e1dc34ed7a7
Current count: 32000, row: 9a2b7792-3a77-4da7-b1d8-f2572de10abf
Current count: 33000, row: 9edc7672-ab66-4c53-a956-c80cfe09c970
Current count: 34000, row: a400fc06-3a54-4c6a-a0da-5ad0ca8e3683
Current count: 35000, row: a8d95a8e-e4bb-4a73-9740-15db43858ad4
Current count: 36000, row: ada63fa5-365e-4bf6-acfd-10bff82a52c7
Current count: 37000, row: b264efa7-a23f-4a0d-8dcb-0ab715f5261a
Current count: 38000, row: b6fd7e53-d3fb-4f6c-9443-91130c58b68d
Current count: 39000, row: bbb93af5-4c48-4ac0-b384-b480be67174b
Current count: 40000, row: c0751f1d-3620-434c-b72a-c56ae8bdbd26
Current count: 41000, row: c4db5d8b-63a7-40c8-9f80-743fc5e5ce00
Current count: 42000, row: c9b43e40-b5a2-4660-92ae-535ef1601d39
Current count: 43000, row: ce67a599-2d57-4b37-8fe0-b3e5e19bf2f7
Current count: 44000, row: d2fc23eb-9ccb-4ed9-859c-cf99c3bf66a9
Current count: 45000, row: d7a39468-e864-4409-8850-58db409437c3
Current count: 46000, row: dc878ff7-f740-490c-b693-e6543e089e6a
Current count: 47000, row: e167adc9-9d02-42e9-b0ae-658cdb28ec4b
Current count: 48000, row: e6475751-9e59-4134-87bd-f2cde1a2bef8
Current count: 49000, row: eb1e895f-f278-4ff2-978b-1416a0a3686e
Current count: 50000, row: f0633581-76fe-4425-bc1c-dceeba4d3423
Current count: 51000, row: f509c828-3a2c-48d7-a5b5-f315b5aeae59
Current count: 52000, row: f9d2dd4d-09d0-4b4e-9717-8ec3556b74d5
Current count: 53000, row: fe8de1bd-7d0f-4fb6-a05a-3553eb879f93
53292 row(s) in 2.8390 seconds

=> 53292
hbase(main):004:0>
```

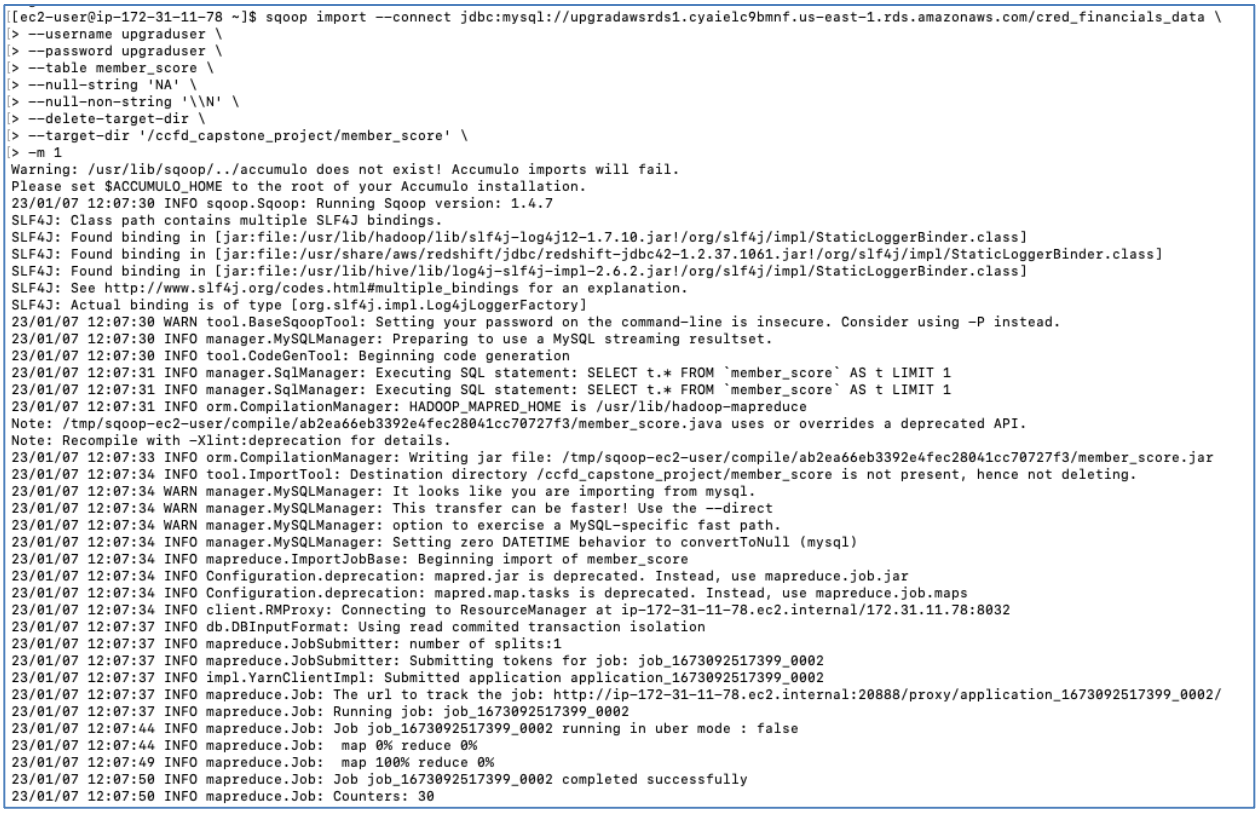
-- Hbase Operations: Ends Here -------------------------------------

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

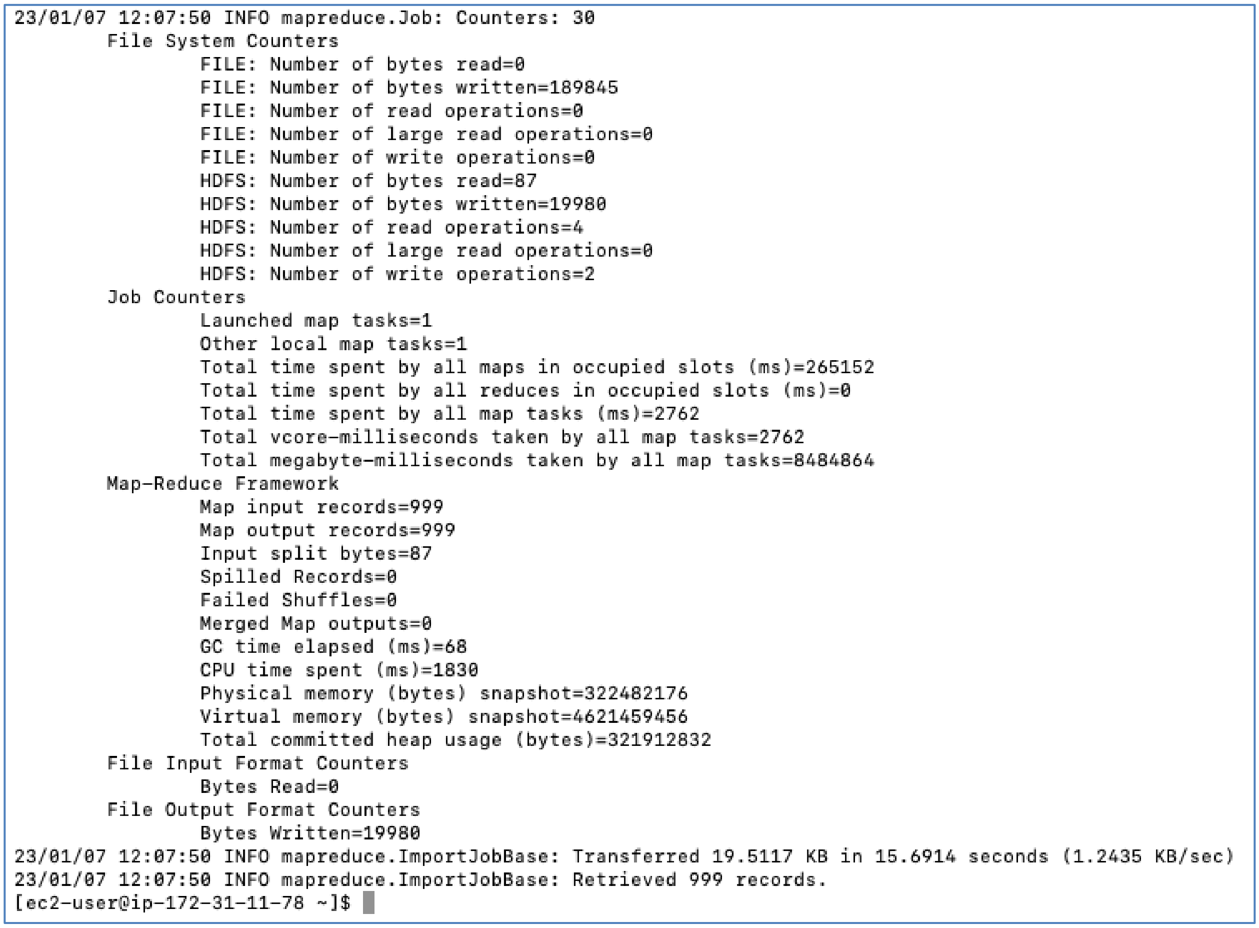
- **Task 2**: Ingest the relevant data from AWS RDS to Hadoop.

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

