

## Scripts Execution

### Explanation of the solution to the batch layer problem

- For the below mentioned tasks, an EMR cluster with **Hadoop, Sqoop, Hive, HBase and Spark**, Root device EBS volume size as 20 GB was created.
  - Task 1:** Loading the historical transaction data (card\_transactions.csv) in a NoSQL database.
  - Task 2:** Ingesting the required data from AWS RDS to Hadoop.
  - Task 3:** Creating a look-up table with the required columns as per the problem statement.
  - Task 4:** After creation of the table, loading of the relevant data into the lookup table.

2. Logging into EMR instance as "ec2-user" through Putty - Gen

3. Switching to root user and then to HDFS user.

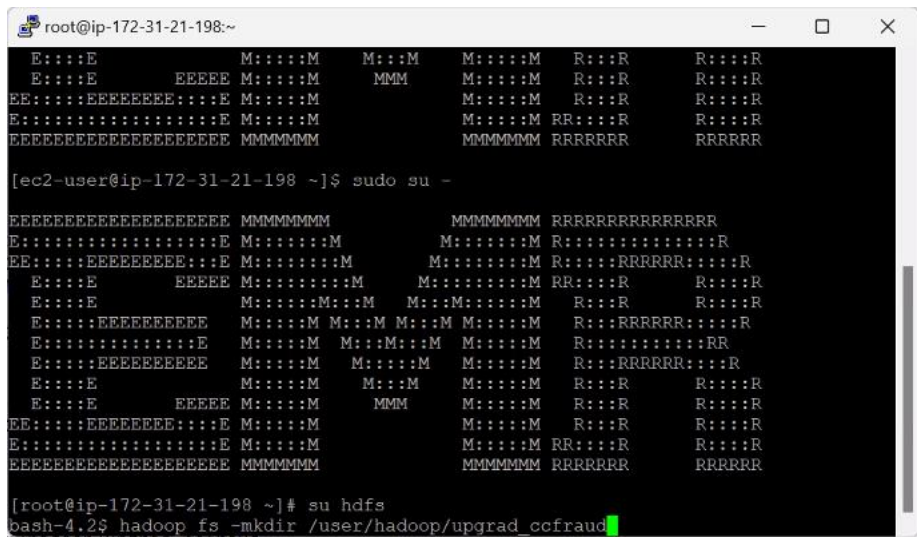
- Create directory and change its ownership -> First, exit from HDFS user -> Exit from root user back to ec2-user.

**sudo su -**

**su - hdfs**

**hadoop fs -mkdir /user/hadoop/upgrad\_ccfraud**

**hadoop fs -chown ec2-user:ec2-user /user/hadoop/upgrad\_ccfraud**



```

root@ip-172-31-21-198:~$ sudo su -
[ec2-user@ip-172-31-21-198 ~]$ sudo su -
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRR
E:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
EE:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
E:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRR RRRRRR

[ec2-user@ip-172-31-21-198 ~]$ sudo su -
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRR
E:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
EE:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
E:EEEEEEEEEEEEEEEEEEEE M:MMMMM M:MMMMM M:MMMMM R:RRRR R:RRRR
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRR RRRRRR

[ec2-user@ip-172-31-21-198 ~]$ su hdfs
bash-4.2$ hadoop fs -mkdir /user/hadoop/upgrad_ccfraud
  
```

4. Uploading the historical transactions file, card\_transactions.csv file to ec2 instance.

5. Creating a directory in HDFS and copy card\_transactions.csv to that location.

**hadoop fs -mkdir /user/hadoop/upgrad\_ccfraud/card\_transactions**

**hadoop fs -put card\_transactions.csv /user/hadoop/upgrad\_ccfraud/card\_transactions/**

The initial steps for setting up the project is complete.

## Task 1 :Loading Historical Transactions Data into NoSQL Database

1. Starting hive session and creating a new database named ccfraud\_capstone -> switch to ccfraud\_capstone database

```
create database ccfraud_capstone;  
use ccfraud_capstone;
```

```
hive> create database ccfraud_capstone;  
OK  
Time taken: 0.481 seconds  
hive> use ccfraud_capstone;  
OK  
Time taken: 0.083 seconds  
hive>
```

2. Setting 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;  
hive>
```

3. Creating 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 '/user/hadoop/upgrad_ccfraud/card_transactions/' TBLPROPERTIES  
  ("skip.header.line.count"="1");
```

```

root@ip-172-31-21-198:~
at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:1079)
at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolServerSideTranslatorPB.mkdirs(ClientNameNodeProtocolServerSideTranslatorPB.java:652)
at org.apache.hadoop.hdfs.protocol.proto.ClientNameNodeProtocolProtos$ClientNameNodeProtocol$2.callBlockingMethod(ClientNameNodeProtocolProtos.java)
at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844)
at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489)
)
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 '/user/hadoop/upgrad_ccfraud/card_transactions/' TBLPROPERTIES
> ("skip.header.line.count"="1");
OK
Time taken: 0.112 seconds
hive>

