Mid-Submission – Logic Explanation

NOTES:

1. I have submitted below artifacts in mid submission for capstone project, organized in different folders:

   1. Task1/LoadCreateNoSQL.txt
   2. Task2/DataIngestion.txt
   3. Task3/PreAnalysis.txt
   4. Task4/Task4.txt
   5. Task4/lookupDataRefresh.hql
   6. Task4/job.properties.withoutcoordinator
   7. Task4/job.properties
   8. Task4/workflow.xml
   9. Task4/coordinator.xml
   10. Task4/sqoop-site.xml
   11. LogicMid.pdf

   I have deliberately provided Task4.txt to show commands used in Task 4. I have deliberately provided job.properties.withoutcoordinator, just in case, we want to test oozie workflow without coordinator. Just copy this as job.properties. Submitted job.properties has contents which setup oozie workflow with coordinator as that's how I was supposed to submit. **I am using HBase as NoSQL database for this project. Please zoom to 180% or 200% to see screenshots with better clarity.**

2. As per submission guidelines for mid submission, we need to submit only scripts and no output. It was also confirmed by TA in this discussion forum link: https://learn.upgrad.com/v/course/119/question/128584

3. We are supposed to have 2 sqoop jobs: 1 for incremental load of card_member table and 1 for full load of member_score table. It was confirmed by TA in this discussion forum link: https://learn.upgrad.com/v/course/119/question/127917

4. We can use random UUID to generate row key in HBase table to store card transactions related data. It was confirmed by TA in this discussion forum link: https://learn.upgrad.com/v/course/119/question/127664

5. I faced lot of issues while executing sqoop action and hive action from oozie workflow. I had to copy lot of jars and xmls in oozie shared library location for sqoop and hive respectively. All these are specified in task 4.

6. Setup directory in HDFS for the project. After connecting to ec2 instance via ec2-user, switch to root user and then to hdfs user. Create directory and change its ownership and then exit from hdfs user and then exit from root user and this will bring back to ec2-user.

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

7. All hadoop shell, hive, hbase, sqoop and oozie commands are executed via ec2-user.

8. Download card_transactions.csv from the resources section in the capstone project on the learning platform and transfer it to ec2 instance via WinSCP. Post that create a directory in HDFS and copy card_transactions.csv in that location.

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

Task 1: Load the transactions history data (card_transactions.csv) in a NoSQL database and create a look-up table with columns specified earlier in the problem statement in it.

================= Hive Commands: Start =====================

1. Start hive from command prompt. Create new database namely capstone_project and switch to use capstone_project.

   ```
   create database capstone_project;
   use capstone_project;
   ```

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;
   ```

3. Create external table card_transactions_ext table which will point to HDFS location created earlier.

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

4. Create table card_transactions_orc. ORC format will help in better performance.

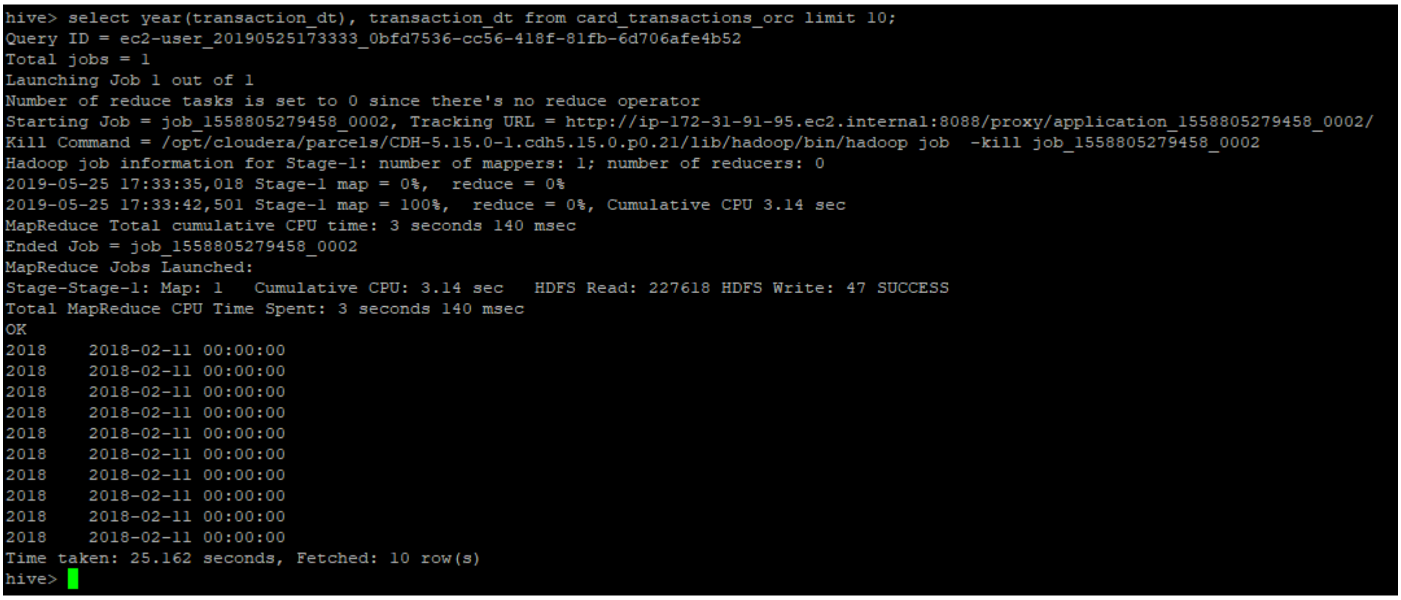
   ```
   CREATE TABLE IF NOT EXISTS CARD_TRANSACTIONS_ORC(
   `CARD_ID` STRING,
   `MEMBER_ID` STRING,
   `AMOUNT` DOUBLE,
   `POSTCODE` STRING,
   `POS_ID` STRING,
   `TRANSACTION_DT` TIMESTAMP,
   `STATUS` STRING)
   STORED AS ORC
   TBLPROPERTIES ("orc.compress"="SNAPPY");
   ```

5. Load data in card_transactions_orc while casting timestamp format for transaction_dt column.

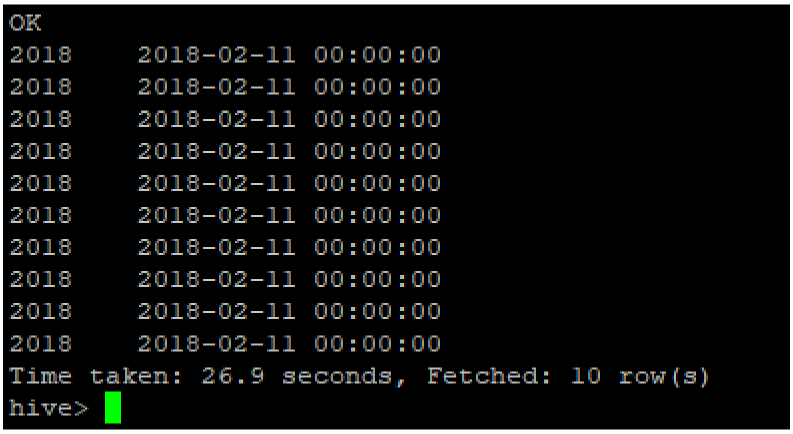
```
INSERT OVERWRITE TABLE CARD_TRANSACTIONS_ORC
SELECT CARD_ID, MEMBER_ID, AMOUNT, POSTCODE, POS_ID,
CAST(FROM_UNIXTIME(UNIX_TIMESTAMP(TRANSACTION_DT,'dd-MM-yyyy HH:mm:ss')) AS TIMESTAMP),
STATUS
FROM CARD_TRANSACTIONS_EXT;
```

6. Verify transaction_dt and year 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;
Query ID = ec2-user_20190525173333_0bfd7536-cc56-418f-81fb-6d706afe4b52
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1558805279458_0002, Tracking URL = http://ip-172-31-91-95.ec2.internal:8088/proxy/application_1558805279458_0002/
Kill Command = /opt/cloudera/parcels/CDH-5.15.0-1.cdh5.15.0.p0.21/lib/hadoop/bin/hadoop job  -kill job_1558805279458_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-05-25 17:33:35,018 Stage-1 map = 0%,  reduce = 0%
2019-05-25 17:33:42,501 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.14 sec
MapReduce Total cumulative CPU time: 3 seconds 140 msec
Ended Job = job_1558805279458_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.14 sec   HDFS Read: 227618 HDFS Write: 47 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 140 msec
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: 25.162 seconds, Fetched: 10 row(s)
hive>
```



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

7. Create card_transactions_hbase hive-hbase integrated table which will be visible in HBase as well. **This table will have all transactions – historical as well as new incoming from streaming layer.**

```
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)
```

