



**Capstone Project
Script Execution
(Credit Card Fraud Detection System)**

Devashish Kapadia

Content

- SQOOP-JOB-CREATION-CARD_MEMBER- DATA IMPORT FROM AWS RDS.
- SQOOP-JOB-CREATION-MEMBER_SCORE- DATA IMPORT FROM AWS RDS.
- HBASE TABLE CREATION(card_transaction)
- HBASE TABLE CREATION(lookup table)
- HBASE TABLE CREATION(card member)
- Create mapping table in hive for HBASE card_transaction table
- Card transaction staging table in hive
- Load data from local path into staging table.
- Insert card transactions which has valid card id
- Dropping staging table
- Create mapping table in hive for Hbase card_member table
- Card member staging table in hive
- Insert data into HBase mapping table
- Create member score table
- Create lookup UCL staging table
- Calculate UCL and insert into lookup UCL staging table
- Create lookup full staging table for UCL & memscore
- Insert UCL and Score into lookup full staging table
- Create mapping table for HBase lookup table for loading limit attributes
- Insert UCL and Score into mapping lookup from staging table
- Create mapping table for HBase lookup table for loading status attributes
- Insert postal code and transaction date into mapping lookup from
card_transactions_hbase
- Create mapping table for Hbase lookup table

SQOOP-JOB-CREATION-CARD_MEMBER- DATA IMPORT FROM AWS RDS.

--Create sqoop job card_member ingest from AWS RDS to HADOOP

```
sqoop job --create incremental_card_member --meta-connect "jdbc:mysql://localhost/sqoop?
user=sqoop&password=sqoop" -- import --connect jdbc:mysql://upgradawsrds.cpclxrkdvmz.us-
east-1.rds.amazonaws.com/cred_financials_data --username upgraduser --password upgraduser
--table card_member --incremental append --check-column member_joining_dt --last-value
"1970-01-01 00:00:00" --warehouse-dir /user/sqoop_import/capstone_project
```

--JOB-verification

```
sqoop job --list
```

-- Execute the job once to get the initial data load

```
sqoop job --exec incremental_card_member
```

```
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ sqoop job --create incremental_card_member --meta-conne
ct "jdbc:mysql://localhost/sqoop?user=sqoop&password=sqoop" -- import --connect
jdbc:mysql://upgradawsrds.cpclxrkdvmz.us-east-1.rds.amazonaws.com/cred_financia
ls_data --username upgraduser --password upgraduser --table card_member --incred
ental append --check-column member_joining_dt --last-value "1970-01-01 00:00:00"
--warehouse-dir /user/sqoop_import/capstone_project
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
19/11/13 09:03:09 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
19/11/13 09:03:11 WARN tool.BaseSqoopTool: Setting your password on the command-
line is insecure. Consider using -P instead.
[cloudera@quickstart ~]$ sqoop job -list
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
19/11/13 09:03:20 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Available jobs:
    incremental_card_member
[cloudera@quickstart ~]$
```

```
cloudera@quickstart:~
File Edit View Search Terminal Help
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)=19914
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=19914
    Total vcore-milliseconds taken by all map tasks=19914
    Total megabyte-milliseconds taken by all map tasks=20391936
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)=162
    CPU time spent (ms)=1470
    Physical memory (bytes) snapshot=189931520
    Virtual memory (bytes) snapshot=1583591424
    Total committed heap usage (bytes)=190840832
File Input Format Counters
    Bytes Read=0
File Output Format Counters
    Bytes Written=85082
19/11/10 06:30:10 INFO mapreduce.ImportJobBase: Transferred 83.0879 KB in 48.2861 seconds (1.7207 KB/sec)
19/11/10 06:30:10 INFO mapreduce.ImportJobBase: Retrieved 999 records.
19/11/10 06:30:10 INFO util.AppendUtils: Creating missing output directory - card_member
19/11/10 06:30:10 INFO tool.ImportTool: Saving incremental import state to the metastore
19/11/10 06:30:10 INFO tool.ImportTool: Updated data for job: incremental_card_member
[cloudera@quickstart ~]$
```

SQOOP-JOB-CREATION-MEMBER_SCORE- DATA IMPORT FROM AWS RDS.

--import for member_score table which is full load every time

```
sqoop job --create cred_member_score \  
--meta-connect "jdbc:mysql://localhost/sqoop?user=sqoop&password=sqoop" \  
-- import \  
--connect jdbc:mysql://upgradawsrds.cpclxrkdvwzmz.us-east-1.rds.amazonaws.com/cred_financials_data \  
--username upgraduser \  
--password upgraduser \  
--table member_score \  
--warehouse-dir /user/sqoop_import/capstone_project
```

--JOB-verification

```
sqoop job --list
```

-- Execute the job once to get the initial data load

```
sqoop job --exec cred_member_score
```

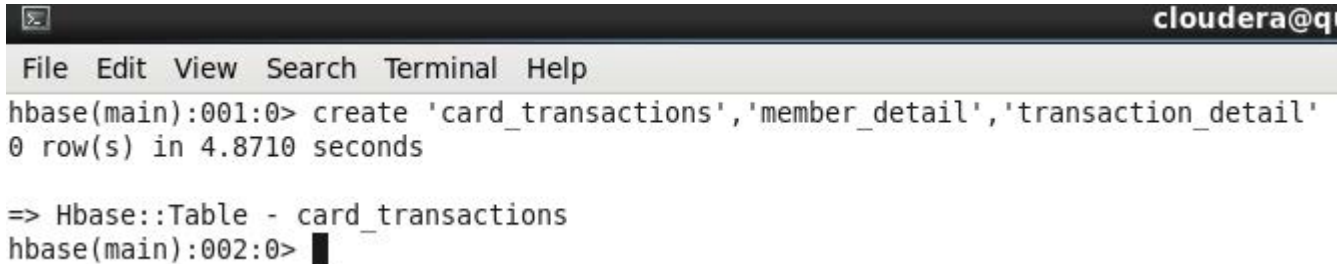
```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ sqoop job --create cred_member_score \  
> --meta-connect "jdbc:mysql://localhost/sqoop?user=sqoop&password=sqoop" \  
> -- import \  
> --connect jdbc:mysql://upgradawsrds.cpclxrkdvwzmz.us-east-1.rds.amazonaws.com/c  
red_financials_data \  
> --username upgraduser \  
> --password upgraduser \  
> --table member_score \  
> --warehouse-dir /user/sqoop_import/capstone_project  
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.  
Please set $ACCUMULO_HOME to the root of your Accumulo installation.  
19/11/13 08:59:54 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0  
19/11/13 08:59:56 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.  
[cloudera@quickstart ~]$ sqoop job -list  
Warning: /usr/lib/sqoop/./accumulo does not exist! Accumulo imports will fail.  
Please set $ACCUMULO_HOME to the root of your Accumulo installation.  
19/11/13 09:00:06 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0  
Available jobs:  
  cred_member_score  
  incremental_card_member  
[cloudera@quickstart ~]$
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
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)=13282  
  Total time spent by all reduces in occupied slots (ms)=0  
  Total time spent by all map tasks (ms)=13282  
  Total vcore-milliseconds taken by all map tasks=13282  
  Total megabyte-milliseconds taken by all map tasks=13600768  
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)=75  
  CPU time spent (ms)=1750  
  Physical memory (bytes) snapshot=184537088  
  Virtual memory (bytes) snapshot=1576345600  
  Total committed heap usage (bytes)=189792256  
File Input Format Counters  
  Bytes Read=0  
File Output Format Counters  
  Bytes Written=19980  
19/11/10 06:33:01 INFO mapreduce.ImportJobBase: Transferred 19.5117 KB in 39.0973 seconds (511.0327 bytes/sec)  
19/11/10 06:33:01 INFO mapreduce.ImportJobBase: Retrieved 999 records.  
[cloudera@quickstart ~]$
```

HBASE TABLE CREATION(card_transaction)

--Hbase shell command for table creation

```
create 'card_transactions','member_detail','transaction_detail'
```

A screenshot of a terminal window with a dark background and a light-colored menu bar. The menu bar contains the text 'File Edit View Search Terminal Help'. The terminal text shows the HBase shell prompt 'hbase(main):001:0>' followed by the command 'create \'card_transactions\',\'member_detail\',\'transaction_detail\''. Below this, it says '0 row(s) in 4.8710 seconds'. Then, it shows the confirmation prompt '=> Hbase::Table - card_transactions' and the next shell prompt 'hbase(main):002:0>' followed by a cursor.