```
sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-
east1.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 '/ccfd_capstone_project/member_score' \
-m 1
```



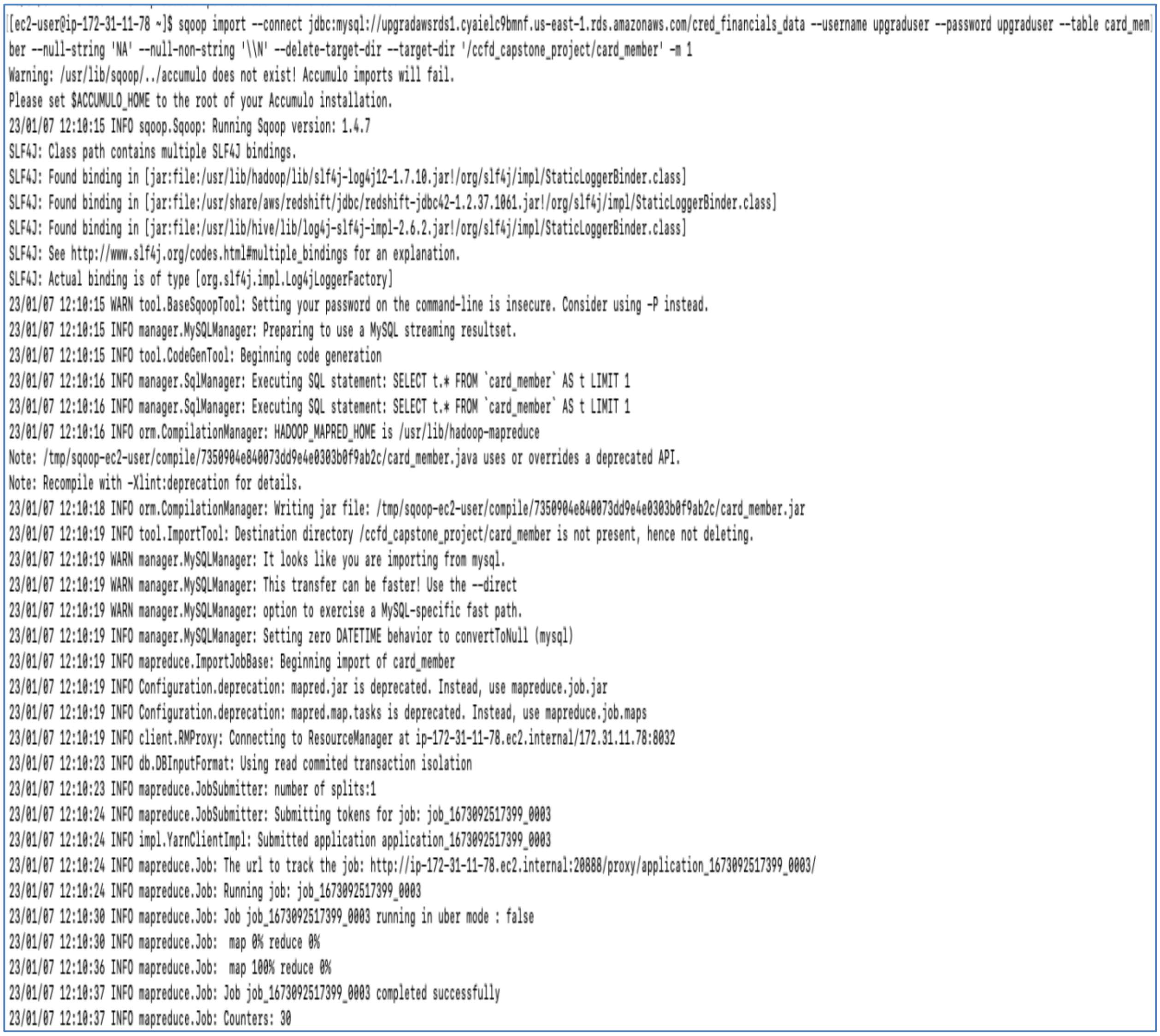
```
[ec2-user@ip-172-31-11-78 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data \
> --username upgraduser \
> --password upgraduser \
> --table member_score \
> --null-string 'NA' \
> --null-non-string '\\N' \
> --delete-target-dir \
> --target-dir '/ccfd_capstone_project/member_score' \
> -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/01/07 12:07:30 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/redshift/jdbc/redshift-jdbc42-1.2.37.1061.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/01/07 12:07:30 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/01/07 12:07:30 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/01/07 12:07:30 INFO tool.CodeGenTool: Beginning code generation
23/01/07 12:07:31 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
23/01/07 12:07:31 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
23/01/07 12:07:31 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-ec2-user/compile/ab2ea66eb3392e4fec28041cc70727f3/member_score.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/01/07 12:07:33 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-ec2-user/compile/ab2ea66eb3392e4fec28041cc70727f3/member_score.jar
23/01/07 12:07:34 INFO tool.ImportTool: Destination directory /ccfd_capstone_project/member_score is not present, hence not deleting.
23/01/07 12:07:34 WARN manager.MySQLManager: It looks like you are importing from mysql.
23/01/07 12:07:34 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
23/01/07 12:07:34 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
23/01/07 12:07:34 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
23/01/07 12:07:34 INFO mapreduce.ImportJobBase: Beginning import of member_score
23/01/07 12:07:34 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
23/01/07 12:07:34 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
23/01/07 12:07:34 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-11-78.ec2.internal/172.31.11.78:8032
23/01/07 12:07:37 INFO db.DBInputFormat: Using read commited transaction isolation
23/01/07 12:07:37 INFO mapreduce.JobSubmitter: number of splits:1
23/01/07 12:07:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1673092517399_0002
23/01/07 12:07:37 INFO impl.YarnClientImpl: Submitted application application_1673092517399_0002
23/01/07 12:07:37 INFO mapreduce.Job: The url to track the job: http://ip-172-31-11-78.ec2.internal:20888/proxy/application_1673092517399_0002/
23/01/07 12:07:37 INFO mapreduce.Job: Running job: job_1673092517399_0002
23/01/07 12:07:44 INFO mapreduce.Job: Job job_1673092517399_0002 running in uber mode : false
23/01/07 12:07:44 INFO mapreduce.Job:  map 0% reduce 0%
23/01/07 12:07:49 INFO mapreduce.Job:  map 100% reduce 0%
23/01/07 12:07:50 INFO mapreduce.Job: Job job_1673092517399_0002 completed successfully
23/01/07 12:07:50 INFO mapreduce.Job: Counters: 30
```



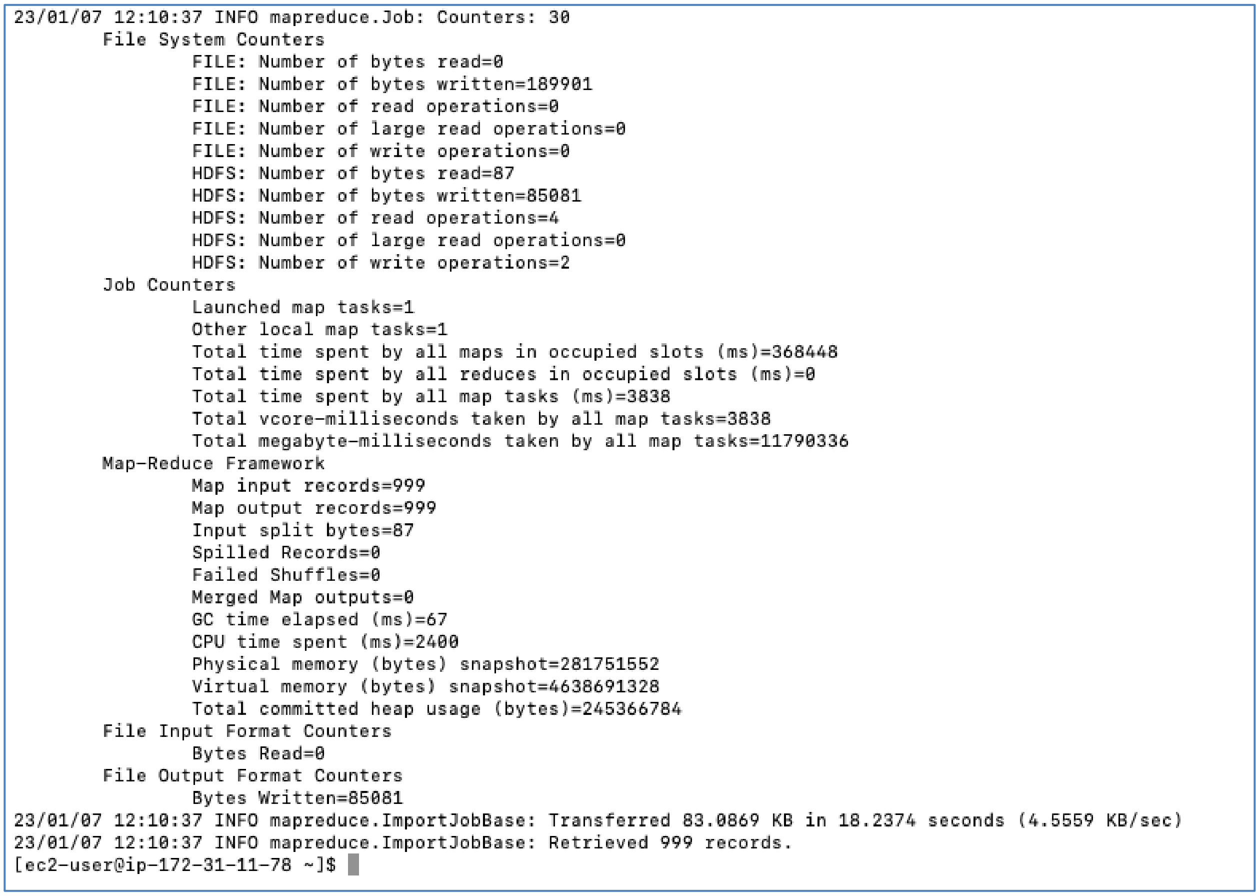
```
23/01/07 12:07:50 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=189845
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=19980
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=265152
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=2762
                Total vcore-milliseconds taken by all map tasks=2762
                Total megabyte-milliseconds taken by all map tasks=8484864
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=68
                CPU time spent (ms)=1830
                Physical memory (bytes) snapshot=322482176
                Virtual memory (bytes) snapshot=4621459456
                Total committed heap usage (bytes)=321912832
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=19980
23/01/07 12:07:50 INFO mapreduce.ImportJobBase: Transferred 19.5117 KB in 15.6914 seconds (1.2435 KB/sec)
23/01/07 12:07:50 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[ec2-user@ip-172-31-11-78 ~]$ 
```

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