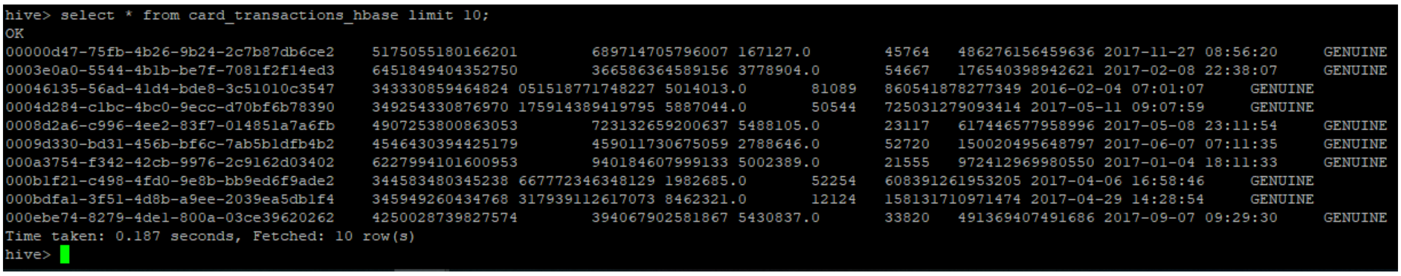
```
ROW FORMAT DELIMITED
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES
("hbase.columns.mapping"=":key, card_transactions_family:card_id, card_transactions_family:member_id,
card_transactions_family:amount, card_transactions_family:postcode, card_transactions_family:pos_id,
card_transactions_family:transaction_dt, card_transactions_family:status")
TBLPROPERTIES ("hbase.table.name"="card_transactions_hive");
```

8. Load data in card_transactions_hbase which will be visible in HBase as well with name as card_transactions_hive. Use randomUUID to populate TRANSACTION_ID field which will become row key in HBase effectively.

```
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;
```

9. Check some data in card_transactions_hbase.

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



```
hive> select * from card_transactions_hbase limit 10;
OK
00000d47-75fb-4b26-9b24-2c7b87db6ce2    5175055180166201        689714705796007 167127.0        45764   486276156459636 2017-11-27 08:56:20     GENUINE
0003e0a0-5544-4b1b-be7f-7081f2f14ed3    6451849404352750        366586364589156 3778904.0       54667   176540398942621 2017-02-08 22:38:07     GENUINE
00046135-56ad-41d4-bde8-3c51010c3547    343330859464824 051518771748227 5014013.0       81089   860541878277349 2016-02-04 07:01:07     GENUINE
0004d284-c1bc-4bc0-9ecc-d70bf6b78390    349254330876970 175914389419795 5887044.0       50544   725031279093414 2017-05-11 09:07:59     GENUINE
0008d2a6-c996-4ee2-83f7-014851a7a6fb    4907253800863053        723132659200637 5488105.0       23117   617446577958996 2017-05-08 23:11:54     GENUINE
0009d330-bd31-456b-bf6c-7ab5b1dfb4b2    4546430394425179        459011730675059 2788646.0       52720   150020495648797 2017-06-07 07:11:35     GENUINE
000a3754-f342-42cb-9976-2c9162d03402    6227994101600953        940184607999133 5002389.0       21555   972412969980550 2017-01-04 18:11:33     GENUINE
000b1f21-c498-4fd0-9e8b-bb9ed6f9ade2    344583480345238 667772346348129 1982685.0       52254   608391261953205 2017-04-06 16:58:46     GENUINE
000bdfa1-3f51-4d8b-a9ee-2039ea5db1f4    345949260434768 317939112617073 8462321.0       12124   158131710971474 2017-04-29 14:28:54     GENUINE
000ebe74-8279-4de1-800a-03ce39620262    4250028739827574        394067902581867 5430837.0       33820   491369407491686 2017-09-07 09:29:30     GENUINE
Time taken: 0.187 seconds, Fetched: 10 row(s)
hive>
```

10. Create lookup_data_hbase hive-hbase integrated table which will be visible in HBase as well with name as lookup_data_hive.

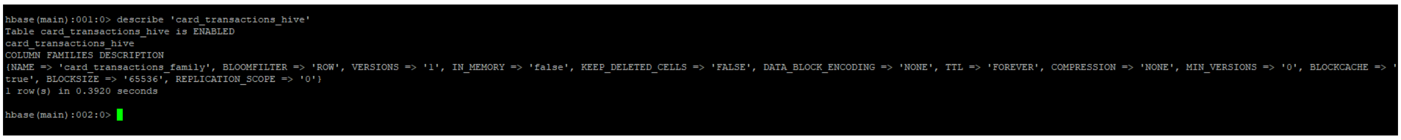
```
CREATE TABLE LOOKUP_DATA_HBASE(`CARD_ID` STRING,`UCL` DOUBLE, `SCORE` INT, `POSTCODE` STRING,
`TRANSACTION_DT` TIMESTAMP)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping"=":key, lookup_card_family:ucl,
lookup_card_family:score, lookup_transaction_family:postcode,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

================== Hive Commands: End =======================

================== HBase Commands: Start ======================

1. Start HBase shell from command prompt. In HBase, check details of card_transactions_hive hive-hbase integrated table.

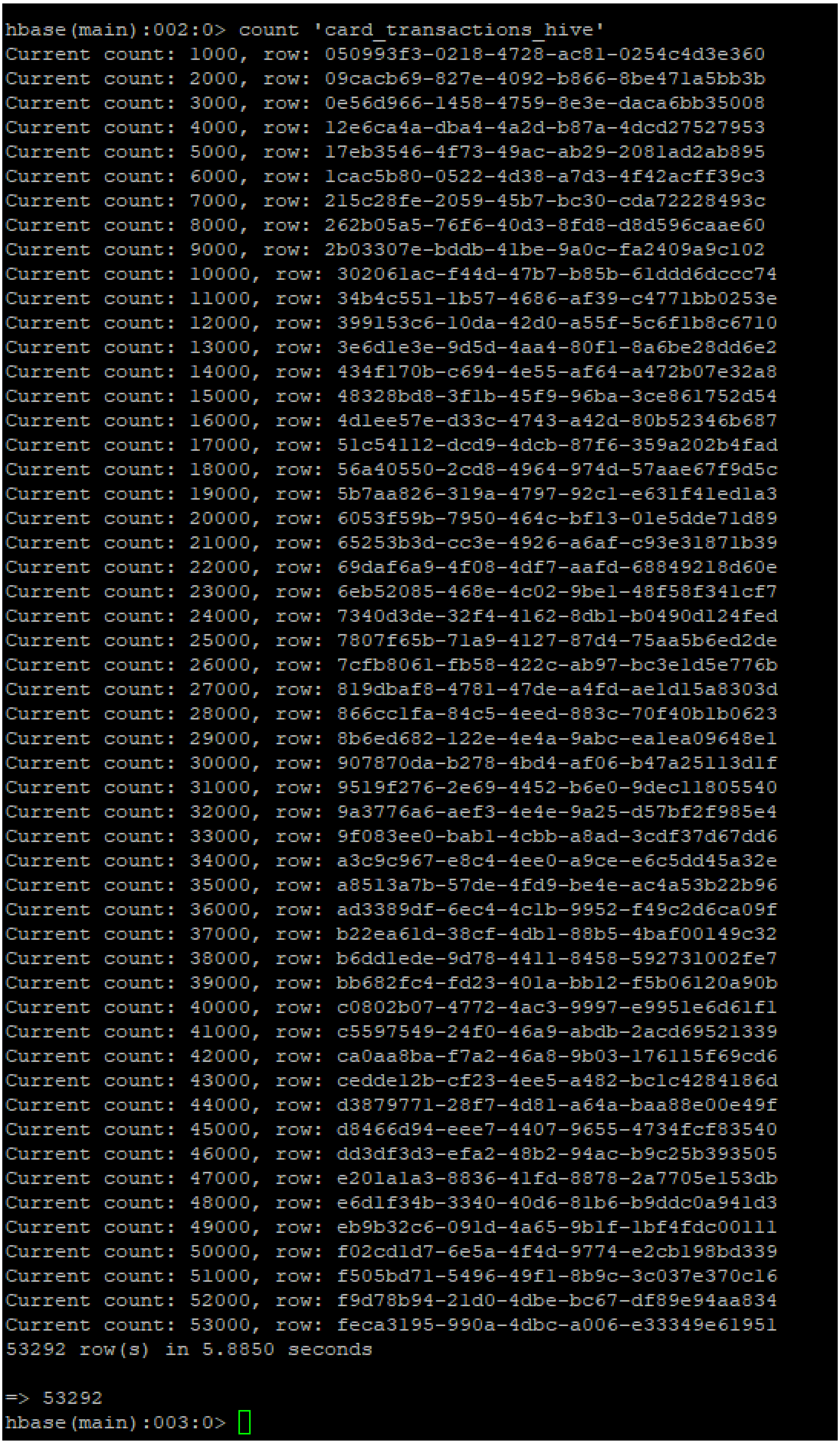
```
describe 'card_transactions_hive'
```