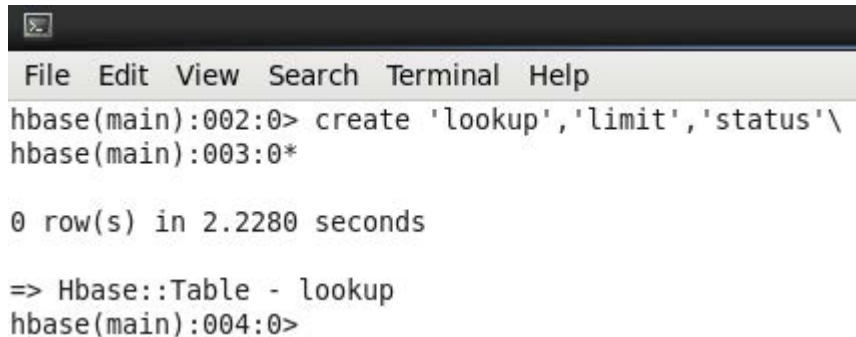
```
hbase(main):001:0> create 'card_transactions','member_detail','transaction_detail'
0 row(s) in 4.8710 seconds

=> Hbase::Table - card_transactions
hbase(main):002:0> █
```

HBASE TABLE CREATION (lookup table)

--Hbase shell command for table creation

```
create 'lookup','limit','status'
```

A screenshot of a terminal window with a dark background and a light-colored menu bar. The menu bar contains the text 'File Edit View Search Terminal Help'. The terminal text shows the HBase shell prompt 'hbase(main):002:0>' followed by the command 'create \'lookup\',\'limit\',\'status\''. Below this, it says 'hbase(main):003:0*' and '0 row(s) in 2.2280 seconds'. Then, it shows the confirmation prompt '=> Hbase::Table - lookup' and the next shell prompt 'hbase(main):004:0>' followed by a cursor.

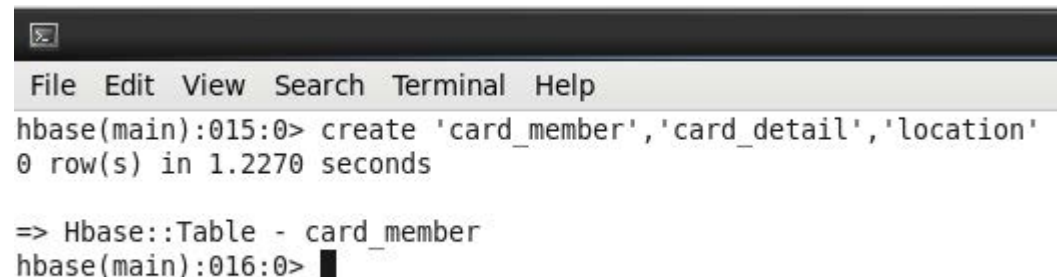
```
hbase(main):002:0> create 'lookup','limit','status'
hbase(main):003:0*
0 row(s) in 2.2280 seconds

=> Hbase::Table - lookup
hbase(main):004:0> █
```

HBASE TABLE CREATION (card member)

-- Hbase shell command for table creation

```
create 'card_member','card_detail','location'
```

A screenshot of a terminal window with a dark background and a light-colored menu bar. The menu bar contains the text 'File Edit View Search Terminal Help'. The terminal text shows the HBase shell prompt 'hbase(main):015:0>' followed by the command 'create \'card_member\',\'card_detail\',\'location\''. Below this, it says '0 row(s) in 1.2270 seconds'. Then, it shows the confirmation prompt '=> Hbase::Table - card_member' and the next shell prompt 'hbase(main):016:0>' followed by a cursor.

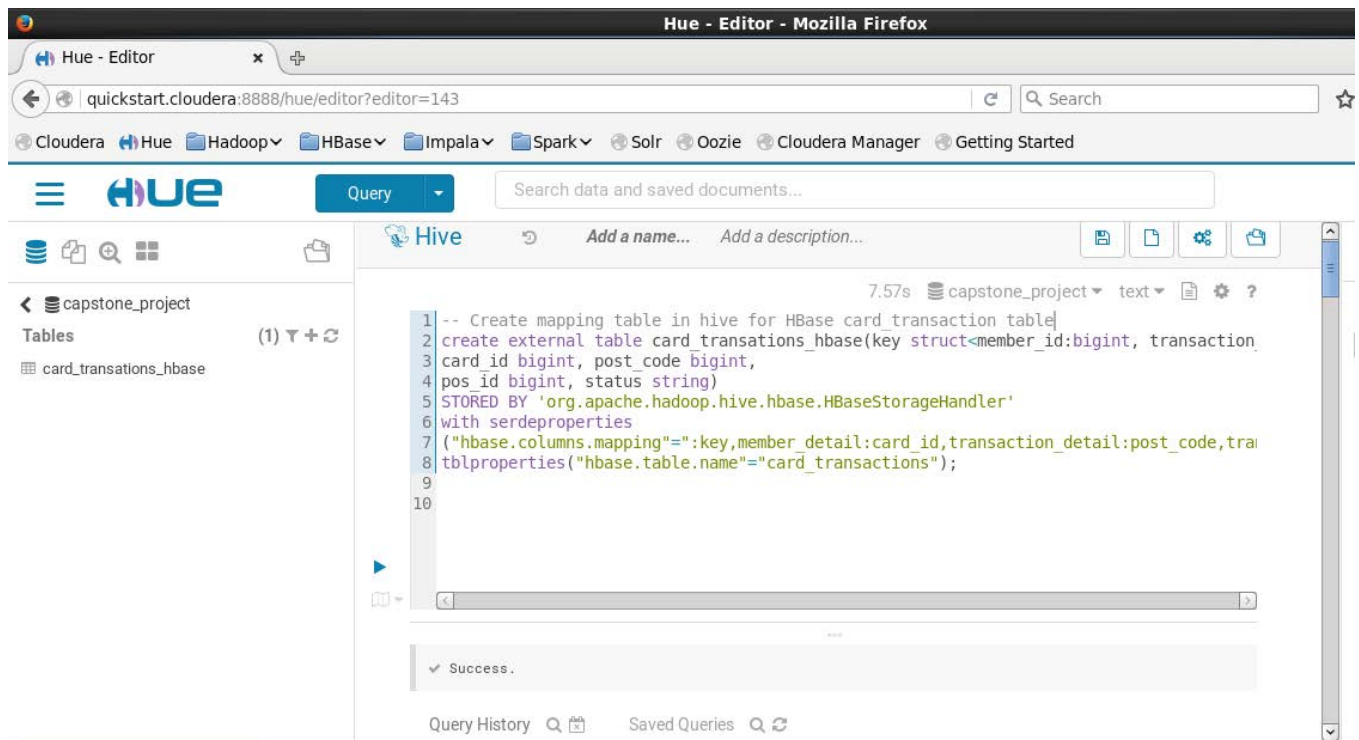
```
hbase(main):015:0> create 'card_member','card_detail','location'
0 row(s) in 1.2270 seconds

=> Hbase::Table - card_member
hbase(main):016:0> █
```