```

4. Creating table "CC\_TRANSACTIONS\_ORC" in ORC format for better performance.

**CREATE TABLE IF NOT EXISTS CC\_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");**

```

root@ip-172-31-21-198:~
at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolProtos.java)
at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844)
at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489)
)
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 '/user/hadoop/upgrad_ccfraud/card_transactions/' TBLPROPERTIES
> ("skip.header.line.count"="1");
OK
Time taken: 0.112 seconds
hive> CREATE TABLE IF NOT EXISTS CC_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: 0.386 seconds
hive>

```

5. Loading the data in "CC\_TRANSACTIONS\_ORC" table and type casting transaction\_dt column to timestamp format

**INSERT OVERWRITE TABLE CC\_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;

```

root@ip-172-31-21-198:~
Time taken: 0.112 seconds
hive> CREATE TABLE IF NOT EXISTS CC_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: 0.386 seconds
hive> INSERT OVERWRITE TABLE CC_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_20221219173434_18524e95-22bf-4a3d-9bd8-b57c0d0d6895
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_1671469134332_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: 7.79 s
-----
Loading data to table ccfraud_capstone.cc_transactions_orc
OK
Time taken: 18.515 seconds
hive>
>

```

6. Verifying **transaction\_dt** and **year** columns in " **CC\_TRANSACTIONS\_ORC** " table.

**select year(transaction\_dt), transaction\_dt from cc\_transactions\_orc limit 10;**

```

root@ip-172-31-21-198:~
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1671469134332_0002)

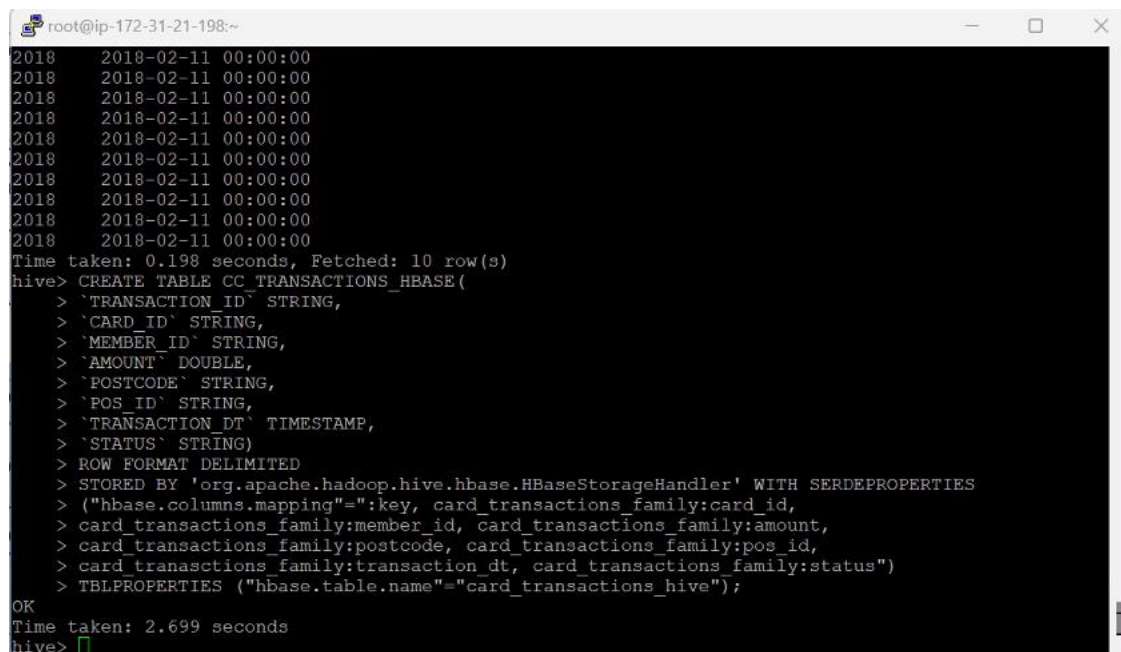
-----
VERTICES    MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container    SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 01/01  [=====>>>] 100%  ELAPSED TIME: 7.79 s
-----
Loading data to table ccfraud_capstone.cc_transactions_orc
OK
Time taken: 18.515 seconds
hive>
> select year(transaction_dt), transaction_dt from cc_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: 0.198 seconds, Fetched: 10 row(s)
hive>

```

7. Creating an integrated hive - hbase table that will be visible in HBase as well. " **CC\_TRANSACTIONS\_HBASE** " table



```
CREATE TABLE CC_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"="cc_transactions_hive");
```



```
root@ip-172-31-21-198:~
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: 0.198 seconds, Fetched: 10 row(s)
hive> CREATE TABLE CC_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: 2.699 seconds
hive>
```

8. Loading data in "CC\_TRANSACTIONS\_HBASE" table that will be visible in HBase as well with table name as "cc\_transactions\_hive". Using randomUUID to populate TRANSACTION\_ID field (row key).