```
sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-
east1.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 '/ccfd_capstone_project/card_member' \
-m 1
```



```
[ec2-user@ip-172-31-11-78 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data --username upgraduser --password upgraduser --table card_mem
ber --null-string 'NA' --null-non-string '\\N' --delete-target-dir --target-dir '/ccfd_capstone_project/card_member' -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/01/07 12:10:15 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/redshift/jdbc/redshift-jdbc42-1.2.37.1061.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/01/07 12:10:15 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/01/07 12:10:15 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/01/07 12:10:15 INFO tool.CodeGenTool: Beginning code generation
23/01/07 12:10:16 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
23/01/07 12:10:16 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
23/01/07 12:10:16 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-ec2-user/compile/7350904e840073dd9e4e0303b0f9ab2c/card_member.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/01/07 12:10:18 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-ec2-user/compile/7350904e840073dd9e4e0303b0f9ab2c/card_member.jar
23/01/07 12:10:19 INFO tool.ImportTool: Destination directory /ccfd_capstone_project/card_member is not present, hence not deleting.
23/01/07 12:10:19 WARN manager.MySQLManager: It looks like you are importing from mysql.
23/01/07 12:10:19 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
23/01/07 12:10:19 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
23/01/07 12:10:19 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
23/01/07 12:10:19 INFO mapreduce.ImportJobBase: Beginning import of card_member
23/01/07 12:10:19 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
23/01/07 12:10:19 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
23/01/07 12:10:19 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-11-78.ec2.internal/172.31.11.78:8032
23/01/07 12:10:23 INFO db.DBInputFormat: Using read commited transaction isolation
23/01/07 12:10:23 INFO mapreduce.JobSubmitter: number of splits:1
23/01/07 12:10:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1673092517399_0003
23/01/07 12:10:24 INFO impl.YarnClientImpl: Submitted application application_1673092517399_0003
23/01/07 12:10:24 INFO mapreduce.Job: The url to track the job: http://ip-172-31-11-78.ec2.internal:20888/proxy/application_1673092517399_0003/
23/01/07 12:10:24 INFO mapreduce.Job: Running job: job_1673092517399_0003
23/01/07 12:10:30 INFO mapreduce.Job: Job job_1673092517399_0003 running in uber mode : false
23/01/07 12:10:30 INFO mapreduce.Job:  map 0% reduce 0%
23/01/07 12:10:36 INFO mapreduce.Job:  map 100% reduce 0%
23/01/07 12:10:37 INFO mapreduce.Job: Job job_1673092517399_0003 completed successfully
23/01/07 12:10:37 INFO mapreduce.Job: Counters: 30
```



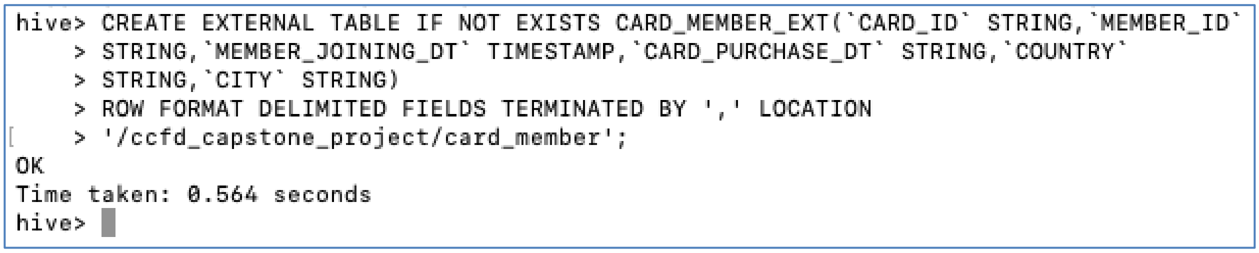
```
23/01/07 12:10:37 INFO mapreduce.Job: Counters: 30
	File System Counters
		FILE: Number of bytes read=0
		FILE: Number of bytes written=189901
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=87
		HDFS: Number of bytes written=85081
		HDFS: Number of read operations=4
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Other local map tasks=1
		Total time spent by all maps in occupied slots (ms)=368448
		Total time spent by all reduces in occupied slots (ms)=0
		Total time spent by all map tasks (ms)=3838
		Total vcore-milliseconds taken by all map tasks=3838
		Total megabyte-milliseconds taken by all map tasks=11790336
	Map-Reduce Framework
		Map input records=999
		Map output records=999
		Input split bytes=87
		Spilled Records=0
		Failed Shuffles=0
		Merged Map outputs=0
		GC time elapsed (ms)=67
		CPU time spent (ms)=2400
		Physical memory (bytes) snapshot=281751552
		Virtual memory (bytes) snapshot=4638691328
		Total committed heap usage (bytes)=245366784
	File Input Format Counters
		Bytes Read=0
	File Output Format Counters
		Bytes Written=85081
23/01/07 12:10:37 INFO mapreduce.ImportJobBase: Transferred 83.0869 KB in 18.2374 seconds (4.5559 KB/sec)
23/01/07 12:10:37 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[ec2-user@ip-172-31-11-78 ~]$ 
```

--- 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.

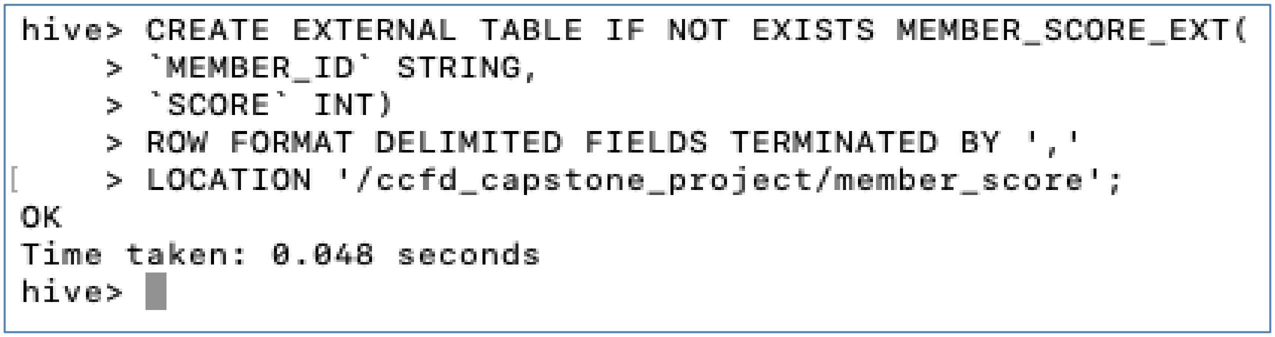
   CREATE EXTERNAL TABLE IF NOT EXISTS CARD_MEMBER_EXT(`CARD_ID` STRING,
   `MEMBER_ID` STRING,
   `MEMBER_JOINING_DT` TIMESTAMP,
   `CARD_PURCHASE_DT` STRING,
   `COUNTRY` STRING,
   `CITY` STRING)
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/ccfd_capstone_project/card_member';