Create mapping table in hive for HBASE card_transaction table

--create mapping table in hive for HBase card transaction table

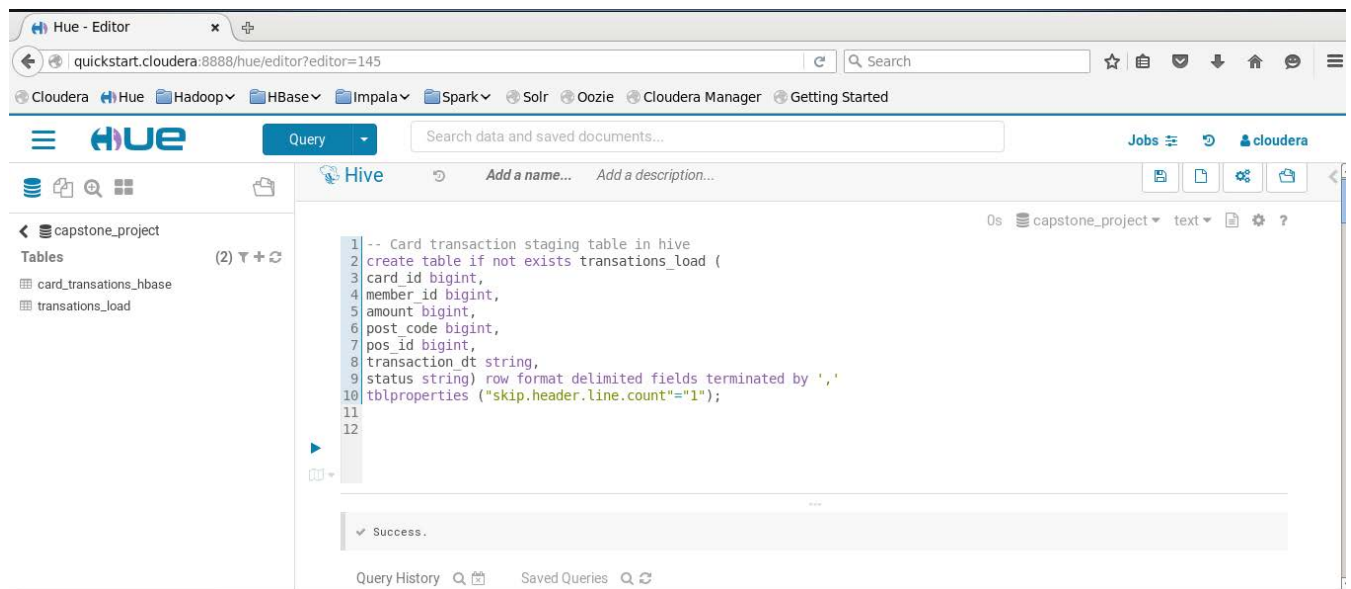
```
create external table card_transations_hbase(key struct<member_id:bigint, transaction_dt:string,
amount:bigint>,
card_id bigint, post_code bigint,
pos_id bigint, status string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key,member_detail:card_id,transaction_detail:post_code,transaction_
detail:pos_id,transaction_detail:status")
tblproperties("hbase.table.name"="card_transactions");
```



Card transaction staging table in hive

--Card transaction staging table in hive

```
create table if not exists transations_load (  
card_id bigint,  
member_id bigint,  
amount bigint,  
post_code bigint,  
pos_id bigint,  
transaction_dt string,  
status string) row format delimited fields terminated by ',' tblproperties  
("skip.header.line.count"="1");
```

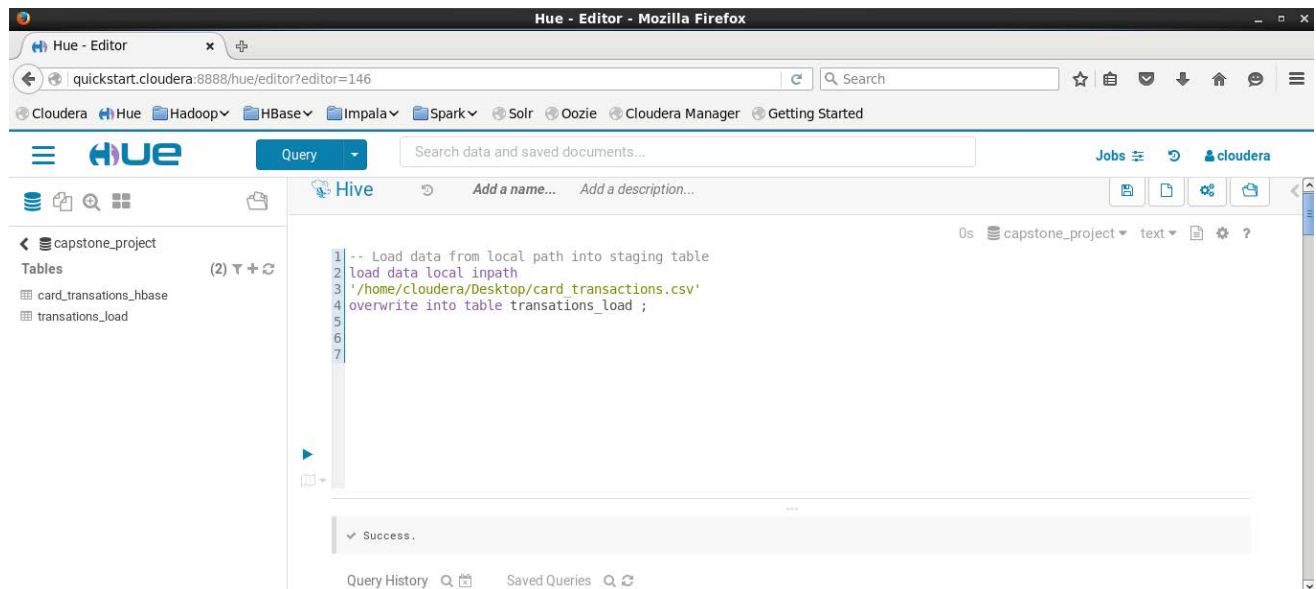


Load data from local path into staging table.

load data local inpath

'/home/cloudera/Desktop/card_transactions.csv'

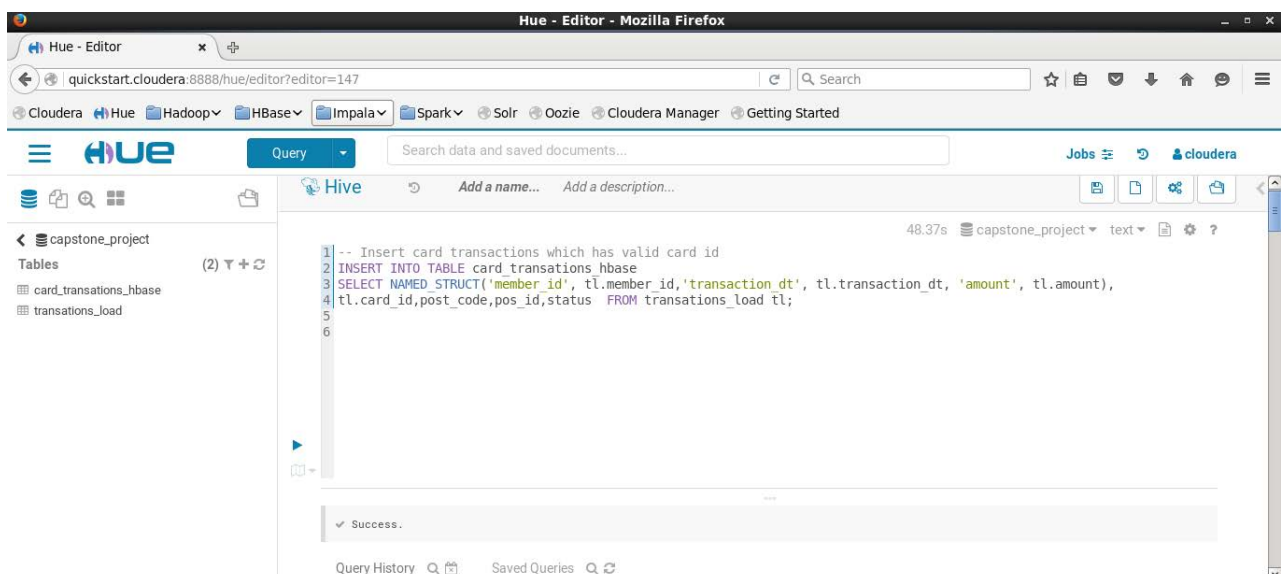
overwrite into table transations_load ;