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

```

root@ip-172-31-21-198:~
at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:348)
at org.apache.hadoop.hive.common.JavaUtils.loadClass(JavaUtils.java:59)
at org.apache.hadoop.hive.common.JavaUtils.loadClass(JavaUtils.java:55)
at org.apache.hadoop.hive.ql.udf.generic.GenericUDFReflect.evaluate(GenericUDFReflect.java:
106)
... 26 more
]], Vertex did not succeed due to OWN_TASK_FAILURE, failedTasks:1 killedTasks:0, Vertex vertex_1671
469134332_0002_3_00 [Map 1] killed/failed due to:OWN_TASK_FAILURE]DAG did not succeed due to VERTEX
_FAILURE. failedVertices:1 killedVertices:0
hive> INSERT OVERWRITE TABLE CC_TRANSACTIONS_HBASE SELECT
> reflect('java.util.UUID', 'randomUUID') as TRANSACTION_ID, CARD_ID, MEMBER_ID, AMOUNT,
> POSTCODE, POS_ID, TRANSACTION_DT, STATUS
> FROM CC_TRANSACTIONS_ORC;
Query ID = root_20221219174451_401af4d4-d312-454b-85dc-8cb828792aba
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_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: 11.20 s
-----
OK
Time taken: 15.654 seconds
hive>

```

9. Verify data in "cc\_transactions\_hbase" table.

**select \* from cc\_transactions\_hbase limit 10;**

```

root@ip-172-31-21-198:~
Map 1 ..... container      SUCCEEDED      1          1          0          0          0          0
-----
VERTICES: 01/01  [=====>>>] 100% ELAPSED TIME: 11.20 s
-----
OK
Time taken: 15.654 seconds
hive> select * from cc_transactions_hbase limit 10;
OK
0001cef0-4d5b-4d88-b0ce-0da380409a66 5284175522526975 413021843879928 1604268.0 140
20 214696179763291 2016-03-27 05:29:37 GENUINE
0005ec01-c98b-4d41-9d17-b2c0ff5e960 5271320386643574 991784667284668 1590198.0 926
25 225865830430921 2017-05-30 20:07:20 GENUINE
00070787-5f76-4fb6-bb79-45aed2e3a5fd 6460729612153589 266629518896098 4366995.0 975
39 893307976868097 2017-01-12 15:47:50 GENUINE
0007767e-56f7-407f-b2fe-b50306d83dfb 5175735819607028 227686059795799 2082575.0 520
54 215945504043719 2018-02-11 00:00:00 GENUINE
0007e0d3-3b77-4304-b2b6-cdd6e2333937 6223813071182434 736137659378265 1714630.0 913
56 748648404529559 2017-07-29 05:40:05 GENUINE
000916dc-4a82-4068-9062-02bb410fa5c4 5134334388575160 978465390240911 4731739.0 610
80 569335687828707 2016-05-27 03:54:25 GENUINE
000a5c44-d3f7-4650-9e03-55d0747a856b 4389973676463558 295554828848966 6305493.0 567
57 911478839996964 2017-04-17 05:47:08 GENUINE
000b85c8-16ff-4f9a-9d18-f13292fa4005 6011706374151856 199243620982420 9732433.0 538
12 478785382784557 2017-12-14 22:16:57 GENUINE
000baa11-7d7b-4682-94dc-5663885bc269 341722035429601 979218131207765 8325634.0 67741 747
682736002337 2017-02-06 05:40:42 GENUINE
000c16d5-d964-4e69-afdc-09bfd7a4cd2e 6011938409004772 577907767500023 1988126.0 388
20 814499910586436 2016-02-23 00:27:07 GENUINE
Time taken: 0.288 seconds, Fetched: 10 row(s)
hive>