```
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
    > '/ccfd_capstone_project/card_member';
OK
Time taken: 0.564 seconds
hive>
```

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

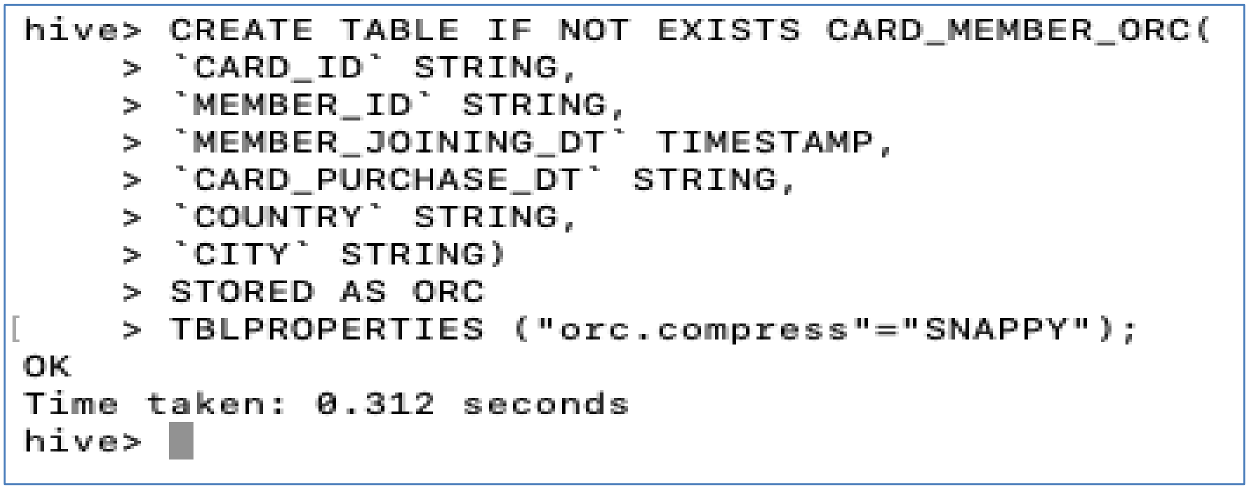
   CREATE EXTERNAL TABLE IF NOT EXISTS MEMBER_SCORE_EXT(
   `MEMBER_ID` STRING,
   `SCORE` INT)
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/ccfd_capstone_project/member_score';



```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS MEMBER_SCORE_EXT(
    > `MEMBER_ID` STRING,
    > `SCORE` INT)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LOCATION '/ccfd_capstone_project/member_score';
OK
Time taken: 0.048 seconds
hive>
```

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

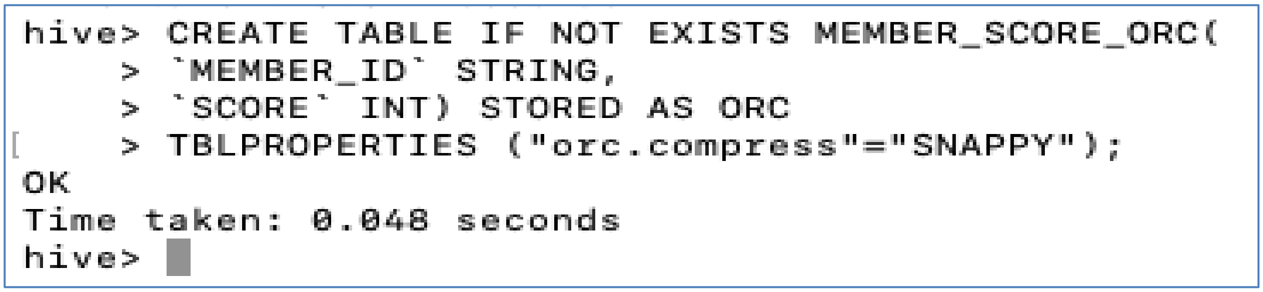
   CREATE TABLE IF NOT EXISTS CARD_MEMBER_ORC(
   `CARD_ID` STRING,
   `MEMBER_ID` STRING,
   `MEMBER_JOINING_DT` TIMESTAMP,
   `CARD_PURCHASE_DT` STRING,
   `COUNTRY` STRING, `CITY` STRING)
   STORED AS ORC TBLPROPERTIES ("orc.compress"="SNAPPY");



```
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: 0.312 seconds
hive>
```

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

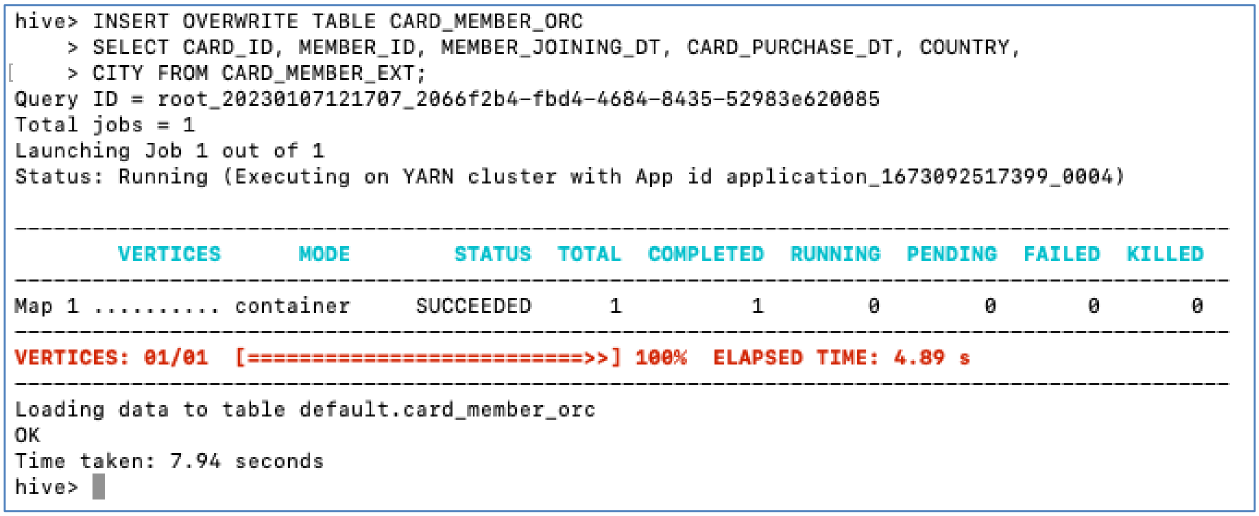
   CREATE TABLE IF NOT EXISTS MEMBER_SCORE_ORC(
   `MEMBER_ID` STRING,
   `SCORE` INT)
   STORED AS ORC
   TBLPROPERTIES ("orc.compress"="SNAPPY");