Inserting data into card transaction HBase

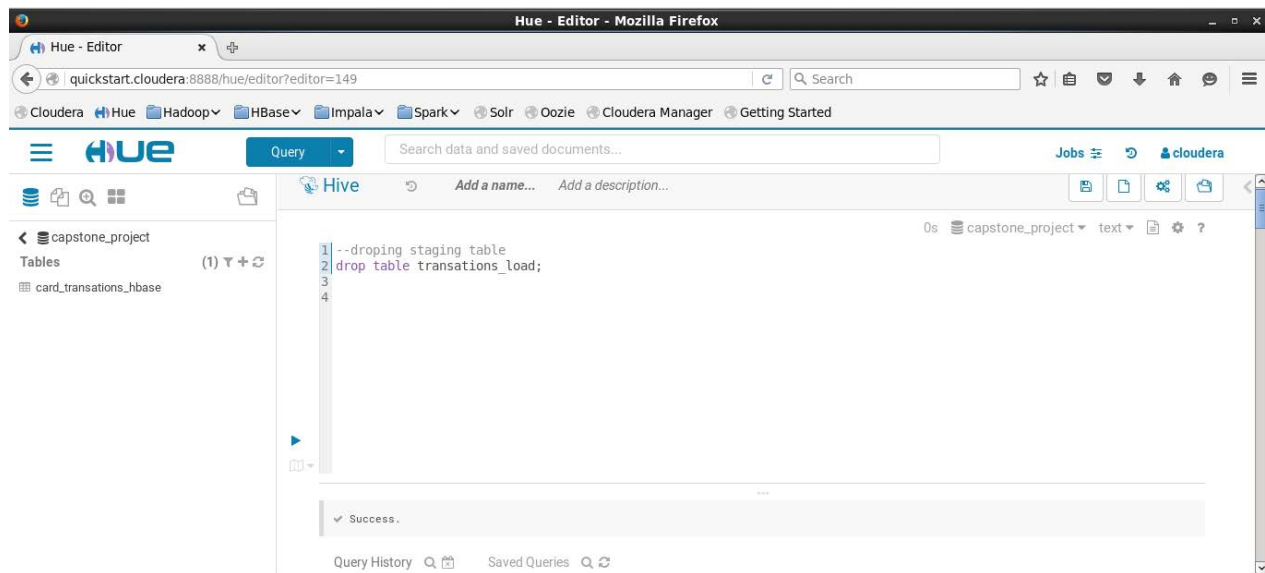
--Insert card transactions which has valid card id

```
INSERT INTO TABLE card_transactions_hbase
SELECT NAMED_STRUCT('member_id', tl.member_id, 'transaction_dt',
tl.transaction_dt, 'amount', tl.amount),
tl.card_id, post_code, pos_id, status FROM transations_load tl;
```



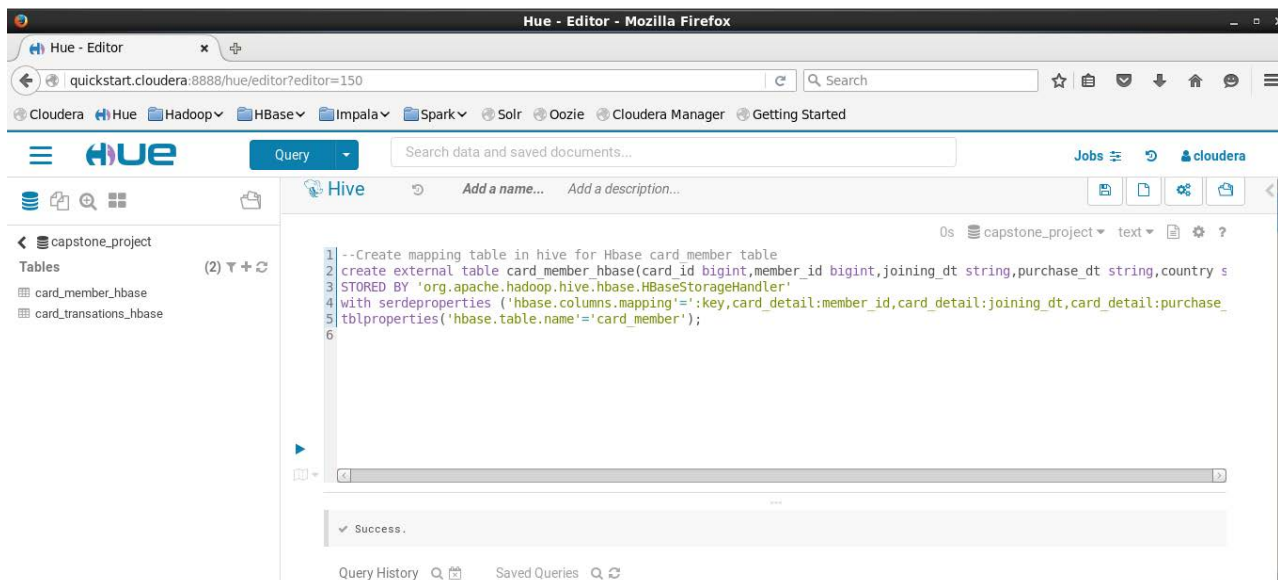
Dropping staging table

drop table transactions_load;



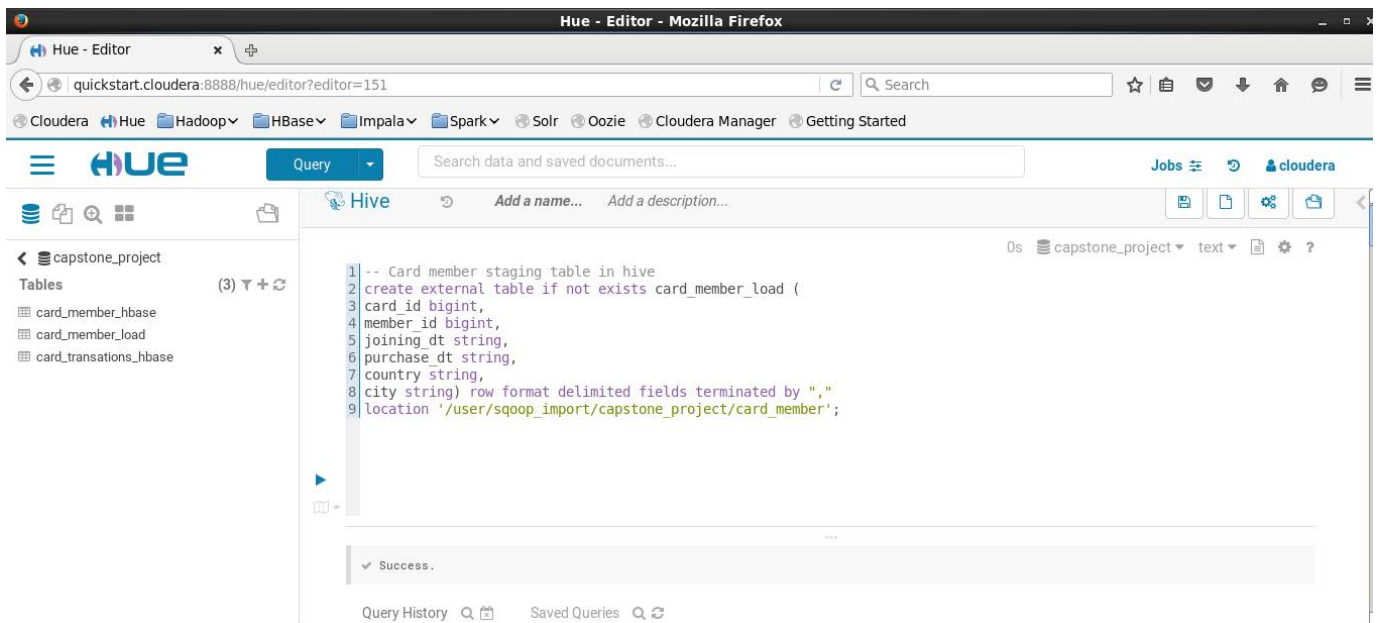
Create mapping table in hive for Hbase card_member table

```
create external table card_member_hbase(card_id bigint,member_id bigint,joining_dt
string,purchase_dt string,country string,city string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
('hbase.columns.mapping'=':key,card_detail:member_id,card_detail:joining_dt,card_detail:purchase
_dt ,location:country,location:city')
tblproperties('hbase.table.name'='card_member');
```