```
hbase(main):001:0> describe 'card_transactions_hive'
Table card_transactions_hive is ENABLED
card_transactions_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'card_transactions_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => '
true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.3920 seconds

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

2. In HBase, check count in card_transactions_hive in HBase

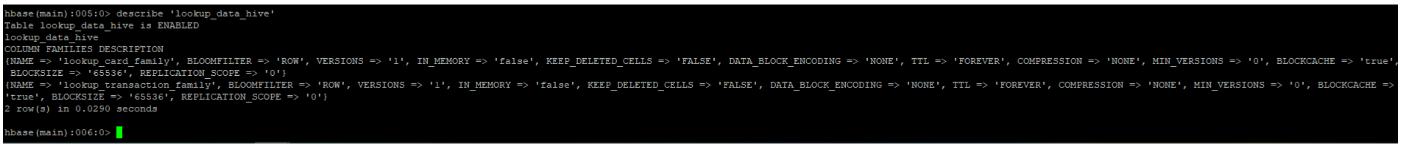
   count 'card_transactions_hive''



```
hbase(main):002:0> count 'card_transactions_hive'
Current count: 1000, row: 050993f3-0218-4728-ac81-0254c4d3e360
Current count: 2000, row: 09cacb69-827e-4092-b866-8be471a5bb3b
Current count: 3000, row: 0e56d966-1458-4759-8e3e-daca6bb35008
Current count: 4000, row: 12e6ca4a-dba4-4a2d-b87a-4dcd27527953
Current count: 5000, row: 17eb3546-4f73-49ac-ab29-2081ad2ab895
Current count: 6000, row: 1cac5b80-0522-4d38-a7d3-4f42acff39c3
Current count: 7000, row: 215c28fe-2059-45b7-bc30-cda72228493c
Current count: 8000, row: 262b05a5-76f6-40d3-8fd8-d8d596caae60
Current count: 9000, row: 2b03307e-bddb-41be-9a0c-fa2409a9c102
Current count: 10000, row: 302061ac-f44d-47b7-b85b-61ddd6dccc74
Current count: 11000, row: 34b4c551-1b57-4686-af39-c4771bb0253e
Current count: 12000, row: 399153c6-10da-42d0-a55f-5c6f1b8c6710
Current count: 13000, row: 3e6d1e3e-9d5d-4aa4-80f1-8a6be28dd6e2
Current count: 14000, row: 434f170b-c694-4e55-af64-a472b07e32a8
Current count: 15000, row: 48328bd8-3f1b-45f9-96ba-3ce861752d54
Current count: 16000, row: 4d1ee57e-d33c-4743-a42d-80b52346b687
Current count: 17000, row: 51c54112-dcd9-4dcb-87f6-359a202b4fad
Current count: 18000, row: 56a40550-2cd8-4964-974d-57aae67f9d5c
Current count: 19000, row: 5b7aa826-319a-4797-92c1-e631f41ed1a3
Current count: 20000, row: 6053f59b-7950-464c-bf13-01e5dde71d89
Current count: 21000, row: 65253b3d-cc3e-4926-a6af-c93e31871b39
Current count: 22000, row: 69daf6a9-4f08-4df7-aafd-68849218d60e
Current count: 23000, row: 6eb52085-468e-4c02-9be1-48f58f341cf7
Current count: 24000, row: 7340d3de-32f4-4162-8db1-b0490d124fed
Current count: 25000, row: 7807f65b-71a9-4127-87d4-75aa5b6ed2de
Current count: 26000, row: 7cfb8061-fb58-422c-ab97-bc3e1d5e776b
Current count: 27000, row: 819dbaf8-4781-47de-a4fd-ae1d15a8303d
Current count: 28000, row: 866cc1fa-84c5-4eed-883c-70f40b1b0623
Current count: 29000, row: 8b6ed682-122e-4e4a-9abc-ea1ea09648e1
Current count: 30000, row: 907870da-b278-4bd4-af06-b47a25113d1f
Current count: 31000, row: 9519f276-2e69-4452-b6e0-9dec11805540
Current count: 32000, row: 9a3776a6-aef3-4e4e-9a25-d57bf2f985e4
Current count: 33000, row: 9f083ee0-bab1-4cbb-a8ad-3cdf37d67dd6
Current count: 34000, row: a3c9c967-e8c4-4ee0-a9ce-e6c5dd45a32e
Current count: 35000, row: a8513a7b-57de-4fd9-be4e-ac4a53b22b96
Current count: 36000, row: ad3389df-6ec4-4c1b-9952-f49c2d6ca09f
Current count: 37000, row: b22ea61d-38cf-4db1-88b5-4baf00149c32
Current count: 38000, row: b6dd1ede-9d78-4411-8458-592731002fe7
Current count: 39000, row: bb682fc4-fd23-401a-bb12-f5b06120a90b
Current count: 40000, row: c0802b07-4772-4ac3-9997-e9951e6d61f1
Current count: 41000, row: c5597549-24f0-46a9-abdb-2acd69521339
Current count: 42000, row: ca0aa8ba-f7a2-46a8-9b03-176115f69cd6
Current count: 43000, row: cedde12b-cf23-4ee5-a482-bc1c4284186d
Current count: 44000, row: d3879771-28f7-4d81-a64a-baa88e00e49f
Current count: 45000, row: d8466d94-eee7-4407-9655-4734fcf83540
Current count: 46000, row: dd3df3d3-efa2-48b2-94ac-b9c25b393505
Current count: 47000, row: e201a1a3-8836-41fd-8878-2a7705e153db
Current count: 48000, row: e6d1f34b-3340-40d6-81b6-b9ddc0a941d3
Current count: 49000, row: eb9b32c6-091d-4a65-9b1f-1bf4fdc00111
Current count: 50000, row: f02cd1d7-6e5a-4f4d-9774-e2cb198bd339
Current count: 51000, row: f505bd71-5496-49f1-8b9c-3c037e370c16
Current count: 52000, row: f9d78b94-21d0-4dbe-bc67-df89e94aa834
Current count: 53000, row: feca3195-990a-4dbc-a006-e33349e61951
53292 row(s) in 5.8850 seconds

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

3. In HBase, check details of lookup_data_hive hive-hbase integrated table

   describe 'lookup_data_hive'



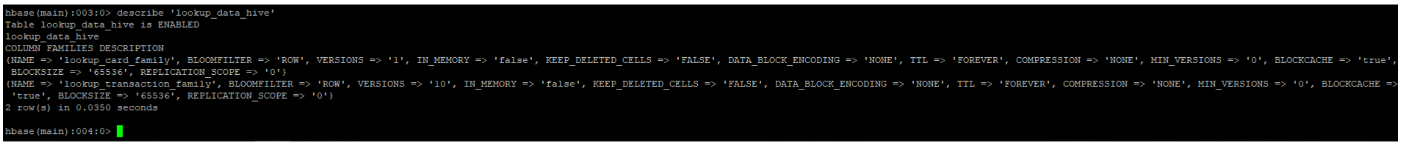
```
hbase(main):005:0> describe 'lookup_data_hive'
Table lookup_data_hive is ENABLED
lookup_data_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'lookup_card_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
 BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'lookup_transaction_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE =>
'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0290 seconds

hbase(main):006:0>
```

4. **In HBase, 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. I have created 2 column families in lookup table namely lookup_card_family and lookup_transaction_family. Column family lookup_card_family has score and ucl as columns and will store only 1 VERSION. Column family lookup_transaction_family has postcode and transaction_dt and will store 10 VERSIONS. It is not asked in the problem statement but a spark program can be written to fetch these 10 versions of transaction_dt and postcode from lookup table corresponding to a specific card_id and then we can loop over card_transactions_hive table to get last 10 transactions for a card_id using HBASE filters. It can result in better customer support service.**

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

5. In HBase, check details of lookup_data_hive and confirm that VERSIONS is set to 10 for lookup_transaction_family.

   describe 'lookup_data_hive'



```
hbase(main):003:0> describe 'lookup_data_hive'
Table lookup_data_hive is ENABLED
lookup_data_hive
COLUMN FAMILIES DESCRIPTION
{NAME => 'lookup_card_family', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
 BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'lookup_transaction_family', BLOOMFILTER => 'ROW', VERSIONS => '10', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE =>
 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0350 seconds

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

================== **HBase Commands: End** =======================

NOTE: **Regarding this requirement: The other details such as member_id, member_joining_dt, card_purchase_dt, country and city should be also stored for getting customer's information (dashboard for customer care executives).**

Since card member details are provided by 3rd party and available in AWS RDS, a separate API can be developed to fetch details for customer support service. Our application system mostly will be deployed on cloud and so this will be fast enough.