```

10. Starting HBase session and verifying details of "cc\_transactions\_hive" table (hive-hbase integrated table).

**describe 'cc\_transactions\_hive'**

11. Verify count of "cc\_transactions\_hive" table

count 'cc\_transactions\_hive'

```

root@ip-172-31-21-198:~
Current count: 28000, row: 87685a0a-05c3-4a42-a279-210baadf35bc
Current count: 29000, row: 8c4e959d-c385-453f-8f62-e0d01217173e
Current count: 30000, row: 90f8731b-3235-4f70-a5cf-3db1156bce4a
Current count: 31000, row: 95a87f67-0b18-4846-8d67-3368421bced6
Current count: 32000, row: 9a3a394f-d232-4d29-b298-52b330bab082
Current count: 33000, row: 9f16276a-f6ad-4e22-9850-e71f1dc869de
Current count: 34000, row: a3d39690-2382-41b4-a306-f1f9a429c475
Current count: 35000, row: a89aad1e-d0d8-4909-a035-1dec9cf09ac6
Current count: 36000, row: ad239bcb-1dc7-459a-bb26-13c2ceb33d73
Current count: 37000, row: b1e32188-98c3-4a1c-ae5c-eff3e121327b
Current count: 38000, row: b6956bff-e40d-4a07-a245-09887745840f
Current count: 39000, row: bb3c6641-8dac-4063-8c6c-7ff3c75f98c5
Current count: 40000, row: c01543a6-6f8c-42ed-a62d-9df2865e9fd6
Current count: 41000, row: c4be3333-b3bb-422f-a371-085b99f74912
Current count: 42000, row: c99b7318-3baa-421e-a976-4613d3452172
Current count: 43000, row: ce61b3f8-9951-4c27-9163-a401fb6875b4
Current count: 44000, row: d2f91fb9-e32d-44b9-8024-eeee031d71ad
Current count: 45000, row: d8066d3d-527d-4f3f-b688-4ee6f40d7155
Current count: 46000, row: dce52057-071e-4ad7-bab3-5bef2f0d247f
Current count: 47000, row: elae4d3d-bd9c-41c3-9d27-e024a2545a04
Current count: 48000, row: e6ab685e-7154-4335-99c6-84d021f89795
Current count: 49000, row: eb793a0e-a874-4908-af1a-0bf507c12c91
Current count: 50000, row: f059e689-6446-4c34-97e8-2949fe13bec3
Current count: 51000, row: f50c5b6d-e444-4214-9780-c24131b08f51
Current count: 52000, row: f9c8db4c-6bc0-4a24-8734-1a4edc19b3dd
Current count: 53000, row: fe93bb44-8892-4d57-b9af-36050f0a83a3
53292 row(s) in 3.8990 seconds
=> 53292
hbase(main):005:0>

```

Count of the "cc\_transactions\_hive" table is 53,292 which is matching with given requirement

## Task 2: Ingesting the required data from AWS RDS to Hadoop.

1. Running Sqoop command to import "member\_score" table from RDS to HDFS.

```

sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielec9bmnf.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 '/user/hadoop/upgrad_ccfraud/card_transactions/member_score' \
-m 1

```

```
ec2-user@ip-172-31-21-198:~$
22/12/19 18:06:19 INFO mapreduce.Job: map 100% reduce 0%
22/12/19 18:06:19 INFO mapreduce.Job: Job job_1671469134332_0005 completed successfully
22/12/19 18:06:19 INFO mapreduce.Job: Counters: 30
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=189927
    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)=217824
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=4538
    Total vcore-milliseconds taken by all map tasks=4538
    Total megabyte-milliseconds taken by all map tasks=6970368
  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)=91
    CPU time spent (ms)=2510
    Physical memory (bytes) snapshot=277565440
    Virtual memory (bytes) snapshot=3282108416
    Total committed heap usage (bytes)=250609664
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=19980
22/12/19 18:06:19 INFO mapreduce.ImportJobBase: Transferred 19.5117 KB in 19.2842 seconds (1.0118 KB/sec)
22/12/19 18:06:19 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[ec2-user@ip-172-31-21-198 ~]$
```

2. Running Sqoop command to import “card\_member” table from RDS to HDFS.

```
sqoop import --connect jdbc:mysql://upgradawsrds1.cyaieic9bmnf.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 '/user/hadoop/upgrad_ccfraud/card_transactions/card_member' \
-m 1
```



```
ec2-user@ip-172-31-21-198:~
22/12/19 18:13:22 INFO mapreduce.Job: Job job_1671469134332_0006 completed successfully
22/12/19 18:13:22 INFO mapreduce.Job: Counters: 30
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=189979
    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)=216960
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=4520
    Total vcore-milliseconds taken by all map tasks=4520
    Total megabyte-milliseconds taken by all map tasks=6942720
  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)=85
    CPU time spent (ms)=3070
    Physical memory (bytes) snapshot=284651520
    Virtual memory (bytes) snapshot=3302727680
    Total committed heap usage (bytes)=246415360
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=85081
22/12/19 18:13:22 INFO mapreduce.ImportJobBase: Transferred 83.0869 KB in 19.3302 seconds (4.2
983 KB/sec)
22/12/19 18:13:22 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[ec2-user@ip-172-31-21-198 ~]$
```

- Starting hive session and creating an 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
'/user/hadoop/upgrad_ccfraud/card_transactions/card_member';
```

```

root@ip-172-31-21-198:/home/ec2-user
ion(message:Got exception: org.apache.hadoop.security.AccessControlException Permission denied
: user=root, access=WRITE, inode="/user/hadoop/upgrad_ccfraud/card_transactions":ec2-user:ec2-
user:drwxr-xr-x
    at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecke
r.java:319)
    at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermis
sionChecker.java:219)
    at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermis
sionChecker.java:189)
    at org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java
:1663)
    at org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java
:1647)
    at org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkAncestorAccess(FSDirectory.
java:1606)
    at org.apache.hadoop.hdfs.server.namenode.FSDirMkdirOp.mkdirs(FSDirMkdirOp.java:60)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3049)
    at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.j
ava:1079)
    at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolServerSideTranslatorPB.mkdi
rs(ClientNameNodeProtocolServerSideTranslatorPB.java:652)
    at org.apache.hadoop.hdfs.protocol.proto.ClientNameNodeProtocolProtos$ClientNameNodePr
otocol$2.callBlockingMethod(ClientNameNodeProtocolProtos.java)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcE
ngine.java:447)
    at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844
)
    at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489)
)
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 '/user/hadoop/upgrad_ccfraud/card_transactions/card_member';
OK
Time taken: 0.088 seconds
hive>

```

4. Create external table "CC\_MEMBER\_SCORE\_EXT" to hold data from member\_score table in RDS.