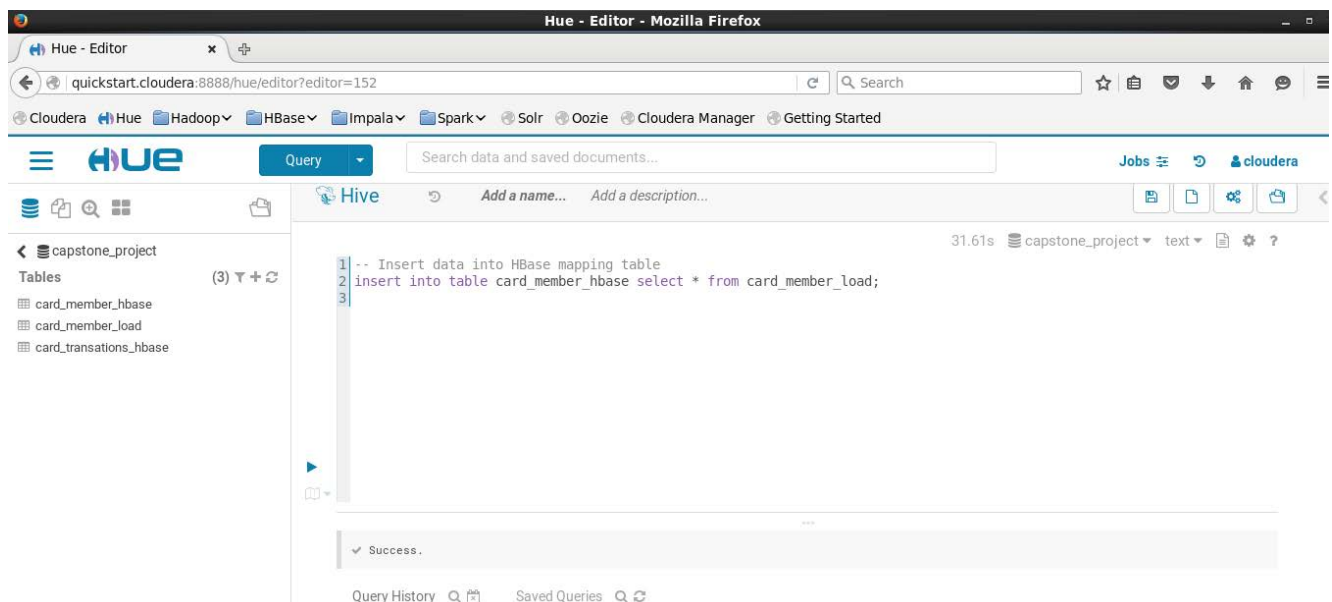
Card member staging table in hive

create external table if not exists card_member_load (
card_id bigint,
member_id bigint,
joining_dt string,
purchase_dt string,
country string,
city string) row format delimited fields terminated by "," location '/user/sqoop_import/
capstone_project/card_member';



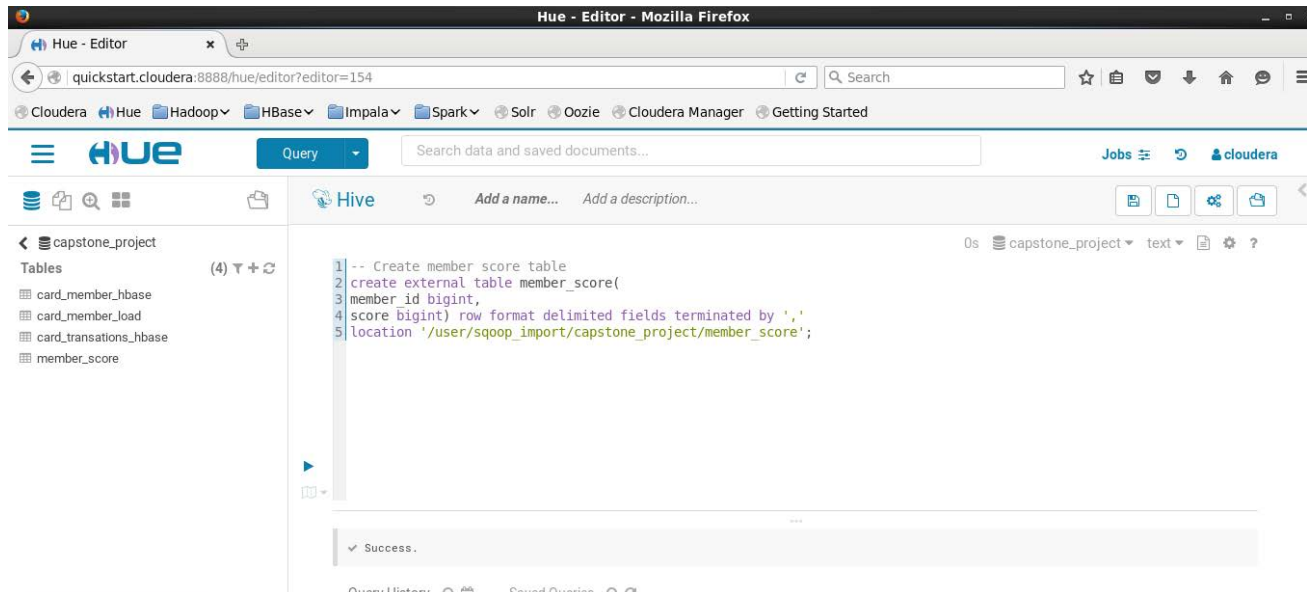
Insert data into HBase mapping table

insert into table card_member_hbase select * from card_member_load;



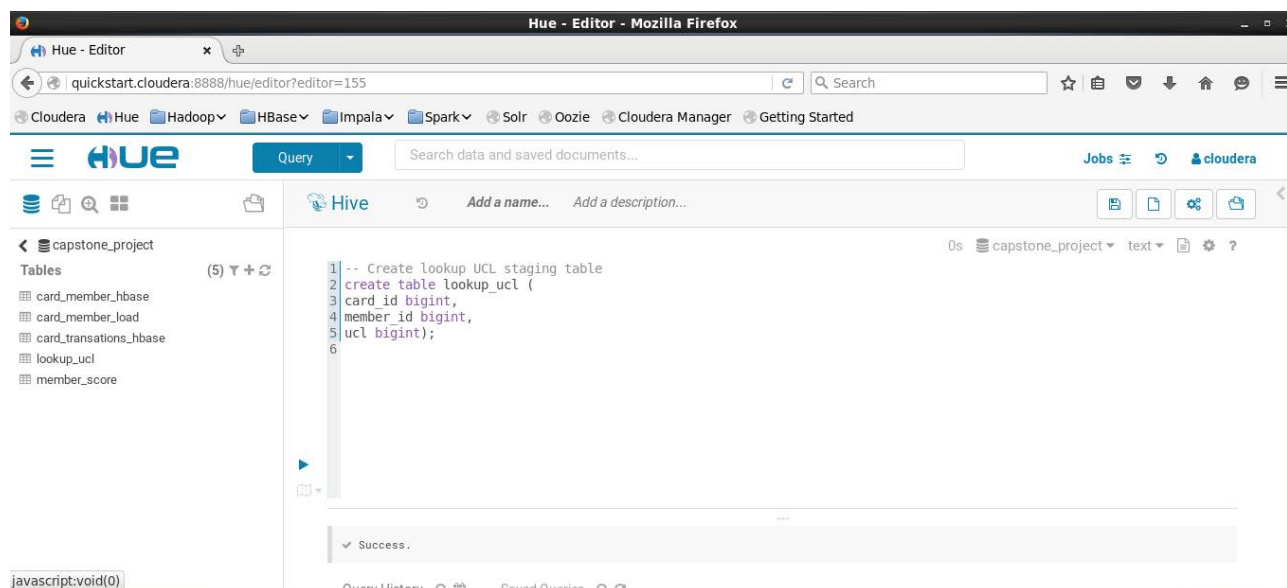
Create member score table

create external table member_score(
member_id bigint,
score bigint) row format delimited fields terminated by ',' location '/user/sqoop_import/
capstone_project/member_score';



Create lookup UCL staging table

create table lookup_ucl (card_id bigint, member_id bigint, ucl bigint);