```
hive> CREATE TABLE IF NOT EXISTS MEMBER_SCORE_ORC(
    > `MEMBER_ID` STRING,
    > `SCORE` INT) STORED AS ORC
[   > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.048 seconds
hive>
```

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

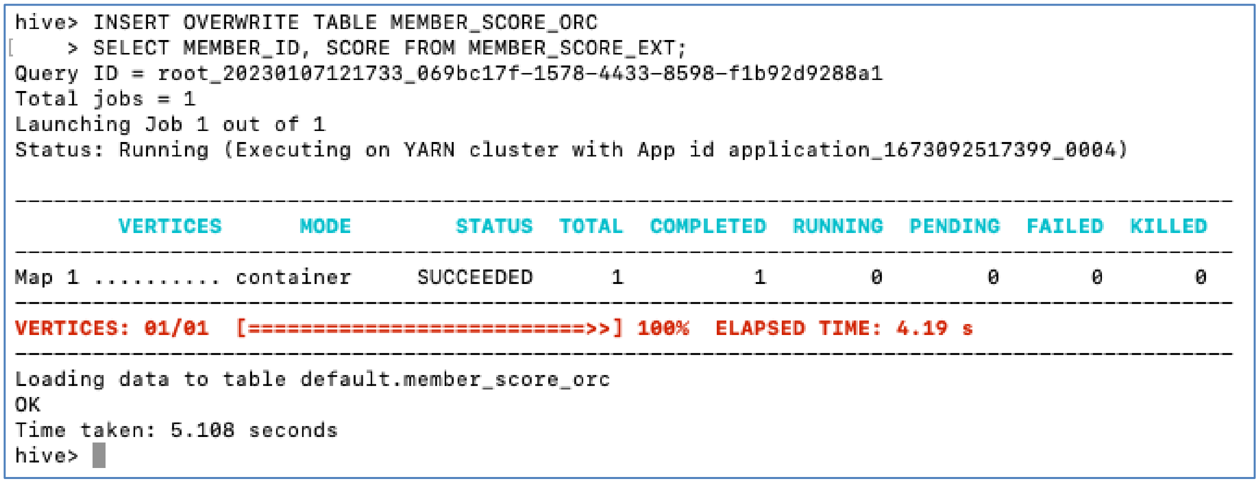
   INSERT OVERWRITE TABLE CARD_MEMBER_ORC
   SELECT CARD_ID,
   MEMBER_ID,
   MEMBER_JOINING_DT,
   CARD_PURCHASE_DT,
   COUNTRY,
   CITY
   FROM CARD_MEMBER_EXT;



```
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_20230107121707_2066f2b4-fbd4-4684-8435-52983e620085
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1673092517399_0004)

----------------------------------------------------------------------------------------------
        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: 4.89 s
----------------------------------------------------------------------------------------------
Loading data to table default.card_member_orc
OK
Time taken: 7.94 seconds
hive>
```

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

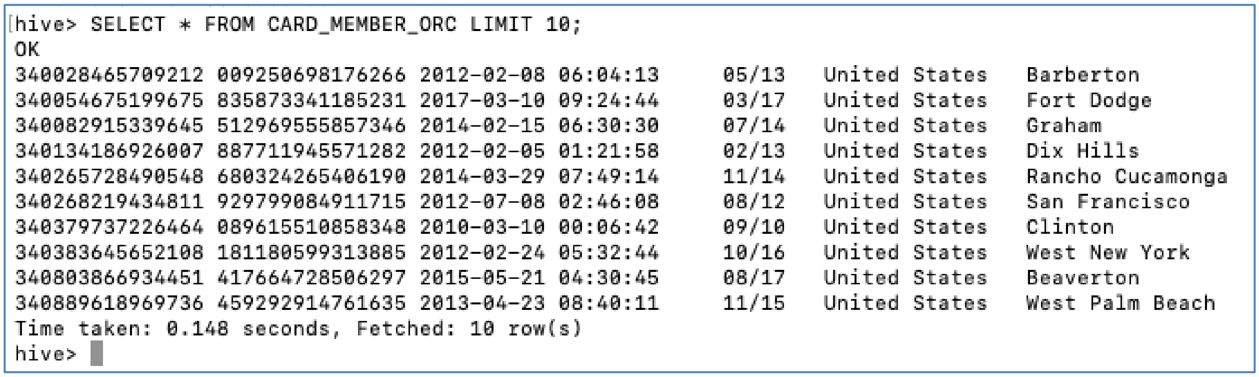
   INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
   SELECT MEMBER_ID,
   SCORE
   FROM MEMBER_SCORE_EXT;



```
hive> INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
    > SELECT MEMBER_ID, SCORE FROM MEMBER_SCORE_EXT;
Query ID = root_20230107121733_069bc17f-1578-4433-8598-f1b92d9288a1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1673092517399_0004)

----------------------------------------------------------------------------------------------
        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: 4.19 s
----------------------------------------------------------------------------------------------
Loading data to table default.member_score_orc
OK
Time taken: 5.108 seconds
hive>
```

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

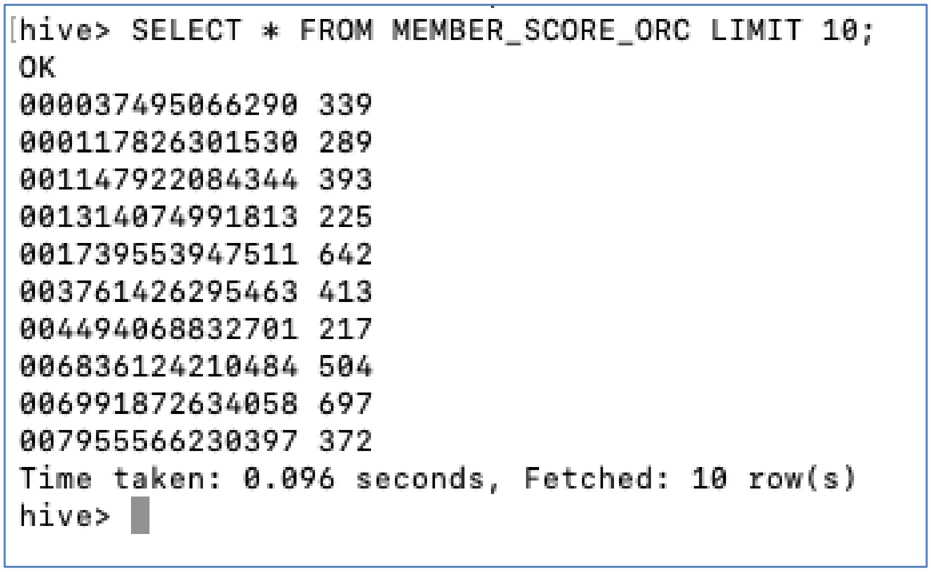
   SELECT * FROM CARD_MEMBER_ORC LIMIT 10;