```

CREATE EXTERNAL TABLE IF NOT EXISTS CC_MEMBER_SCORE_EXT(
  'MEMBER_ID' STRING,
  'SCORE' INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/upgrad_ccfraud/card_transactions/member_score';

```

```

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 '/user/hadoop/upgrad_ccfraud/card_transacti
OK
Time taken: 0.088 seconds
hive> CREATE EXTERNAL TABLE IF NOT EXISTS CC_MEMBER_SCORE_EXT(
> 'MEMBER_ID' STRING,
> 'SCORE' INT)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
> LOCATION '/user/hadoop/upgrad_ccfraud/card_transactions/member_score';
OK
Time taken: 0.066 seconds
hive>

```

5. Create " CC\_MEMBER\_ORC" table with ORC for better performance.

```

CREATE TABLE IF NOT EXISTS CC_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");**

```

root@ip-172-31-21-198:/home/ec2-user
ava:1079)
    at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdi
rs(ClientNamenodeProtocolServerSideTranslatorPB.java:652)
    at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodePr
otocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcE
ngine.java:447)
    at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844
)
    at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489)
)
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 '/user/hadoop/upgrad_ccfraud/card_transacti
OK
Time taken: 0.088 seconds
hive> CREATE EXTERNAL TABLE IF NOT EXISTS CC_MEMBER_SCORE_EXT(
> `MEMBER_ID` STRING,
> `SCORE` INT)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
> LOCATION '/user/hadoop/upgrad_ccfraud/card_transactions/member_score';
OK
Time taken: 0.066 seconds
hive> CREATE TABLE IF NOT EXISTS CC_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: 0.401 seconds
hive>

```

6. Create " **CC\_MEMBER\_SCORE\_ORC** " table with ORC for better performance.

**CREATE TABLE IF NOT EXISTS CC\_MEMBER\_SCORE\_ORC(**  
**`MEMBER\_ID` STRING,**  
**`SCORE` INT) STORED AS ORC**  
**TBLPROPERTIES ("orc.compress"="SNAPPY");**

```

hive> CREATE TABLE IF NOT EXISTS CC_MEMBER_SCORE_ORC(
> `MEMBER_ID` STRING,
> `SCORE` INT) STORED AS ORC
> TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.074 seconds
hive>

```

7. Load data into " **CC\_MEMBER\_ORC** " table from "card\_member\_ext" table.

**INSERT OVERWRITE TABLE CC\_MEMBER\_ORC**  
**SELECT CARD\_ID, MEMBER\_ID, MEMBER\_JOINING\_DT, CARD\_PURCHASE\_DT, COUNTRY, CITY**  
**FROM CARD\_MEMBER\_EXT;**

```

root@ip-172-31-21-198:/home/ec2-user
at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:233)
at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:184)
at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:403)
at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.util.RunJar.run(RunJar.java:239)
at org.apache.hadoop.util.RunJar.main(RunJar.java:153)
FAILED: ParseException line 4:16 cannot recognize input near 'orc' '.' 'compress' in table properties list
hive> CREATE TABLE IF NOT EXISTS CC_MEMBER_SCORE_ORC(
> `MEMBER_ID` STRING,
> `SCORE` INT) STORED AS ORC
> TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.074 seconds
hive> INSERT OVERWRITE TABLE CC_MEMBER_ORC
> SELECT CARD_ID, MEMBER_ID, MEMBER_JOINING_DT, CARD_PURCHASE_DT, COUNTRY, CITY
> FROM CARD_MEMBER_EXT;
Query ID = root_20221219182115_bd0360ee-25da-4755-8516-9a53fd841d89
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_1671469134332_0008)

-----
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: 4.69 s
-----
Loading data to table default.cc_member_orc
OK
Time taken: 14.35 seconds
hive>

```

8. Load data into “CC\_MEMBER\_SCORE\_ORC” table from “CC\_MEMBER\_SCORE\_EXT” table.

```

INSERT OVERWRITE TABLE CC_MEMBER_SCORE_ORC
SELECT MEMBER_ID, SCORE FROM CC_MEMBER_SCORE_EXT;