Task 2: Write a script to ingest the relevant data from AWS RDS to Hadoop.

================== Sqoop Commands: Start =======================

1. Run below Sqoop command to import member_score table from RDS into HDFS, from command prompt.

   ```
   sqoop import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data --username upgraduser --password upgraduser --table member_score --null-string 'NA' --null-non-string '\\N' --delete-target-dir --target-dir '/capstone_project/member_score'
   ```

2. Run below Sqoop command to import card_member table from RDS into HDFS, from command prompt.

   ```
   sqoop import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data --username upgraduser --password upgraduser --table card_member --null-string 'NA' --null-non-string '\\N' --delete-target-dir --target-dir '/capstone_project/card_member'
   ```

================== Sqoop Commands: End =======================

================== Hive Commands: Start =======================

1. Start hive from command prompt. Create external table card_member_ext which will point to HDFS location to hold data from card_member table in RDS. Sqoop command will write in this location.

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

2. Create external table member_score_ext which will point to HDFS location to hold data from member_score table in RDS. Sqoop command will write in this location.

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

3. Create card_member_orc table. ORC format will help in 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");
```

4. Create member_score_orc table. ORC format will help in better performance.

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

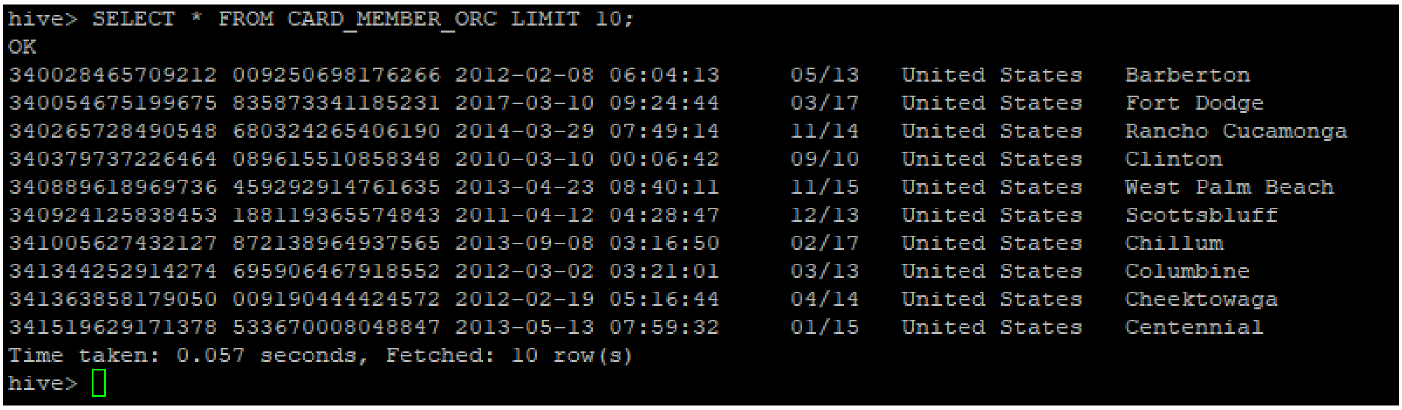
5. Load data into card_member_orc from card_member_ext.

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

6. Load data into member_score_orc from member_score_ext.

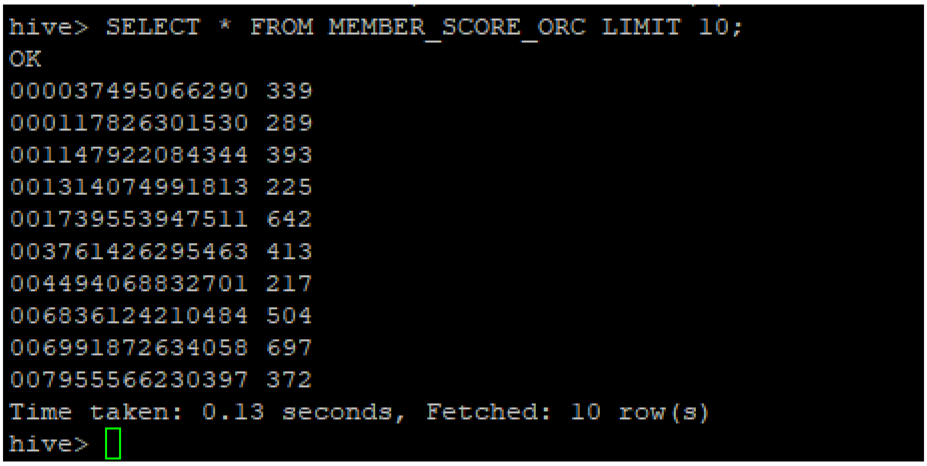
```
INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
SELECT MEMBER_ID, SCORE FROM MEMBER_SCORE_EXT;
```

7. Verify some 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
340265728490548 680324265406190 2014-03-29 07:49:14     11/14   United States   Rancho Cucamonga
340379737226464 089615510858348 2010-03-10 00:06:42     09/10   United States   Clinton
340889618969736 459292914761635 2013-04-23 08:40:11     11/15   United States   West Palm Beach
340924125838453 188119365574843 2011-04-12 04:28:47     12/13   United States   Scottsbluff
341005627432127 872138964937565 2013-09-08 03:16:50     02/17   United States   Chillum
341344252914274 695906467918552 2012-03-02 03:21:01     03/13   United States   Columbine
341363858179050 009190444424572 2012-02-19 05:16:44     04/14   United States   Cheektowaga
341519629171378 533670008048847 2013-05-13 07:59:32     01/15   United States   Centennial
Time taken: 0.057 seconds, Fetched: 10 row(s)
hive>
```

8. Verify some 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: 0.13 seconds, Fetched: 10 row(s)
hive>
```

================== **Hive Commands: End** ======================

Task 3: Write a script to calculate the moving average and standard deviation of the last 10 transactions for each card_id for the data present in Hadoop and NoSQL database. If the total number of transactions for a particular card_id is less than 10, then calculate the parameters based on the total number of records available for that card_id. The script should be able to extract and feed the other relevant data ('postcode', 'transaction_dt', 'score', etc.) for the look-up table along with card_id and UCL.

================== Hive Commands: Start =======================

1. Start hive from command prompt. Create table ranked_card_transactions_orc to store last 10 transactions for each card_id. ORC format will help in 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");
```

2. Create table card_ucl_orc to store UCL values for each card_id. ORC format will help in better performance.

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

3. Load data in ranked_card_transactions_orc table. In innermost query, select card_id, amount, postcode, transaction_dt from **card_transactions_hbase** where status is GENUINE. In immediate outer query, select card_id, amount, postcode, transaction_dt and create a new field namely rank using rank() analytic function by partitioning over card_id and order by transaction_dt in descending order, followed by amount in descending order. In outermost query, select card_id, amount, postcode, transaction_dt and rank where rank is less than or equal to 10. Insert all this data in ranked_card_transactions_orc. This will ensure that for each card_id, we have obtained last 10 transactions at the max and if any card_id does not have 10 transactions, then all transactions for that card_id have been obtained.

```
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;
```

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.

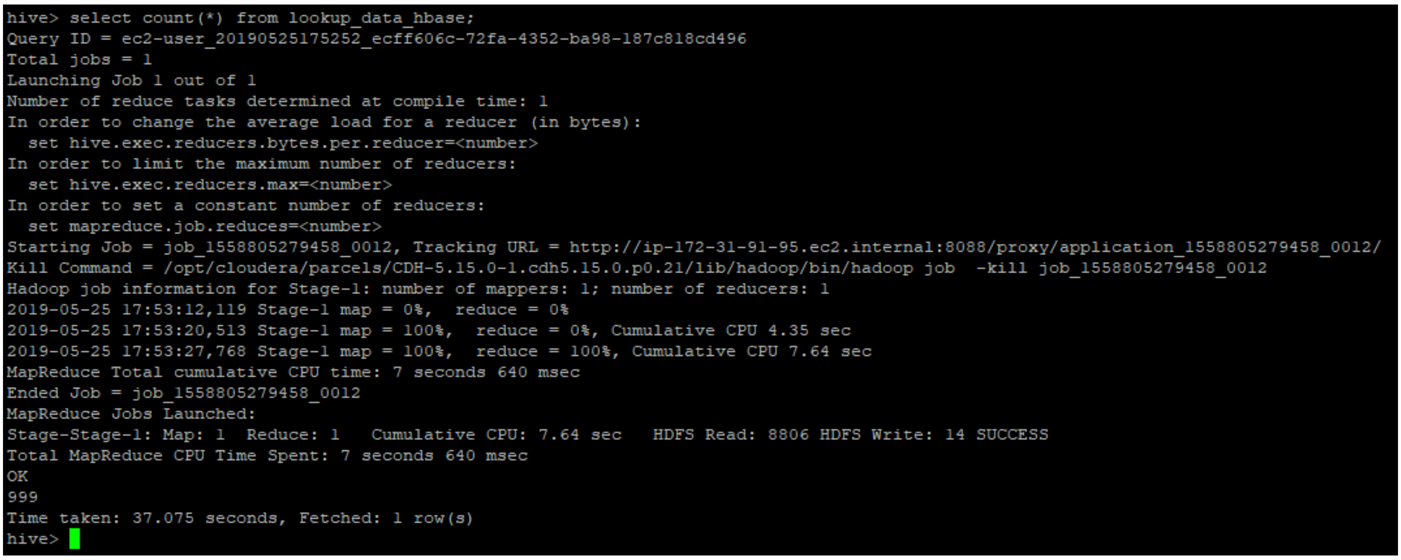
```
INSERT OVERWRITE TABLE CARD_UCL_ORC
SELECT A.CARD_ID, (A.AVERAGE + (3 * A.STANDARD_DEVIATION)) AS UCL FROM (
SELECT CARD_ID, AVG(AMOUNT) AS AVERAGE, STDDEV(AMOUNT) AS STANDARD_DEVIATION FROM
RANKED_CARD_TRANSACTIONS_ORC
GROUP BY CARD_ID) A;
```