```
[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: 0.148 seconds, Fetched: 10 row(s)
hive>
```

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

   SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;



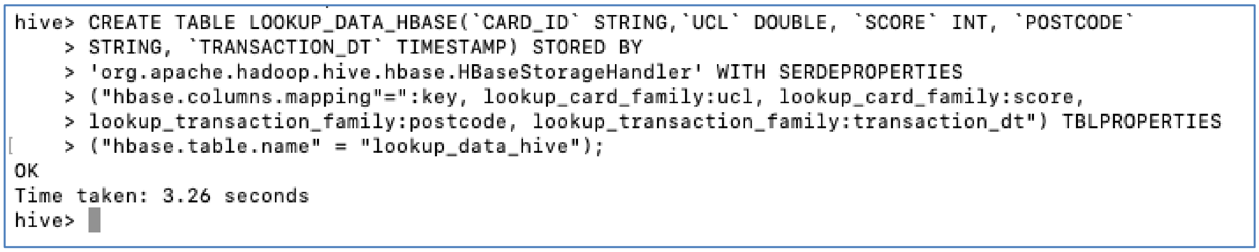
```
[hive> SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;
OK
000037495066290 339
000117826301530 289
001147922084344 393
001314074991813 225
001739553947511 642
003761426295463 413
004494068832701 217
006836124210484 504
006991872634058 697
007955566230397 372
Time taken: 0.096 seconds, Fetched: 10 row(s)
hive>
```

- **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");



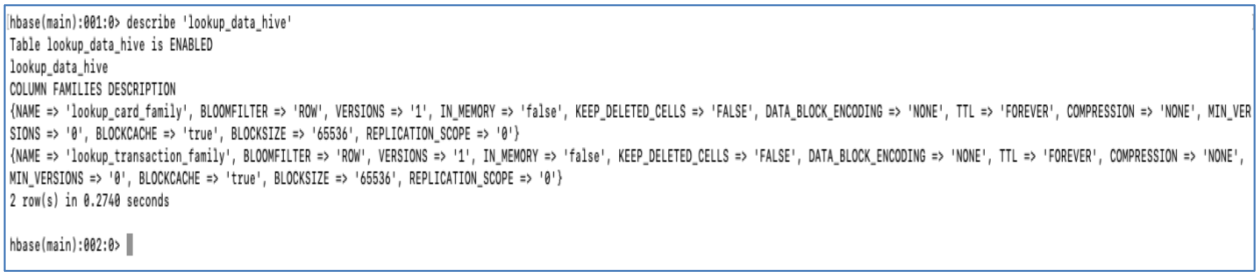
```
hive> CREATE TABLE LOOKUP_DATA_HBASE(`CARD_ID` STRING,`UCL` DOUBLE, `SCORE` INT, `POSTCODE`
    > STRING, `TRANSACTION_DT` TIMESTAMP) STORED BY
    > 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES
    > ("hbase.columns.mapping"=":key, lookup_card_family:ucl, lookup_card_family:score,
    > lookup_transaction_family:postcode, lookup_transaction_family:transaction_dt") TBLPROPERTIES
    > ("hbase.table.name" = "lookup_data_hive");
OK
Time taken: 3.26 seconds
hive>
```

-- Hive Operations: Ends Here--

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

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

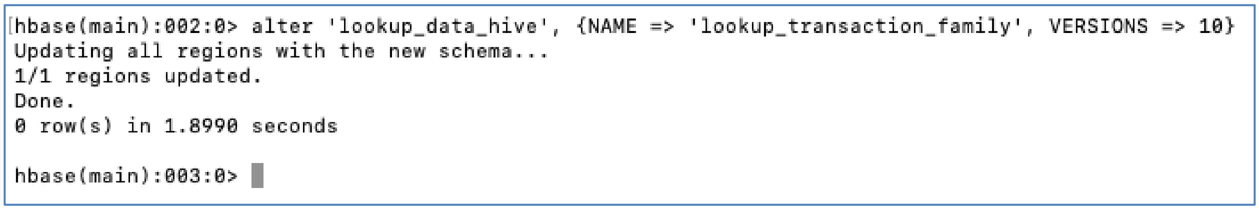
   describe 'lookup_data_hive'



```
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_VER
SIONS => '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 => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE',
MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.2740 seconds

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

2. 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.

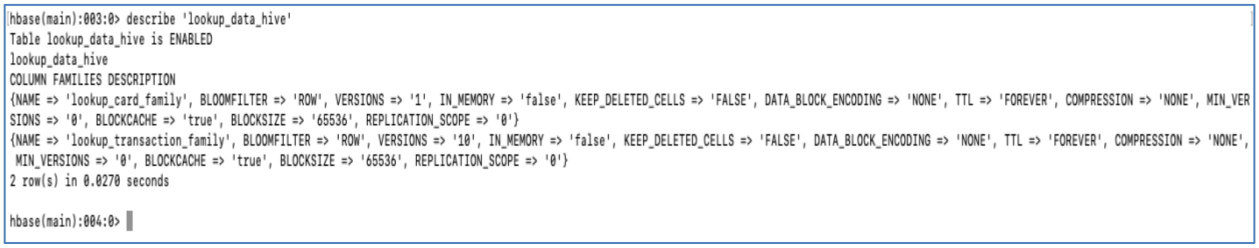
   alter 'lookup_data_hive', {NAME => 'lookup_transaction_family', VERSIONS => 10}



```
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 1.8990 seconds

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

3. 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_VER
SIONS => '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 => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE',
 MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0270 seconds

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

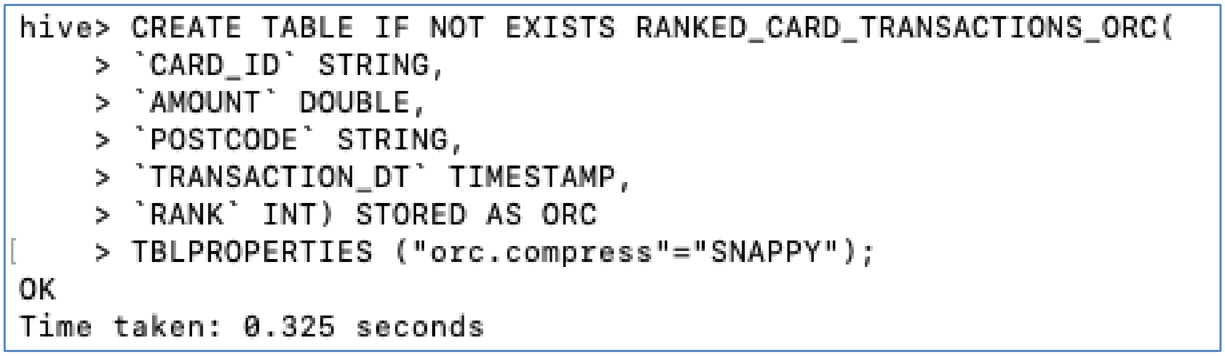
---------------------------------------Hbase Operations: Ends Here---------------------------------------

- **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.

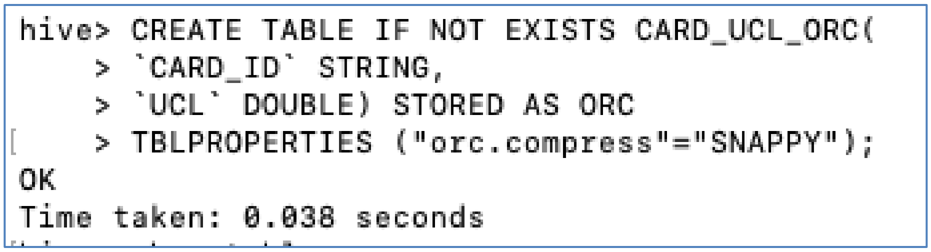
```
CREATE TABLE IF NOT EXISTS RANKED_CARD_TRANSACTIONS_ORC(
`CARD_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`TRANSACTION_DT` TIMESTAMP,
`RANK` INT)
STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");
```



```
hive> 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: 0.325 seconds
```

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

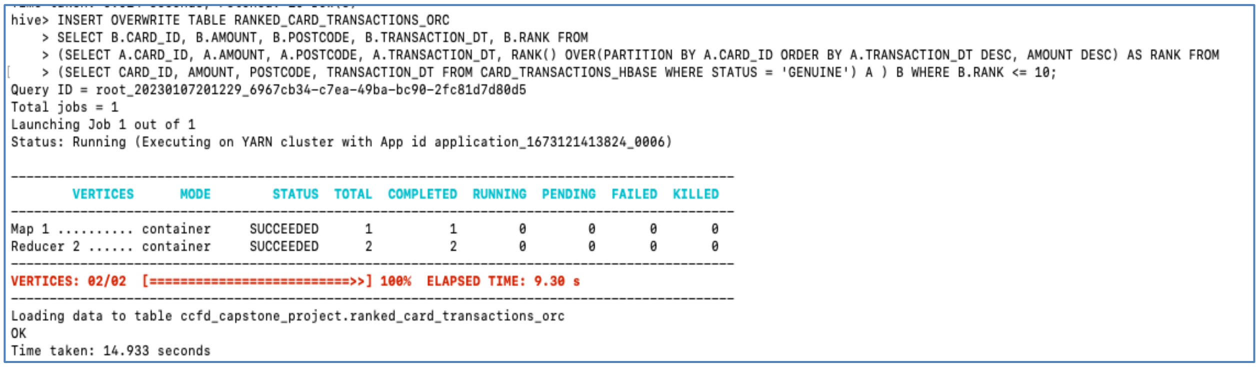
```
CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
`CARD_ID` STRING,
`UCL` DOUBLE)
STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");
```