Calculate UCL and insert into lookup UCL staging table

insert into table lookup_ucl

select cid, mid, (AVG(amt) + (3 * STDDEV_POP(amt))) as ucl from

(select card_id as cid, key.member_id as mid, key.amount as amt,

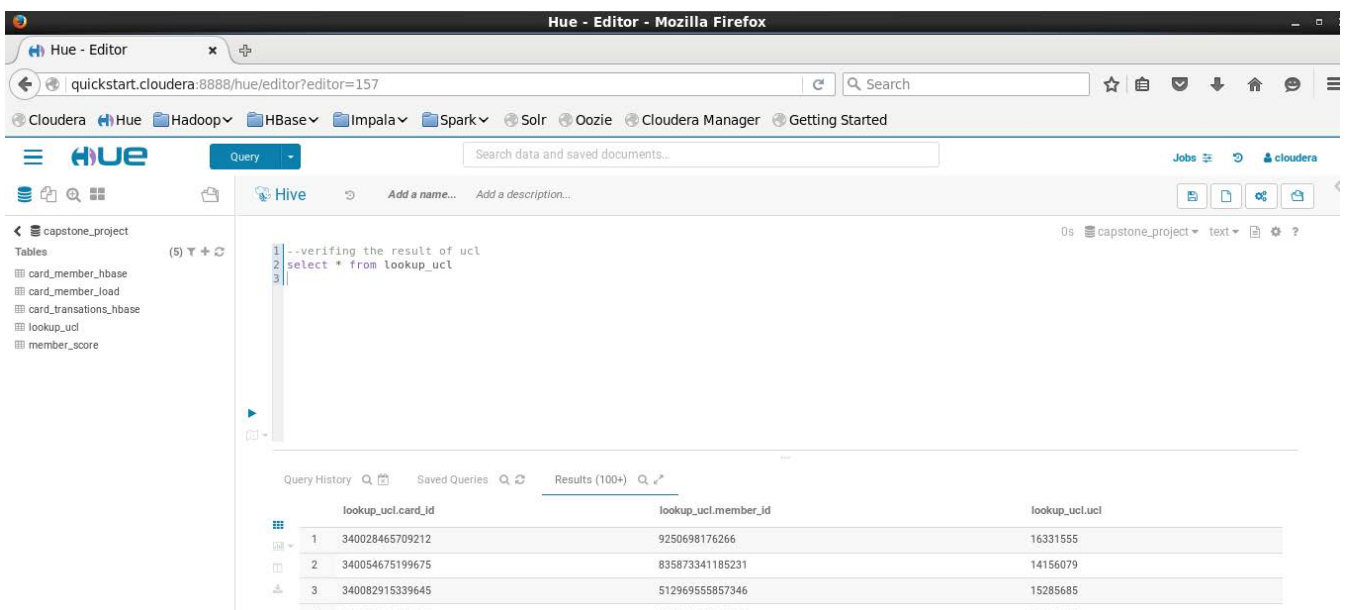
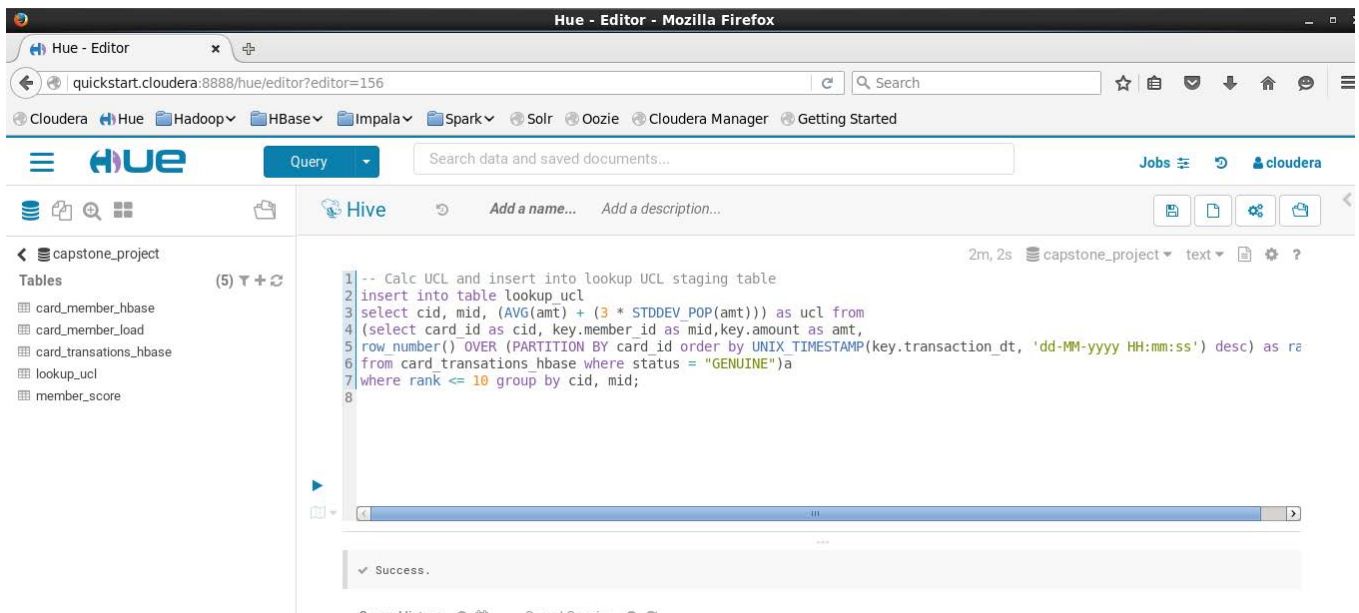
row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank

from card_transations_hbase where status = "GENUINE")a

where rank <= 10 group by cid, mid;

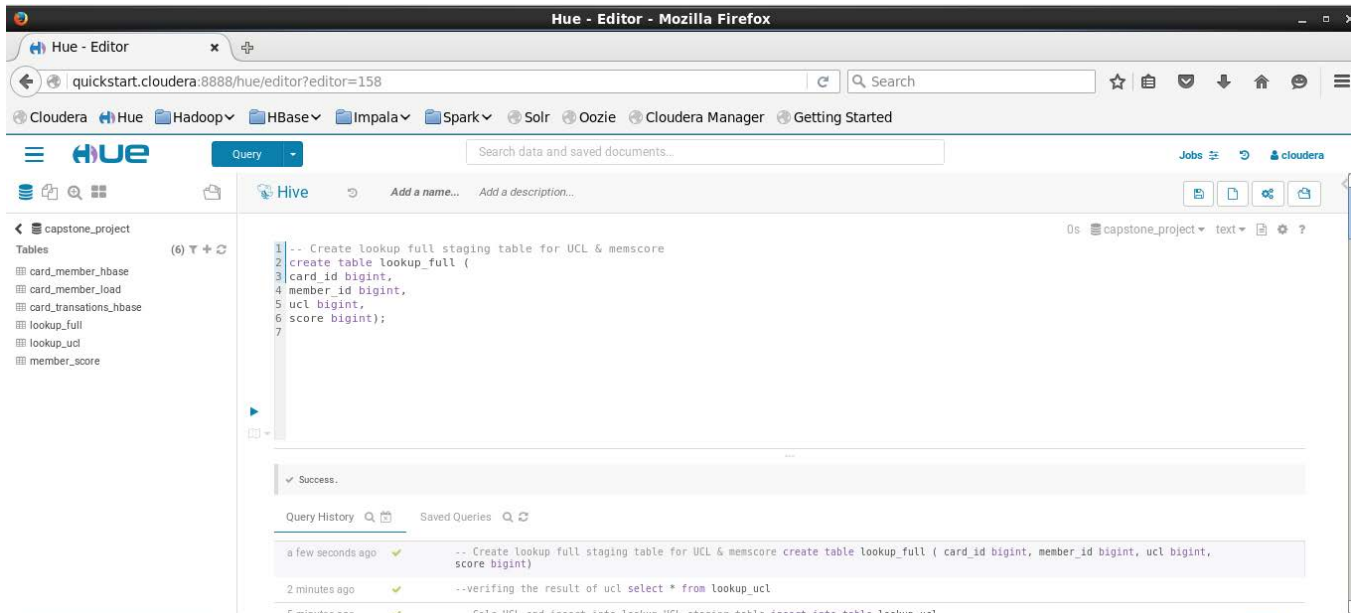
--Verifying result

select * from lookup_ucl



Create lookup full staging table for UCL & memscore

create table lookup_full (card_id bigint, member_id bigint, ucl bigint, score bigint);