5. Load data in lookup_data_hbase table. Create intermediate table or sort of inline view which can be used in JOIN condition by selecting card_id, score from card_member_orc joining member_score_orc on member_id and name it as CMS. In main query, select card_id, UCL, score, postcode, transaction_dt from ranked_card_transactions_orc joining card_ucl_orc on card_id column and joining cms on card_id where rank is 1. This will ensure that we have obtained data of latest transaction for each card_id.

```
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;
```

6. Verify count in lookup_data_hbase table.

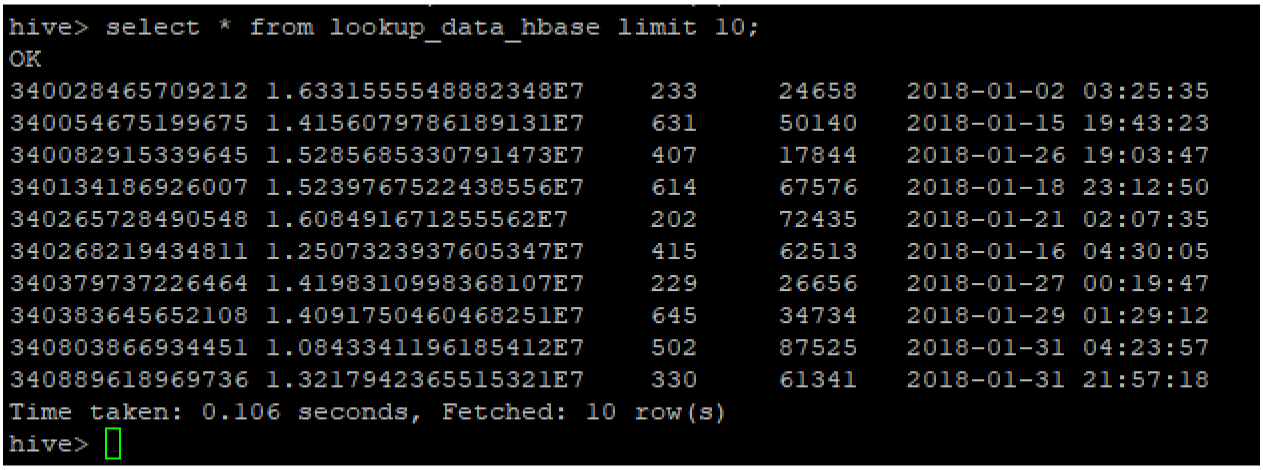
```
select count(*) from lookup_data_hbase;
```



```
hive> select count(*) from lookup_data_hbase;
Query ID = ec2-user_20190525175252_ecff606c-72fa-4352-ba98-187c818cd496
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1558805279458_0012, Tracking URL = http://ip-172-31-91-95.ec2.internal:8088/proxy/application_1558805279458_0012/
Kill Command = /opt/cloudera/parcels/CDH-5.15.0-1.cdh5.15.0.p0.21/lib/hadoop/bin/hadoop job  -kill job_1558805279458_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-05-25 17:53:12,119 Stage-1 map = 0%,  reduce = 0%
2019-05-25 17:53:20,513 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.35 sec
2019-05-25 17:53:27,768 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.64 sec
MapReduce Total cumulative CPU time: 7 seconds 640 msec
Ended Job = job_1558805279458_0012
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.64 sec   HDFS Read: 8806 HDFS Write: 14 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 640 msec
OK
999
Time taken: 37.075 seconds, Fetched: 1 row(s)
hive> 
```

7. Verify some data in lookup_data_hbase table.

```
select * from lookup_data_hbase limit 10;
```



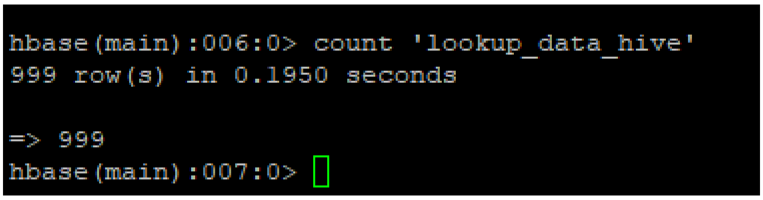
```
hive> select * from lookup_data_hbase limit 10;
OK
340028465709212 1.6331555548882348E7    233     24658   2018-01-02 03:25:35
340054675199675 1.4156079786189131E7    631     50140   2018-01-15 19:43:23
340082915339645 1.5285685330791473E7    407     17844   2018-01-26 19:03:47
340134186926007 1.5239767522438556E7    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: 0.106 seconds, Fetched: 10 row(s)
hive> 
```

================== Hive Commands: End =======================

================== HBase Commands: Start =======================

1. Start HBase shell from command prompt. In HBase, check count in lookup_data_hive table.

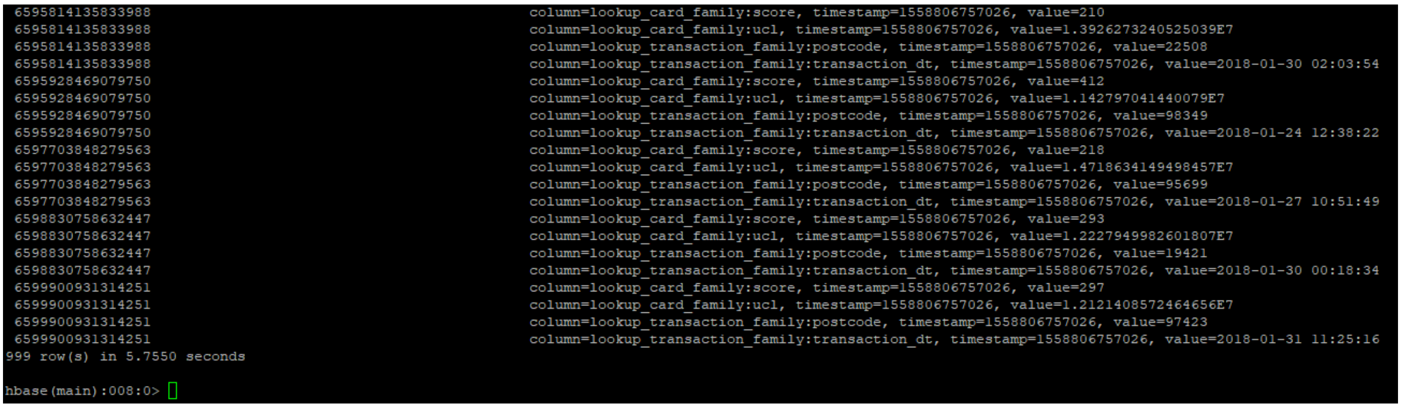
   count 'lookup_data_hive'



```
hbase(main):006:0> count 'lookup_data_hive'
999 row(s) in 0.1950 seconds

=> 999
hbase(main):007:0> 
```