```



```

root@ip-172-31-21-198:/home/ec2-user
> TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.074 seconds
hive> INSERT OVERWRITE TABLE CC_MEMBER_ORC
> SELECT CARD_ID, MEMBER_ID, MEMBER_JOINING_DT, CARD_PURCHASE_DT, COUNTRY, CITY
> FROM CARD_MEMBER_EXT;
Query ID = root_20221219182115_bd8360ee-25da-4755-8516-9a53fd841d89
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_1671469134332_0008)

-----
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: 4.69 s
-----
Loading data to table default.cc_member_orc
OK
Time taken: 14.35 seconds
hive> INSERT OVERWRITE TABLE CC_MEMBER_SCORE_ORC
> SELECT MEMBER_ID, SCORE FROM CC_MEMBER_SCORE_EXT;
Query ID = root_20221219182207_f5f354e8-a68a-4005-898e-d9e073ca56e3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0008)

-----
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: 4.82 s
-----
Loading data to table default.cc_member_score_orc
OK
Time taken: 5.872 seconds
hive>

```

9. Verify data in "CC\_MEMBER\_ORC " table.

**SELECT \* FROM CC\_MEMBER\_ORC LIMIT 10;**

```

VERTICES: 01/01 [=====>>>] 100% ELAPSED TIME: 4.82 s
-----
Loading data to table default.cc_member_score_orc
OK
Time taken: 5.872 seconds
hive> SELECT * FROM CC_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: 0.198 seconds, Fetched: 10 row(s)
hive>

```

10. Verify data in "CC\_MEMBER\_SCORE\_ORC" table.

**SELECT \* FROM CC\_MEMBER\_SCORE\_ORC LIMIT 10;**

```

root@ip-172-31-21-198:/home/ec2-user
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0008)

-----
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: 4.82 s
-----
Loading data to table default.cc_member_score_orc
OK
Time taken: 5.872 seconds
hive> SELECT * FROM CC_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: 0.198 seconds, Fetched: 10 row(s)
hive> SELECT * FROM CC_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: 0.117 seconds, Fetched: 10 row(s)
hive>

```

### Task 3: Creating a look-up table with the required columns as per the problem statement.

1. Creating "CC\_lookup\_data\_hbase" table (hive-hbase integrated table) which will be visible in HBase( lookup\_data\_hive).

```

CREATE TABLE CC_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_transactions_family:postcode,
lookup_transactions_family:transactions_dt") TBLPROPERTIES
("hbase.table.name" = "cc_lookup_data_hive");

```

```

hive> CREATE TABLE CC_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_transactions_family:postcode, lookup_transactions_family:transactions_dt") TBLPROPERTIES
> ("hbase.table.name" = "cc_lookup_data_hive");
OK
Time taken: 1.65 seconds
hive>

```

2. Verifying details of **cc\_lookup\_data\_hive** (hive-hbase integrated) table:  
**describe 'cc\_lookup\_data\_hive'**

```
hive> CREATE TABLE CC_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_transactions_family:postcode, lookup_transactions_family:transactions_dt") TBLPROPERTIES
> ("hbase.table.name" = "cc_lookup_data_hive");
OK
Time taken: 1.65 seconds
hive> exit;
[root@ip-172-31-21-198 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 'cc_lookup_data_hive'
Table cc_lookup_data_hive is ENABLED
cc_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_transactions_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP
_DELETED CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_V
ERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
2 row(s) in 0.3090 seconds

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

3. Altering “cc\_lookup\_data\_hive” table and setting VERSIONS to 10 for lookup\_transaction\_family since as per the problem statement we have to store last 10 transactions in lookup table.

**alter 'cc\_lookup\_data\_hive', {NAME => 'lookup\_transactions\_family', VERSIONS => 10}**

```
hbase(main):001:0> describe 'cc_lookup_data_hive'
Table cc_lookup_data_hive is ENABLED
cc_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_transactions_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP
_DELETED CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_V
ERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
2 row(s) in 0.3090 seconds

hbase(main):002:0> alter 'cc_lookup_data_hive', {NAME => 'lookup_transactions_family', VERSIONS => 10}
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 2.4090 seconds

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

4. Verify details of “cc\_lookup\_data\_hive” (hive-hbase integrated) table after altering version to 10 :  
**describe 'cc\_lookup\_data\_hive'**

```
[root@ip-172-31-21-198 ec2-user]# describe 'cc_lookup_data_hive'
bash: describe: command not found
[root@ip-172-31-21-198 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 'cc_lookup_data_hive'
Table cc_lookup_data_hive is ENABLED
cc_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_transactions_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 0.2970 seconds

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

**Task 4: After creation of the table, loading of the relevant data into the lookup table.**

1. Starting hive session and creating table “**ranked\_cc\_transactions\_orc**” to store last 10 transactions for eachcard\_id. for better performance.

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

```
hbase(main):001:0> describe 'cc_lookup_data_hive'
Table cc_lookup_data_hive is ENABLED
cc_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_transactions_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 0.2970 seconds

hbase(main):002:0> exit;
hbase(main):003:0* quit;
hbase(main):004:0* exit
[root@ip-172-31-21-198 ec2-user]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE TABLE IF NOT EXISTS RANKED_CC_TRANSACTIONS_ORC(
> `CARD_ID` STRING,
> `AMOUNT` DOUBLE,
> `POSTCODE` STRING,
> `TRANSACTION_DT` TIMESTAMP,
> `RANK` INT) STORED AS ORC
> TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 1.495 seconds
hive>
```

2. Create table "**cc\_ucl\_orc**" to store UCL values for each card\_id. For better performance.