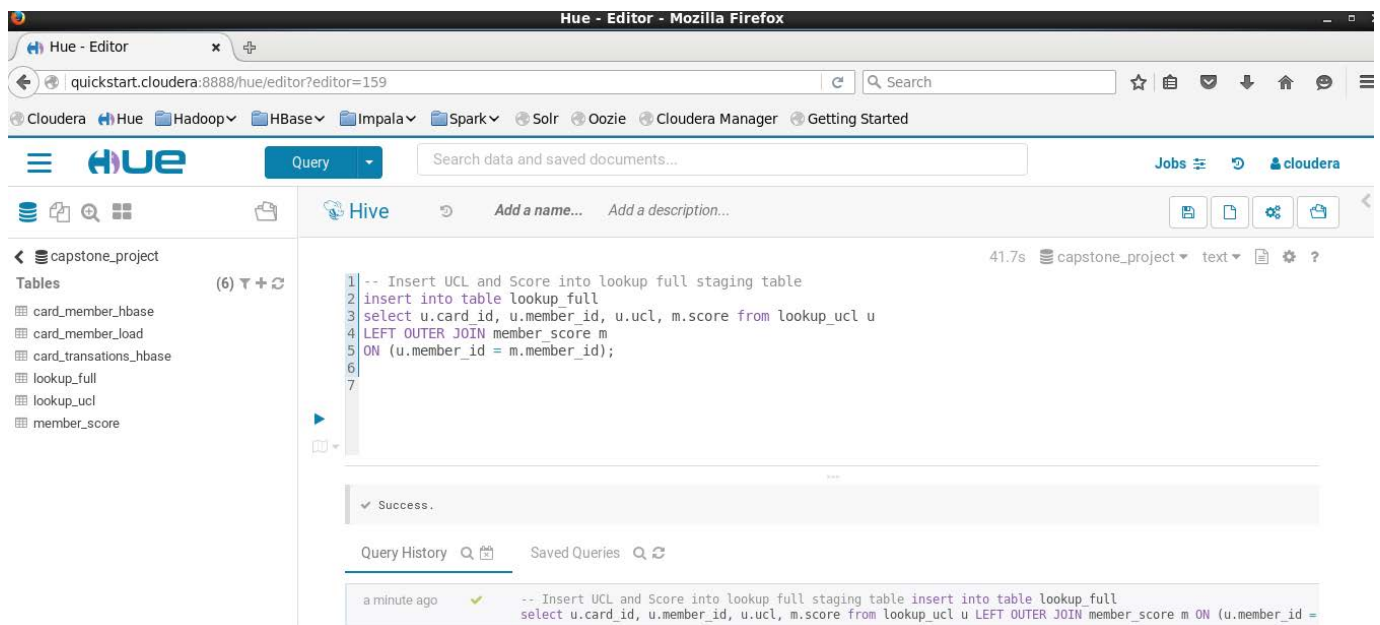
Insert UCL and Score into lookup full staging table

insert into table lookup_full

select u.card_id, u.member_id, u.ucl, m.score from lookup_ucl u LEFT OUTER JOIN

member_score m

ON (u.member_id = m.member_id);



Create mapping table for HBase lookup table for loading limit attributes

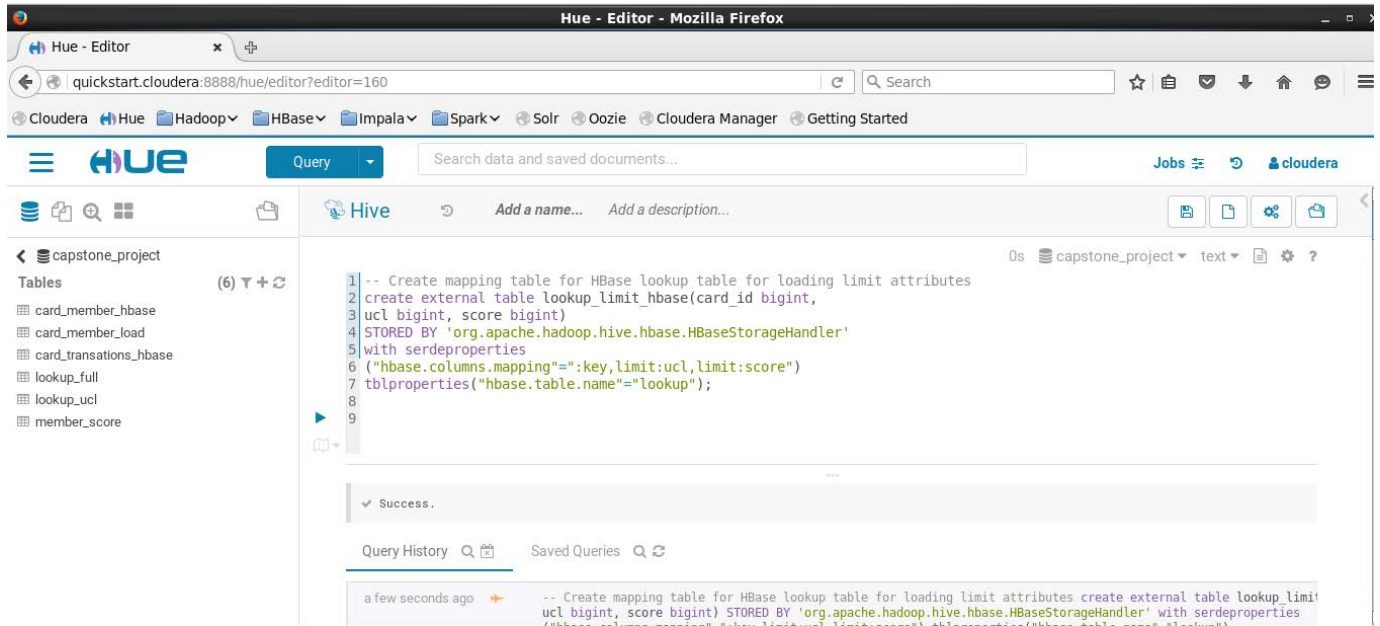
create external table lookup_limit_hbase(card_id bigint,
ucl bigint, score bigint)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

with serdeproperties

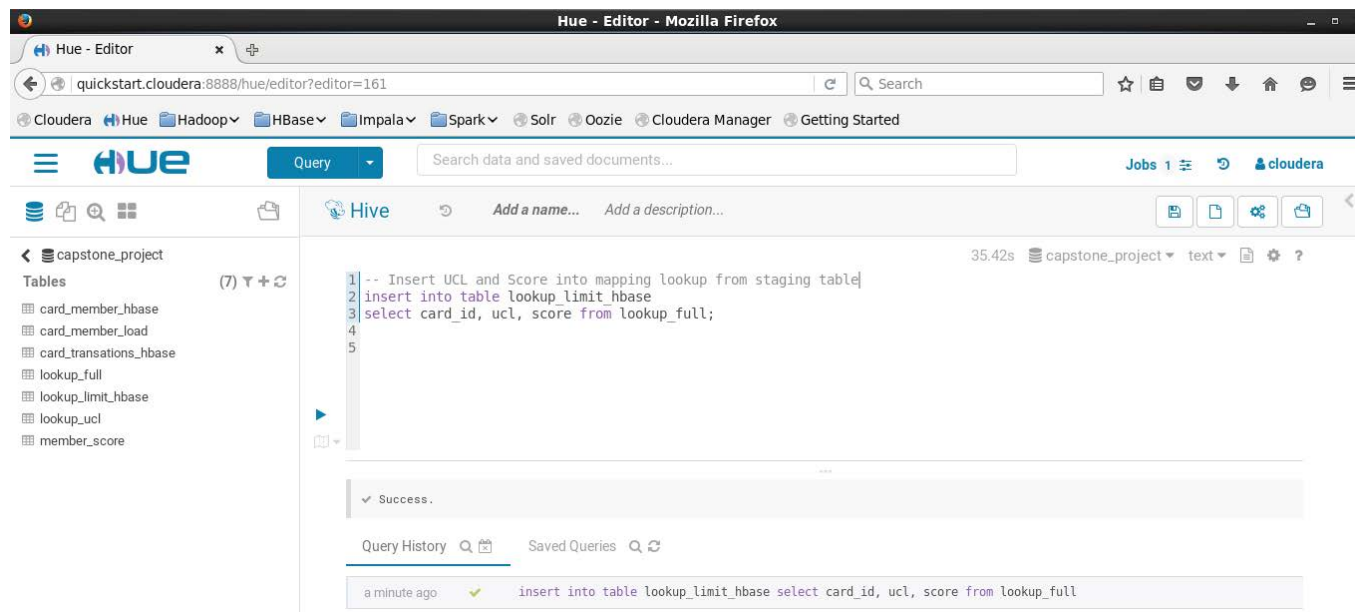
("hbase.columns.mapping"=":key,limit:ucl,limit:score")

tblproperties("hbase.table.name"="lookup");