2. In HBase, check data in lookup_data_hive table.

   scan 'lookup_data_hive'



```
 6595814135833988                                    column=lookup_card_family:score, timestamp=1558806757026, value=210
 6595814135833988                                    column=lookup_card_family:ucl, timestamp=1558806757026, value=1.3926273240525039E7
 6595814135833988                                    column=lookup_transaction_family:postcode, timestamp=1558806757026, value=22508
 6595814135833988                                    column=lookup_transaction_family:transaction_dt, timestamp=1558806757026, value=2018-01-30 02:03:54
 6595928469079750                                    column=lookup_card_family:score, timestamp=1558806757026, value=412
 6595928469079750                                    column=lookup_card_family:ucl, timestamp=1558806757026, value=1.142797041440079E7
 6595928469079750                                    column=lookup_transaction_family:postcode, timestamp=1558806757026, value=98349
 6595928469079750                                    column=lookup_transaction_family:transaction_dt, timestamp=1558806757026, value=2018-01-24 12:38:22
 6597703848279563                                    column=lookup_card_family:score, timestamp=1558806757026, value=218
 6597703848279563                                    column=lookup_card_family:ucl, timestamp=1558806757026, value=1.4718634149498457E7
 6597703848279563                                    column=lookup_transaction_family:postcode, timestamp=1558806757026, value=95699
 6597703848279563                                    column=lookup_transaction_family:transaction_dt, timestamp=1558806757026, value=2018-01-27 10:51:49
 6598830758632447                                    column=lookup_card_family:score, timestamp=1558806757026, value=293
 6598830758632447                                    column=lookup_card_family:ucl, timestamp=1558806757026, value=1.2227949982601807E7
 6598830758632447                                    column=lookup_transaction_family:postcode, timestamp=1558806757026, value=19421
 6598830758632447                                    column=lookup_transaction_family:transaction_dt, timestamp=1558806757026, value=2018-01-30 00:18:34
 6599900931314251                                    column=lookup_card_family:score, timestamp=1558806757026, value=297
 6599900931314251                                    column=lookup_card_family:ucl, timestamp=1558806757026, value=1.2121408572464656E7
 6599900931314251                                    column=lookup_transaction_family:postcode, timestamp=1558806757026, value=97423
 6599900931314251                                    column=lookup_transaction_family:transaction_dt, timestamp=1558806757026, value=2018-01-31 11:25:16
999 row(s) in 5.7550 seconds

hbase(main):008:0> 
```

================== HBase Commands: End =======================

Task 4: Set up a job scheduler to schedule the scripts run after every 4 hours. The job should take the data from the NoSQL database and AWS RDS and perform the relevant analyses as per the rules and should feed the data in the look-up table.Task 4: Set up a job scheduler to schedule the scripts run after every 4 hours. The job should take the data from the NoSQL database and AWS RDS and perform the relevant analyses as per the rules and should feed the data in the look-up table.

================== Sqoop Commands: Start ======================

1. Start sqoop metastore from command prompt.

   sudo -u sqoop sqoop-metastore

2. Run below command to setup sqoop job to import card_member data incrementally from RDS into HDFS, from command prompt.

   sqoop job --create extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop -- import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data --username upgraduser --password upgraduser --table card_member --null-string 'NA' --null-non-string '\\N' --incremental lastmodified --check-column member_joining_dt --last-value 0 --merge-key card_id --target-dir '/capstone_project/card_member'

3. Run below command to setup sqoop job to import member_score data from RDS into HDFS, from command prompt.

   sqoop job --create extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop -- import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data --username upgraduser --password upgraduser --table member_score --null-string 'NA' --null-non-string '\\N' --delete-target-dir --target-dir '/capstone_project/member_score'

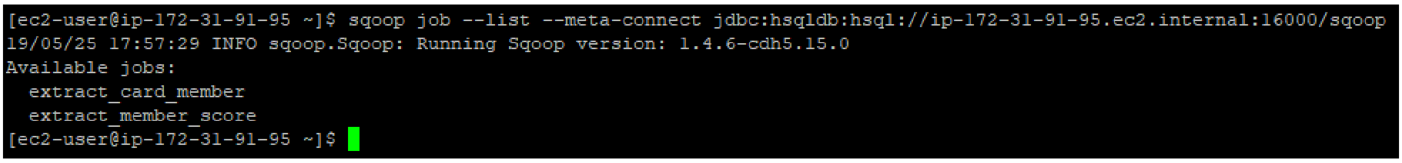
4. Execute sqoop jobs once from command prompt so just to be sure if setup correctly using below commands.

   sqoop job --exec extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop

   sqoop job --exec extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop

5. Verify sqoop jobs using below commands.

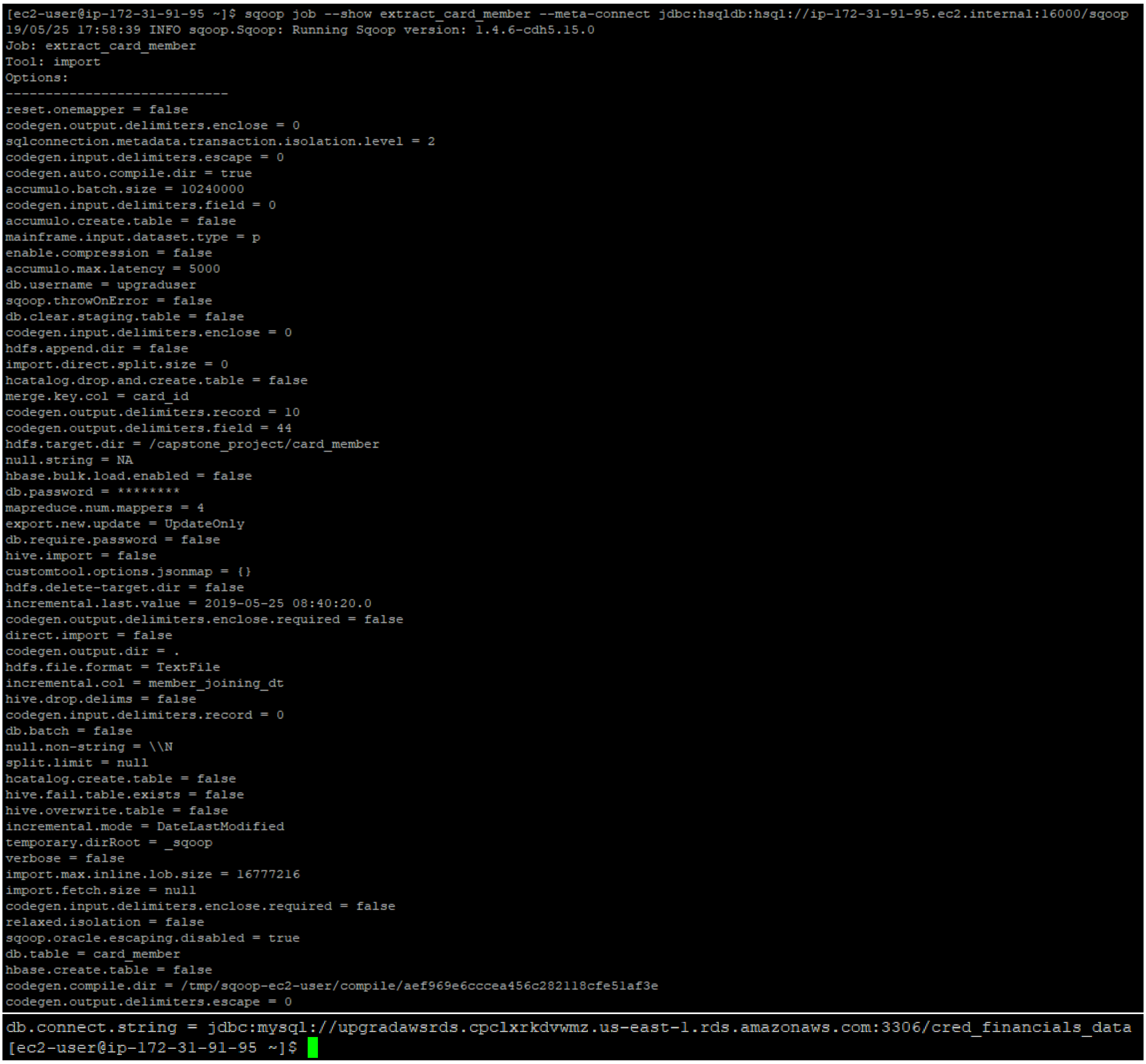
   sqoop job --list --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop



```
[ec2-user@ip-172-31-91-95 ~]$ sqoop job --list --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop
19/05/25 17:57:29 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.0
Available jobs:
  extract_card_member
  extract_member_score
[ec2-user@ip-172-31-91-95 ~]$
```