```
hive> CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
    > `CARD_ID` STRING,
    > `UCL` DOUBLE) STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.038 seconds
```

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

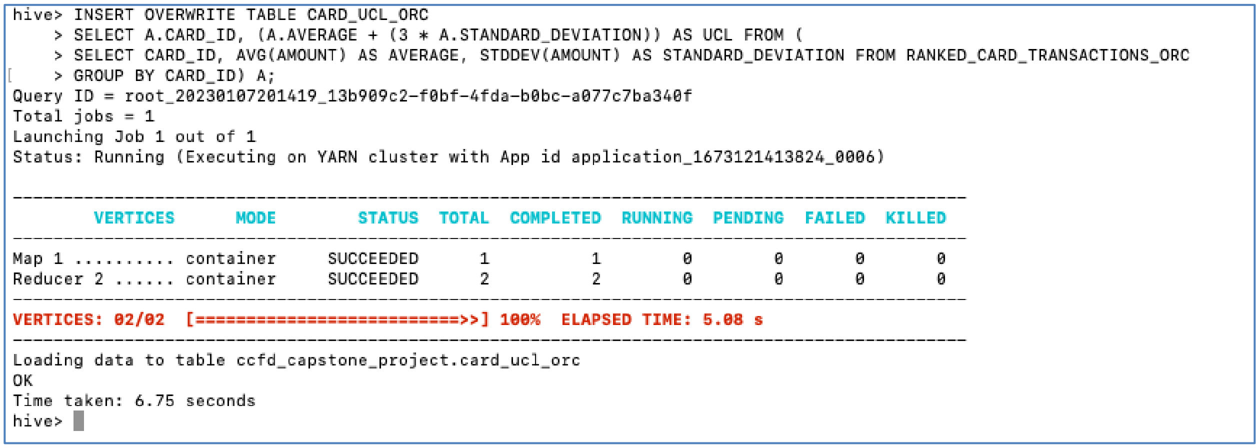
INSERT OVERWRITE TABLE RANKED_CARD_TRANSACTIONS_ORC
SELECT B.CARD_ID,
B.AMOUNT,
B.POSTCODE,
B.TRANSACTION_DT,
B.RANK
FROM (SELECT A.CARD_ID,
A.AMOUNT,
A.POSTCODE,
A.TRANSACTION_DT,
RANK() OVER(PARTITION BY A.CARD_ID ORDER BY A.TRANSACTION_DT DESC, AMOUNT DESC) AS RANK
FROM (SELECT CARD_ID, AMOUNT, POSTCODE, 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_20230107201229_6967cb34-c7ea-49ba-bc90-2fc81d7d80d5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1673121413824_0006)

----------------------------------------------------------------------------------------------
        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: 9.30 s
----------------------------------------------------------------------------------------------
Loading data to table ccfd_capstone_project.ranked_card_transactions_orc
OK
Time taken: 14.933 seconds
```

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.

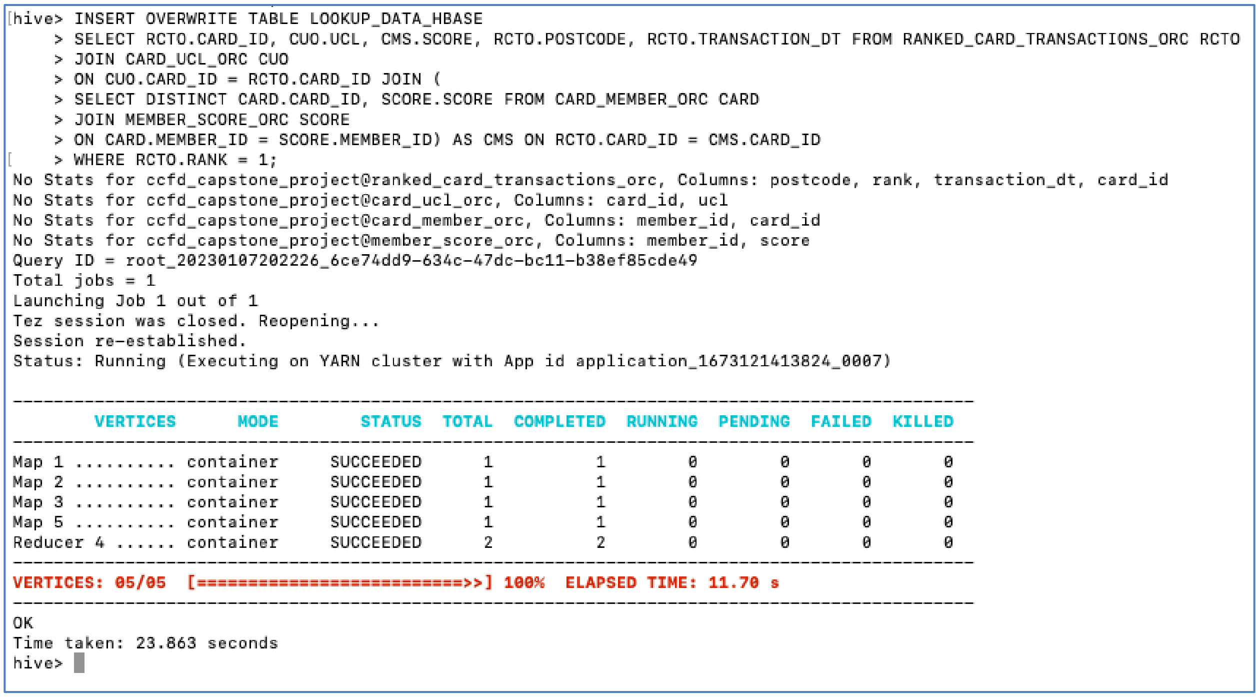
```
INSERT OVERWRITE TABLE CARD_UCL_ORC
SELECT A.CARD_ID,
(A.AVERAGE + (3 * A.STANDARD_DEVIATION)) AS UCL FROM ( SELECT CARD_ID,
AVG(AMOUNT) AS AVERAGE, STDDEV(AMOUNT) AS STANDARD_DEVIATION
FROM RANKED_CARD_TRANSACTIONS_ORC
GROUP BY CARD_ID) A;
```



```
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_20230107201419_13b909c2-f0bf-4fda-b0bc-a077c7ba340f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1673121413824_0006)

----------------------------------------------------------------------------------------------
        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: 5.08 s
----------------------------------------------------------------------------------------------
Loading data to table ccfd_capstone_project.card_ucl_orc
OK
Time taken: 6.75 seconds
hive>
```

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

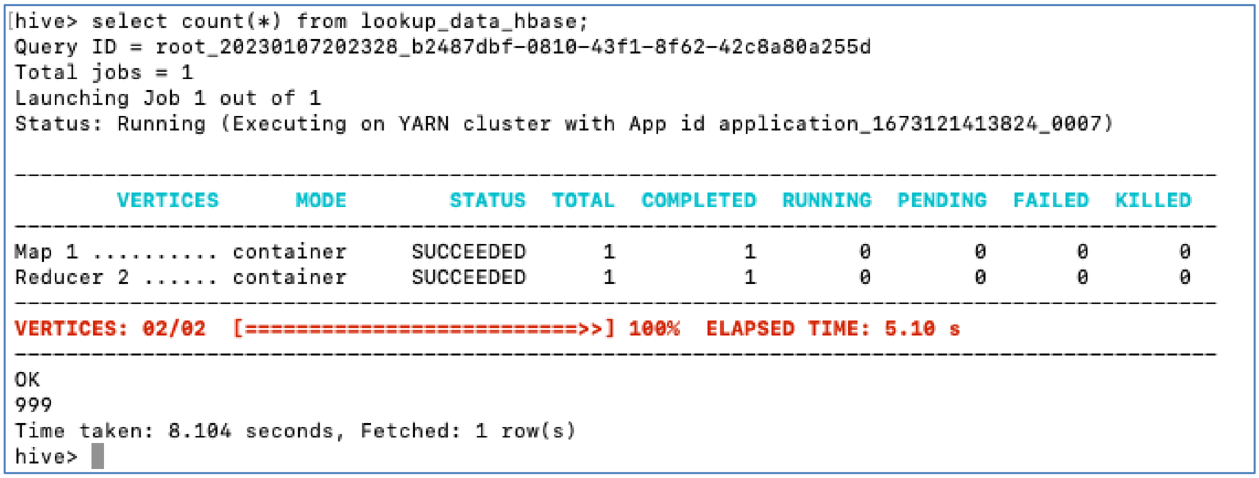
   INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE SELECT RCTO.CARD_ID,
   CUO.UCL,
   CMS.SCORE,
   RCTO.POSTCODE,
   RCTO.TRANSACTION_DTFROM RANKED_CARD_TRANSACTIONS_ORC
   RCTO JOIN CARD_UCL_ORC CUO ON CUO.CARD_ID = RCTO.CARD_IDJOIN (
   SELECT DISTINCT CARD.CARD_ID,
   SCORE.SCOREFROM CARD_MEMBER_ORC CARD
   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 SCORE
    > ON CARD.MEMBER_ID = SCORE.MEMBER_ID) AS CMS ON RCTO.CARD_ID = CMS.CARD_ID
    > WHERE RCTO.RANK = 1;
No Stats for ccfd_capstone_project@ranked_card_transactions_orc, Columns: postcode, rank, transaction_dt, card_id
No Stats for ccfd_capstone_project@card_ucl_orc, Columns: card_id, ucl
No Stats for ccfd_capstone_project@card_member_orc, Columns: member_id, card_id
No Stats for ccfd_capstone_project@member_score_orc, Columns: member_id, score
Query ID = root_20230107202226_6ce74dd9-634c-47dc-bc11-b38ef85cde49
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_1673121413824_0007)