```
CREATE TABLE IF NOT EXISTS CC_UCL_ORC(
`CARD_ID` STRING,
```



```
`UCL` DOUBLE) STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");
```

```
root@ip-172-31-21-198:/home/ec2-user
bash: describe: command not found
[root@ip-172-31-21-198 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 'cc_lookup_data_hive'
Table cc_lookup_data_hive is ENABLED
cc_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_transactions_family', BLOOMFILTER => 'ROW', VERSIONS => '10', IN_MEMORY => 'false', KEE
P_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 0.2970 seconds

hbase(main):002:0> exit;
hbase(main):003:0* quit;
hbase(main):004:0* exit
[root@ip-172-31-21-198 ec2-user]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE TABLE IF NOT EXISTS RANKED_CC_TRANSACTIONS_ORC(
  > `CARD_ID` STRING,
  > `AMOUNT` DOUBLE,
  > `POSTCODE` STRING,
  > `TRANSACTION_DT` TIMESTAMP,
  > `RANK` INT) STORED AS ORC
  > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 1.495 seconds
hive> CREATE TABLE IF NOT EXISTS CC_UCL_ORC(
  > `CARD_ID` STRING,
  > `UCL` DOUBLE) STORED AS ORC
  > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.06 seconds
hive>
```

3. Loading data in “ranked\_cc\_transactions\_orc” table

```
INSERT OVERWRITE TABLE RANKED_CC_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 cc_transactions_HBASE
WHERE STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;
```

```

root@ip-172-31-21-198:/home/ec2-user
-06-19 17:38:42 GENUINE
0016a375-91c1-4315-b8d0-c77b907573cf 345027679564944 808853092206668 9365301.0 72179 307736895557139 2017-07-20 0
0:27:46 GENUINE
Time taken: 0.244 seconds, Fetched: 10 row(s)
hive> show tables;
OK
card_member_ext
card_transactions_ext
cc_lookup_data_hbase
cc_member_orc
cc_member_score_ext
cc_member_score_orc
cc_transactions_hbase
cc_transactions_orc
cc_ucl_orc
lookup_data_hbase
ranked_cc_transactions_orc
Time taken: 0.02 seconds, Fetched: 11 row(s)
hive> INSERT OVERWRITE TABLE RANKED_CC_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 cc_transactions_HBASE
> WHERE STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;
Query ID = root_20221219191611_8c9d209f-b01b-4326-b59a-b5bd4b010fe7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)

-----
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: 12.98 s
-----
Loading data to table default.ranked_cc_transactions_orc
OK
Time taken: 17.481 seconds
hive>

```

4. Loading data in "cc\_ucl\_orc" table.  
In the 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  $(avg + (3 * stddev))$ . Inserting all this data in card\_ucl\_orc.

```

INSERT OVERWRITE TABLE CC_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_CC_TRANSACTIONS_ORC
GROUP BY CARD_ID) A;

```

```

root@ip-172-31-21-198:/home/ec2-user
hive> INSERT OVERWRITE TABLE RANKED_CC_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 cc_transactions_HBASE
> WHERE STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;
Query ID = root_20221219191611_8c9d209f-b01b-4326-b59a-b5bd4b010fe7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)

-----
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: 12.98 s
-----
Loading data to table default.ranked_cc_transactions_orc
OK
Time taken: 17.481 seconds
hive> INSERT OVERWRITE TABLE CC_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_CC_TRANSACTIONS_ORC
> GROUP BY CARD_ID) A;
Query ID = root_20221219192128_b1e7ab05-a2f9-450c-8085-1a8237a7ea35
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  RUNNING     1         0         1         0         0         0
Reducer 2 ..... container  INITED      2         0         0         2         0         0
-----
VERTICES: 00/02 [>-----] 0% ELAPSED TIME: 4.07 s
-----

```

5. Load data in **cc\_lookup\_data\_hbase** table.

```

INSERT OVERWRITE TABLE CC_LOOKUP_DATA_HBASE
SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE, RCTO.TRANSACTION_DT FROM
RANKED_CC_TRANSACTIONS_ORC RCTO
JOIN CC_UCL_ORC CUO
ON CUO.CARD_ID = RCTO.CARD_ID JOIN (
SELECT DISTINCT CARD.CARD_ID, SCORE.SCORE FROM CC_MEMBER_ORC CARD
JOIN CC_MEMBER_SCORE_ORC SCORE
ON CARD.MEMBER_ID = SCORE.MEMBER_ID) AS CMS ON RCTO.CARD_ID = CMS.CARD_ID
WHERE RCTO.RANK = 1;

```