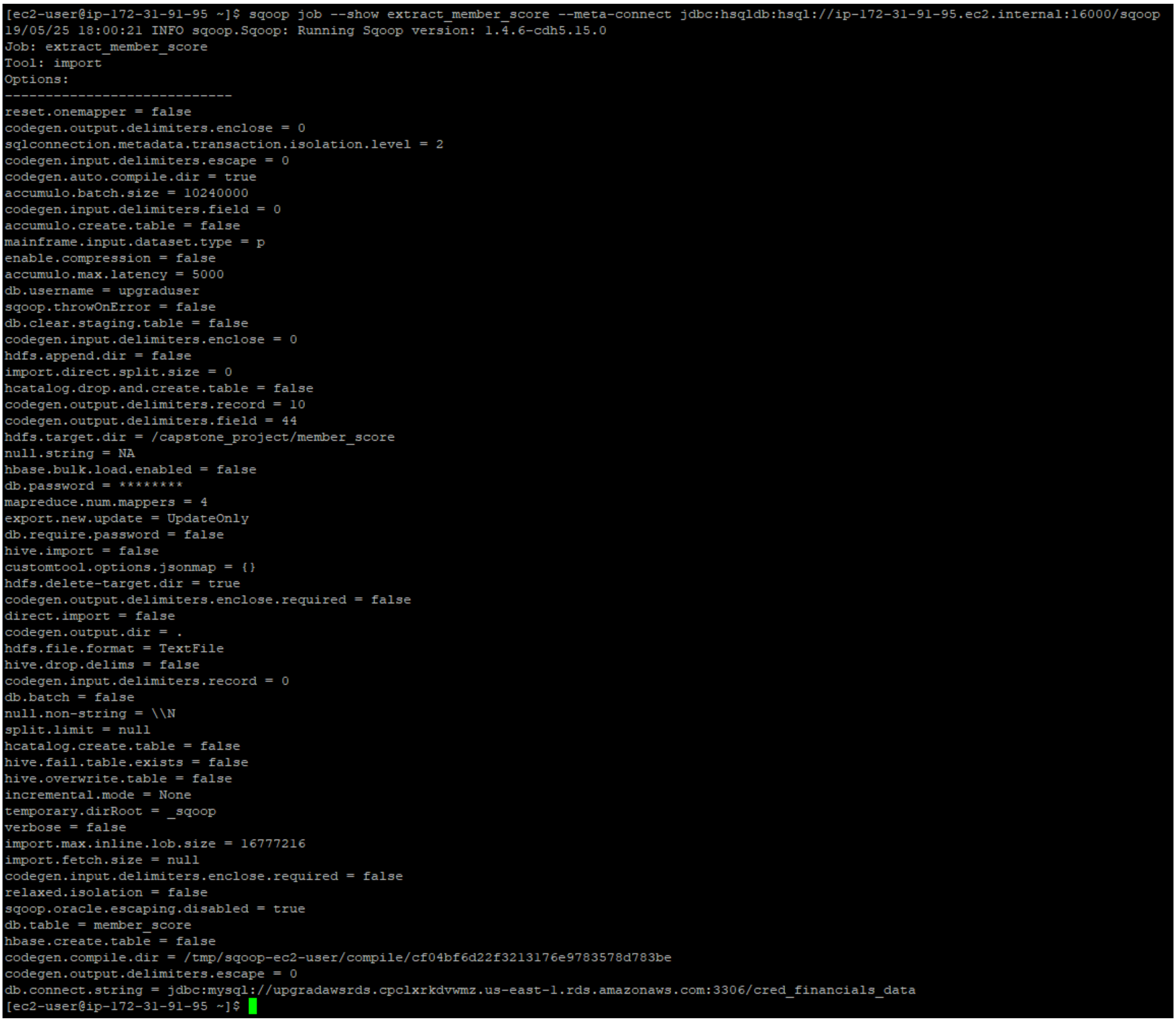
sqoop job --show extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop

NOTE: This below screenshot was taken after job was run once. So you would notice that incremental.last.value has changed to the timestamp when job was run. Next run will import data having member_joining_dt after this timestamp.



```
[ec2-user@ip-172-31-91-95 ~]$ sqoop job --show extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop
19/05/25 17:58:39 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.0
Job: extract_card_member
Tool: import
Options:
----------------------------
reset.onemapper = false
codegen.output.delimiters.enclose = 0
sqlconnection.metadata.transaction.isolation.level = 2
codegen.input.delimiters.escape = 0
codegen.auto.compile.dir = true
accumulo.batch.size = 10240000
codegen.input.delimiters.field = 0
accumulo.create.table = false
mainframe.input.dataset.type = p
enable.compression = false
accumulo.max.latency = 5000
db.username = upgraduser
sqoop.throwOnError = false
db.clear.staging.table = false
codegen.input.delimiters.enclose = 0
hdfs.append.dir = false
import.direct.split.size = 0
hcatalog.drop.and.create.table = false
merge.key.col = card_id
codegen.output.delimiters.record = 10
codegen.output.delimiters.field = 44
hdfs.target.dir = /capstone_project/card_member
null.string = NA
hbase.bulk.load.enabled = false
db.password = ********
mapreduce.num.mappers = 4
export.new.update = UpdateOnly
db.require.password = false
hive.import = false
customtool.options.jsonmap = {}
hdfs.delete-target.dir = false
incremental.last.value = 2019-05-25 08:40:20.0
codegen.output.delimiters.enclose.required = false
direct.import = false
codegen.output.dir = .
hdfs.file.format = TextFile
incremental.col = member_joining_dt
hive.drop.delims = false
codegen.input.delimiters.record = 0
db.batch = false
null.non-string = \\N
split.limit = null
hcatalog.create.table = false
hive.fail.table.exists = false
hive.overwrite.table = false
incremental.mode = DateLastModified
temporary.dirRoot = _sqoop
verbose = false
import.max.inline.lob.size = 16777216
import.fetch.size = null
codegen.input.delimiters.enclose.required = false
relaxed.isolation = false
sqoop.oracle.escaping.disabled = true
db.table = card_member
hbase.create.table = false
codegen.compile.dir = /tmp/sqoop-ec2-user/compile/aef969e6cccea456c282118cfe51af3e
codegen.output.delimiters.escape = 0
db.connect.string = jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data
[ec2-user@ip-172-31-91-95 ~]$
```

sqoop job --show extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop



```
[ec2-user@ip-172-31-91-95 ~]$ sqoop job --show extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop
19/05/25 18:00:21 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.0
Job: extract_member_score
Tool: import
Options:
----------------------------
reset.onemapper = false
codegen.output.delimiters.enclose = 0
sqlconnection.metadata.transaction.isolation.level = 2
codegen.input.delimiters.escape = 0
codegen.auto.compile.dir = true
accumulo.batch.size = 10240000
codegen.input.delimiters.field = 0
accumulo.create.table = false
mainframe.input.dataset.type = p
enable.compression = false
accumulo.max.latency = 5000
db.username = upgraduser
sqoop.throwOnError = false
db.clear.staging.table = false
codegen.input.delimiters.enclose = 0
hdfs.append.dir = false
import.direct.split.size = 0
hcatalog.drop.and.create.table = false
codegen.output.delimiters.record = 10
codegen.output.delimiters.field = 44
hdfs.target.dir = /capstone_project/member_score
null.string = NA
hbase.bulk.load.enabled = false
db.password = ********
mapreduce.num.mappers = 4
export.new.update = UpdateOnly
db.require.password = false
hive.import = false
customtool.options.jsonmap = {}
hdfs.delete-target.dir = true
codegen.output.delimiters.enclose.required = false
direct.import = false
codegen.output.dir = .
hdfs.file.format = TextFile
hive.drop.delims = false
codegen.input.delimiters.record = 0
db.batch = false
null.non-string = \\N
split.limit = null
hcatalog.create.table = false
hive.fail.table.exists = false
hive.overwrite.table = false
incremental.mode = None
temporary.dirRoot = _sqoop
verbose = false
import.max.inline.lob.size = 16777216
import.fetch.size = null
codegen.input.delimiters.enclose.required = false
relaxed.isolation = false
sqoop.oracle.escaping.disabled = true
db.table = member_score
hbase.create.table = false
codegen.compile.dir = /tmp/sqoop-ec2-user/compile/cf04bf6d22f3213176e9783578d783be
codegen.output.delimiters.escape = 0
db.connect.string = jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com:3306/cred_financials_data
[ec2-user@ip-172-31-91-95 ~]$
```

================== Sqoop Commands: End =======================

================== OOZIE Setup: Start =======================

1. Update OOZIE shared library and copy various needed files so oozie workflow can execute sqoop and hive actions.

   (a) Switch to root user and then to hdfs user.

   sudo su –
   su – hdfs

   (b) Export OOZIE_URL.

   export OOZIE_URL=http://ip-172-31-91-95.ec2.internal:11000/oozie

   (c) Check oozie shared library for sqoop.

   oozie admin -shareliblist sqoop

(d) Start updating oozie shared library.

```
oozie admin -sharelibupdate
```

```
[ShareLib update status]
        sharelibDirOld = hdfs://ip-172-31-91-
95.ec2.internal:8020/user/oozie/share/lib/lib_20180731033840
        host = http://ec2-54-208-194-25.compute-1.amazonaws.com:11000/oozie
        sharelibDirNew = hdfs://ip-172-31-91-
95.ec2.internal:8020/user/oozie/share/lib/lib_20180731033840
        status = Successful
```

(e) Find mysql connector jar.

```
find / -name mysql*jar
```

(f) Above command found mysql connector jar at this location - /var/lib/oozie/mysql-connector-java.jar