----------------------------------------------------------------------------------------------
        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: 11.70 s
----------------------------------------------------------------------------------------------
OK
Time taken: 23.863 seconds
hive>
```

6. Verify count in “**lookup_data_hbase**” table.

   select count(*) from lookup_data_hbase;

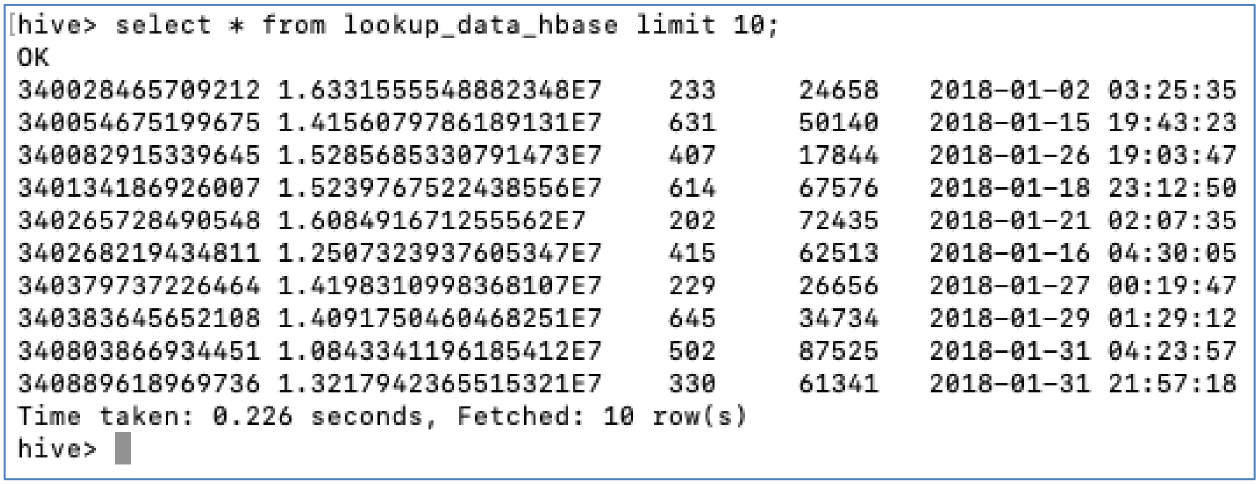


```
hive> select count(*) from lookup_data_hbase;
Query ID = root_20230107202328_b2487dbf-0810-43f1-8f62-42c8a80a255d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1673121413824_0007)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.10 s
----------------------------------------------------------------------------------------------
OK
999
Time taken: 8.104 seconds, Fetched: 1 row(s)
hive>
```

Total number for record is **999** which is matching with given requirement.

7. Verify some data in “**lookup_data_hbase**” table.

   select * from lookup_data_hbase limit 10;

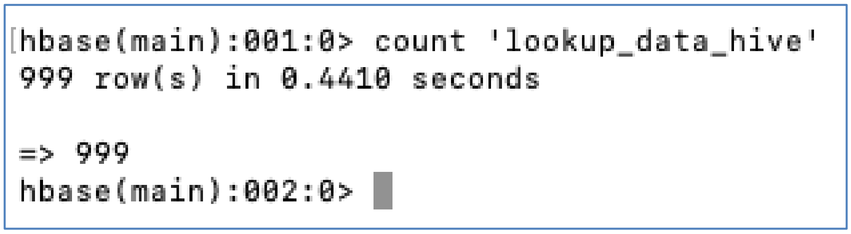


```
hive> select * from lookup_data_hbase limit 10;
OK
340028465709212 1.6331555548882348E7    233     24658   2018-01-02 03:25:35
340054675199675 1.4156079786189131E7    631     50140   2018-01-15 19:43:23
340082915339645 1.5285685330791473E7    407     17844   2018-01-26 19:03:47
340134186926007 1.5239767522438556E7    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: 0.226 seconds, Fetched: 10 row(s)
hive>
```

--Hive Operations: Ends Here --

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

1. Start HBase shell and verify count in “**lookup_data_hive**” table. count 'lookup_data_hive'

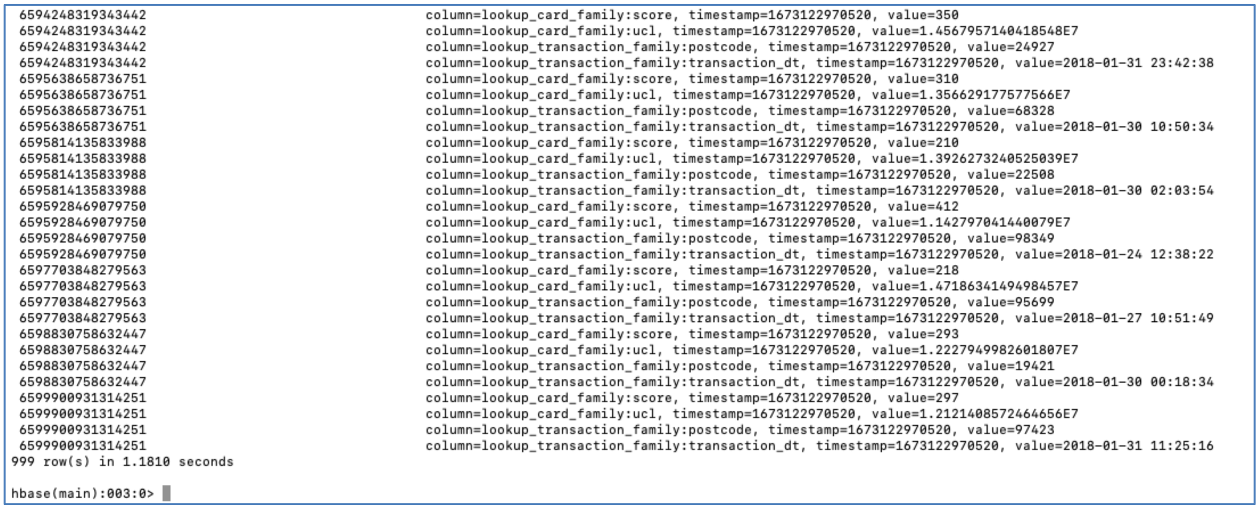


```
[hbase(main):001:0> count 'lookup_data_hive'
999 row(s) in 0.4410 seconds

=> 999
hbase(main):002:0> 
```

Total number for record is **999** which is matching with given requirement.

2. Verify data in “**lookup_data_hive**” table.

   scan 'lookup_data_hive'



```
 6594248319343442                                   column=lookup_card_family:score, timestamp=1673122970520, value=350
 6594248319343442                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.4567957140418548E7
 6594248319343442                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=24927
 6594248319343442                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-31 23:42:38
 6595638658736751                                   column=lookup_card_family:score, timestamp=1673122970520, value=310
 6595638658736751                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.356629177577566E7
 6595638658736751                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=68328
 6595638658736751                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-30 10:50:34
 6595814135833988                                   column=lookup_card_family:score, timestamp=1673122970520, value=210
 6595814135833988                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.3926273240525039E7
 6595814135833988                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=22508
 6595814135833988                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-30 02:03:54
 6595928469079750                                   column=lookup_card_family:score, timestamp=1673122970520, value=412
 6595928469079750                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.142797041440079E7
 6595928469079750                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=98349
 6595928469079750                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-24 12:38:22
 6597703848279563                                   column=lookup_card_family:score, timestamp=1673122970520, value=218
 6597703848279563                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.4718634149498457E7
 6597703848279563                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=95699
 6597703848279563                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-27 10:51:49
 6598830758632447                                   column=lookup_card_family:score, timestamp=1673122970520, value=293
 6598830758632447                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.2227949982601807E7
 6598830758632447                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=19421
 6598830758632447                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-30 00:18:34
 6599900931314251                                   column=lookup_card_family:score, timestamp=1673122970520, value=297
 6599900931314251                                   column=lookup_card_family:ucl, timestamp=1673122970520, value=1.2121408572464656E7
 6599900931314251                                   column=lookup_transaction_family:postcode, timestamp=1673122970520, value=97423
 6599900931314251                                   column=lookup_transaction_family:transaction_dt, timestamp=1673122970520, value=2018-01-31 11:25:16
999 row(s) in 1.1810 seconds

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

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