Insert UCL and Score into mapping lookup from staging table

insert into table lookup_limit_hbase select card_id, ucl, score from lookup_full;



Create mapping table for HBase lookup table for loading status attributes

```
create external table lookup_status_hbase(cid bigint,
insert into table lookup_status_hbase
pc bigint, tdt string)
select card_id, post_code, td from
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
(select card_id, key.transaction_dt as td, post_code,
with serdeproperties
row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-
('hbase.columns.mapping'='"key,status:pc,status:tdt"')
yyyymmdd HH:mm:ss') desc) as rank
tblproperties("hbase.table.name"="lookup");
from card_transations_hbase where status = 'GENUINE') a
```

The screenshot shows the Hue web interface in a Mozilla Firefox browser. The interface includes a sidebar on the left with a table list under 'capstone_project'. The main area displays a Hive query being executed. Below the query editor, there is a 'Success' message and a 'Query History' section showing previous queries and their execution times.

```
1 -- Insert postal code and transaction date into mapping lookup from staging table
2 insert into table lookup_status_hbase
3 select card_id , post_code, td from
4 (select card_id ,key.transaction_dt as td, post_code,
5 row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank
6 from card_transations_hbase where status = 'GENUINE') a
7 where rank == 1;
```

Query History:

- a minute ago -- Insert postal code and transaction date into mapping lookup from staging table insert into table lookup_status_hbase select card_id , post code, td from (select card_id ,key.transaction_dt as td, post_code, row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank from card_transations_hbase where status = 'GENUINE') a where rank == 1
- 3 minutes ago -- Create mapping table for HBase lookup table for loading status attributes create external table lookup_status_hbase(cid bigint,

Insert postal code and transaction date into mapping lookup from card_transations_hbase

insert into table lookup_status_hbase

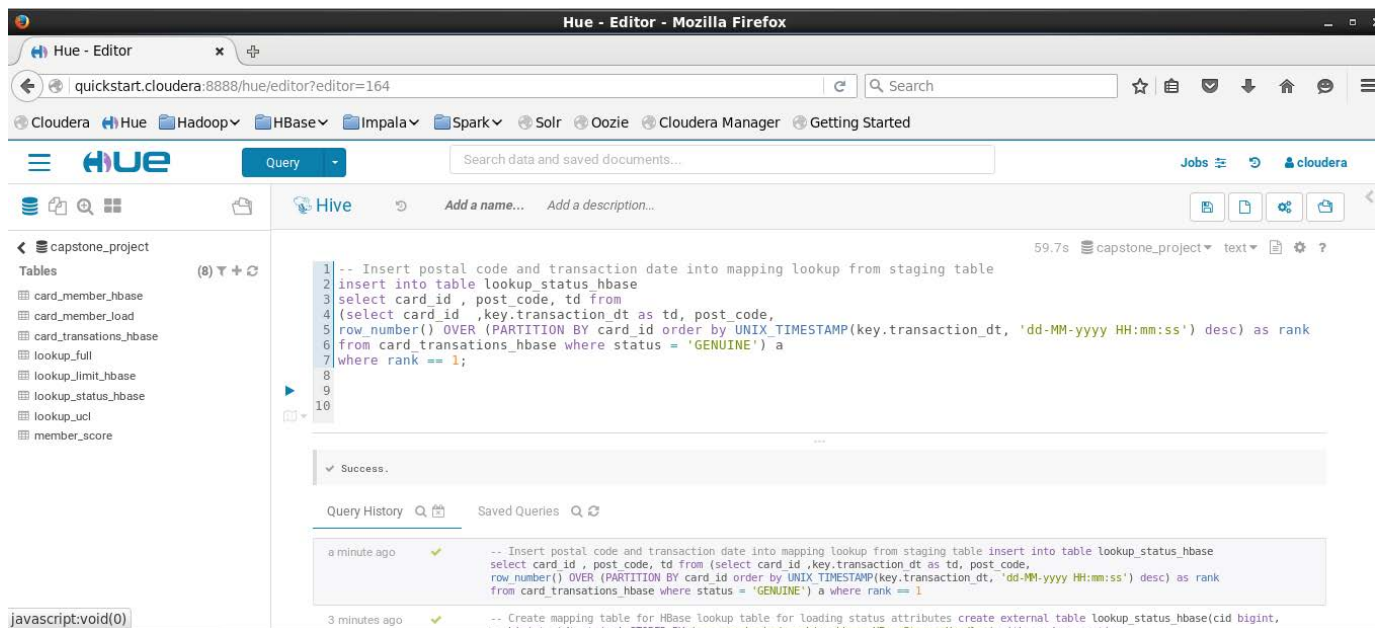
select card_id , post_code, td from

(select card_id ,key.transaction_dt as td, post_code,

row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank

from card_transations_hbase where status = 'GENUINE') a

where rank == 1;



Create mapping table for Hbase lookup table

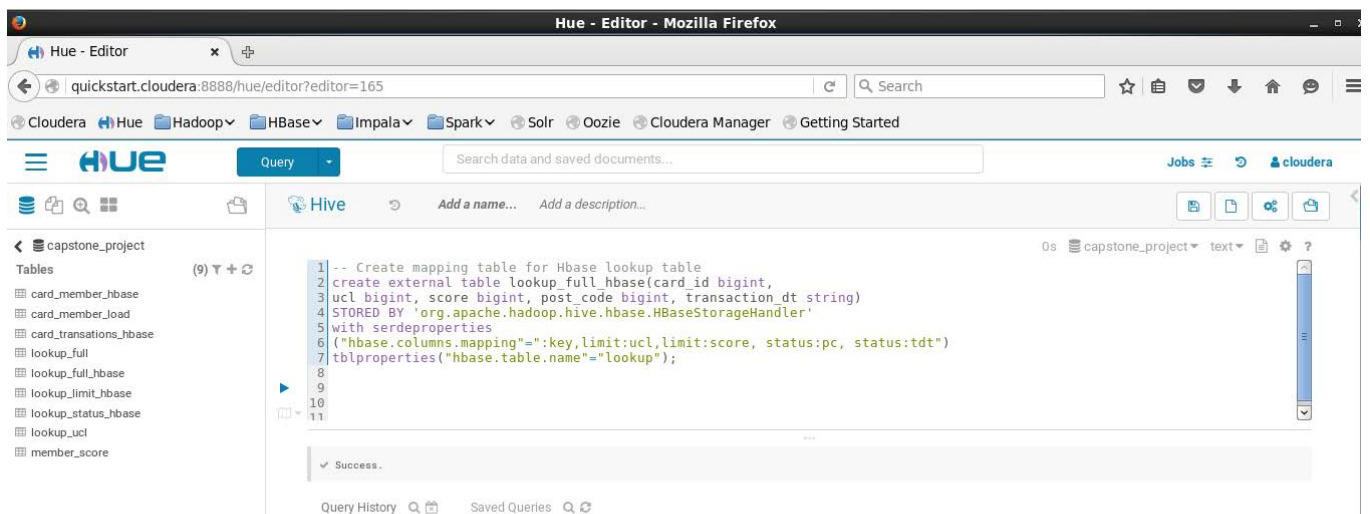
create external table lookup_full_hbase(card_id bigint,

ucl bigint, score bigint, post_code bigint, transaction_dt string)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' with serdeproperties

("hbase.columns.mapping"=":key,limit:ucl,limit:score, status:pc, status:tdt")

tblproperties("hbase.table.name"="lookup");



Drop tables

--Dropping all staging table

```
drop table lookup_ucl;  
drop table lookup_full;  
drop lookup_limit_hbase  
drop lookup_limit_hbase  
drop table lookup_status_hbase;
```