```

root@ip-172-31-21-198:/home/ec2-user
-----
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: 6.64 s
-----
Loading data to table default.cc_ucl_orc
OK
Time taken: 7.524 seconds
hive> INSERT OVERWRITE TABLE CC_LOOKUP_DATA_HBASE
> SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE, RCTO.TRANSACTION_DT FROM
> RANKED_CC_TRANSACTIONS_ORC RCTO
> JOIN CC_UCL_ORC CUO
> ON CUO.CARD_ID = RCTO.CARD_ID JOIN (
> SELECT DISTINCT CARD.CARD_ID, SCORE.SCORE FROM CC_MEMBER_ORC CARD
> JOIN CC_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 default@ranked_cc_transactions_orc, Columns: postcode, rank, transaction_dt, card_id
No Stats for default@cc_ucl_orc, Columns: card_id, ucl
No Stats for default@cc_member_orc, Columns: member_id, card_id
No Stats for default@cc_member_score_orc, Columns: member_id, score
Query ID = root_20221219192220_7cf9b5e3-100e-4874-b65f-984440d7cc83
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)
-----
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: 12.27 s
-----
OK
Time taken: 18.233 seconds
hive>

```

6. Verifying count in “cc\_lookup\_data\_hbase” table.

**select count(\*) from cc\_lookup\_data\_hbase;**

```

root@ip-172-31-21-198:/home/ec2-user
NoViableAltException (240 [])
    at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1300)
    at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:208)
    at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:77)
    at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:70)
    at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:468)
    at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1317)
    at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1457)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1237)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1227)
    at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:233)
    at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:184)
    at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:403)
    at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:239)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:153)
FAILED: ParseException line 1:8 cannot recognize input near 'VERTICES' 'exit' '<EOF>'
hive> select count(*) from cc_lookup_data_hbase;
Query ID = root_20221219192420_311bfaf17-616d-4a42-b4ae-159e9ddf720f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)
-----
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: 6.60 s
-----
OK
999
Time taken: 11.158 seconds, Fetched: 1 row(s)
hive>

```



Total number of records is **999** which is matching with given requirement.

- Verify some data in “cc\_lookup\_data\_hbase” table.

**select \* from cc\_lookup\_data\_hbase limit 10;**

```
root@ip-172-31-21-198:/home/ec2-user
at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.util.RunJar.run(RunJar.java:239)
at org.apache.hadoop.util.RunJar.main(RunJar.java:153)
FAILED: ParseException line 1:8 cannot recognize input near 'VERTICES' 'exit' '<EOF>'
hive> select count(*) from cc_lookup_data_hbase;
Query ID = root_20221219192420_311bfa17-616d-4a42-b4ae-159e9ddf720f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671469134332_0012)

-----
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: 6.60 s
-----
OK
999
Time taken: 11.158 seconds, Fetched: 1 row(s)
hive> select * from cc_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: 0.192 seconds, Fetched: 10 row(s)
hive>
```

- Start HBase shell session and verifying count in “cc\_lookup\_data\_hive” table.

**count 'cc\_lookup\_data\_hive'**

```
hbase(main):002:0> count 'cc_lookup_data_hive'
999 row(s) in 0.3560 seconds
=> 999
hbase(main):003:0>
```

Total number of records is **999** which is matching with given requirement.

- Verifying data in “cc\_lookup\_data\_hive” table.

**scan 'cc\_lookup\_data\_hive'**

```

root@ip-172-31-21-198:/home/ec2-user
6592184145413632 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=53186
6592184145413632 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-2
8 00:54:30
6594248319343442 column=lookup_card_family:score, timestamp=1671477758927, value=350
6594248319343442 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.4567957140418548E7
6594248319343442 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=24927
6594248319343442 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-3
1 23:42:38
6595638658736751 column=lookup_card_family:score, timestamp=1671477758927, value=310
6595638658736751 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.356629177577566E7
6595638658736751 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=68328
6595638658736751 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-3
0 10:50:34
6595814135833988 column=lookup_card_family:score, timestamp=1671477758927, value=210
6595814135833988 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.3926273240525039E7
6595814135833988 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=22508
6595814135833988 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-3
0 02:03:54
6595928469079750 column=lookup_card_family:score, timestamp=1671477758927, value=412
6595928469079750 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.142797041440079E7
6595928469079750 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=98349
6595928469079750 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-2
4 12:38:22
6597703848279563 column=lookup_card_family:score, timestamp=1671477758927, value=218
6597703848279563 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.4718634149498457E7
6597703848279563 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=95699
6597703848279563 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-2
7 10:51:49
6598830758632447 column=lookup_card_family:score, timestamp=1671477758927, value=293
6598830758632447 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.2227949982601807E7
6598830758632447 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=19421
6598830758632447 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-3
0 00:18:34
6599900931314251 column=lookup_card_family:score, timestamp=1671477758927, value=297
6599900931314251 column=lookup_card_family:ucl, timestamp=1671477758927, value=1.2121408572464656E7
6599900931314251 column=lookup_transactions_family:postcode, timestamp=1671477758927, value=97423
6599900931314251 column=lookup_transactions_family:transactions_dt, timestamp=1671477758927, value=2018-01-3
1 11:25:16
999 row(s) in 5.6360 seconds
hbase(main):006:0>

```