(g) Copy mysql connector jar to oozie shared lib location for sqoop, change ownership to oozie and provide necessary permissions.

```
hadoop fs -put /var/lib/oozie/mysql-connector-java.jar
/user/oozie/share/lib/lib_20180731033840/sqoop/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/sqoop/mysql-connector-java.jar

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/sqoop/mysql-connector-java.jar
```

(h) Check oozie shared library for hive.

```
oozie admin -shareliblist hive
```

(i) Copy hive-site.xml to oozie shared lib location for hive, change ownership to oozie and provide necessary permissions.

```
hadoop fs -put /etc/hive/conf/hive-site.xml /user/oozie/share/lib/lib_20180731033840/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/hive/hive-site.xml

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/hive/hive-site.xml
```

(j) Copy hbase-site.xml to oozie shared lib location for hive, change ownership to oozie and provide necessary permissions.

```
hadoop fs -put /etc/hbase/conf/hbase-site.xml /user/oozie/share/lib/lib_20180731033840/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/hive/hbase-site.xml

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/hive/hbase-site.xml
```

(k) Copy metrics-core-2.2.0.jar to oozie shared lib location for hive, change ownership to oozie and provide necessary permissions.

```
hadoop fs -put /opt/cloudera/parcels/CDH/jars/metrics-core-2.2.0.jar
/user/oozie/share/lib/lib_20180731033840/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/hive/metrics-core-2.2.0.jar

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/hive/metrics-core-2.2.0.jar
```

(l) Copy hive-hbase-handler-1.1.0-cdh5.15.0.jar to oozie shared lib location for hive, change ownership to oozie and provide necessary permissions.

```
hadoop fs -put /opt/cloudera/parcels/CDH/jars/hive-hbase-handler-1.1.0-cdh5.15.0.jar
/user/oozie/share/lib/lib_20180731033840/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/hive/hive-hbase-handler-1.1.0-
cdh5.15.0.jar

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/hive/hive-hbase-handler-1.1.0-
cdh5.15.0.jar
```

(m) Copy all hbase related jars to oozie shared lib location for hive, change ownership to oozie and provide necessary permissions.

```
for i in `ls /opt/cloudera/parcels/CDH/jars/hbase* | grep -v test`; do hadoop fs -put $i
/user/oozie/share/lib/lib_20180731033840/hive/.; done

hadoop fs -chown oozie /user/oozie/share/lib/lib_20180731033840/hive/hbase*

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20180731033840/hive/hbase*
```

(n) Finish updating oozie shared library.

```
oozie admin -sharelibupdate

[ShareLib update status]
    sharelibDirOld = hdfs://ip-172-31-91-
95.ec2.internal:8020/user/oozie/share/lib/lib_20180731033840
    host = http://ec2-54-208-194-25.compute-1.amazonaws.com:11000/oozie
    sharelibDirNew = hdfs://ip-172-31-91-
95.ec2.internal:8020/user/oozie/share/lib/lib_20180731033840
    status = Successful
```

2. Update **sqoop-site.xml** (/etc/sqoop/conf/sqoop-site.xml). Add these properties within configuration tag.

```
<configuration>
    <property>
        <name>sqoop.metastore.client.autoconnect.url</name>
        <value>jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop</value>
        <description>The connect string to use when connecting to a
            job-management metastore. If unspecified, uses ~/.sqoop/.
            You can specify a different path here.
        </description>
    </property>

    <property>
        <name>sqoop.metastore.client.record.password</name>
        <value>true</value>
        <description>If true, allow saved passwords in the metastore.
        </description>
    </property>
</configuration>
```

3. Create directory in HDFS for oozie workflow using below command.

   ```
   hadoop fs -mkdir -p /capstone_project/oozie_workflow/app
   ```

4. Put sqoop-site.xml in oozie workflow application location.

   ```
   hadoop fs -put /etc/sqoop/conf/sqoop-site.xml /capstone_project/oozie_workflow/app/.
   ```

5. Put workflow.xml in oozie workflow application location.

   ```
   hadoop fs -put workflow.xml /capstone_project/oozie_workflow/app/.
   ```

   Below are the contents of **workflow.xml**:

```
<workflow-app name="capstone_project_wf" xmlns="uri:oozie:workflow:0.4">

  <start to="extract_card_member"/>

  <action name="extract_card_member">
    <sqoop xmlns="uri:oozie:sqoop-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <job-xml>sqoop-site.xml</job-xml>
      <configuration>
        <property>
          <name>fs.hdfs.impl.disable.cache</name>
          <value>true</value>
        </property>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
    </configuration>
      <command>job --exec extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-
95.ec2.internal:16000/sqoop</command>
    </sqoop>

    <ok to="extract_member_score"/>
    <error to="kill_node"/>
  </action>

  <action name="extract_member_score">
    <sqoop xmlns="uri:oozie:sqoop-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
```

```
      <name-node>${nameNode}</name-node>
      <job-xml>sqoop-site.xml</job-xml>
      <configuration>
        <property>
          <name>fs.hdfs.impl.disable.cache</name>
          <value>true</value>
        </property>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
    </configuration>
      <command>job --exec extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-172-31-91-95.ec2.internal:16000/sqoop</command>
    </sqoop>

    <ok to="lookup_data_refresh"/>
    <error to="kill_node"/>
  </action>

  <action name="lookup_data_refresh">
    <hive xmlns="uri:oozie:hive-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <script>${lookupScript}</script>
    </hive>

    <ok to="finish"/>
    <error to="kill_node"/>
  </action>

  <kill name="kill_node">
     <message>Your job failed!</message>
  </kill>

  <end name="finish"/>

</workflow-app>
```

(a) 3 action nodes have been setup in oozie workflow. 1st is a sqoop action to import card_member data incrementally from RDS into HDFS. 2nd is also a sqoop action, to import member_score data from RDS into HDFS. 3rd is a hive action to find out last 10 transactions, compute UCL and load look up table in hbase.

(b) I tried to use fork-join workflow so we can import card_member and member_score in parallel but it didn't work. Our cluster is a single node and resource manager has a fair scheduler. It runs jobs sequentially. I even tried to setup more queues and send different jobs to different queues but still didn't work. In order to run multiple jobs in parallel, resource manager should have capacity scheduler which is not available in our single node cluster. So finally, I had to setup import of card_member and member_score in a sequence.

6. Put lookupDataRefresh.hql in oozie workflow application location.

```
hadoop fs -put lookupDataRefresh.hql /capstone_project/oozie_workflow/app/.
```

Below are the hive commands in **lookupDataRefresh.hql**:

```
set hive.stats.autogather=true;
set hive.auto.convert.join=false;
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;

USE CAPSTONE_PROJECT;

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

INSERT OVERWRITE TABLE MEMBER_SCORE_ORC
SELECT MEMBER_ID, SCORE FROM MEMBER_SCORE_EXT;

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 DISTINCT CARD_ID, AMOUNT , POSTCODE , TRANSACTION_DT FROM CARD_TRANSACTIONS_HBASE
WHERE
STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;

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;

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;
```

(a) First set few parameters for hive session.
(b) Switch to use capstone_project.
(c) Load data in card_member_orc from card_member_ext;
(d) Load data in member_score_orc from member_score_ext;
(e) Load data in ranked_card_transactions_orc using same logic as explained earlier in task 3, point 3.
(f) Load data in card_ucl_orc using same logic as explained earlier in task 3, point 4.
(g) Load data in lookup_data_hbase using same logic as explained earlier in task 3, point 5.

7. Put coordinator.xml in oozie workflow location.

```
hadoop fs -put coordinator.xml /capstone_project/oozie_workflow/.
```

Below are the contents of **coordinator.xml**:

```
<coordinator-app name="capstone_proj_coord" start="${start}" end="${end}"
frequency="${coord:hours(4)}" timezone="UTC" xmlns="uri:oozie:coordinator:0.2">
        <controls>
                <timeout>5</timeout>
                <concurrency>1</concurrency>
                <execution>FIFO</execution>
                <throttle>5</throttle>
        </controls>
        <action>
                <workflow>
                        <app-path>${workflowpath}</app-path>
                        <configuration>
                                <property>
                                        <name>jobTracker</name>
                                        <value>${jobTracker}</value>
                                </property>
                                <property>
                                        <name>nameNode</name>
                                        <value>${nameNode}</value>
                                </property>
                                <property>
                                        <name>queueName</name>
                                        <value>${queueName}</value>
                                </property>
                        </configuration>
                </workflow>
        </action>
</coordinator-app>
```

================== **OOZIE Setup: End** =======================

================== **OOZIE Workflow Execution: Start** =======================

1. Copy job.properties.withoutcoordinator as job.properties.

```
cp job.properties.withoutcoordinator job.properties
```

Below are the contents of **job.properties.withoutcoordinator**:

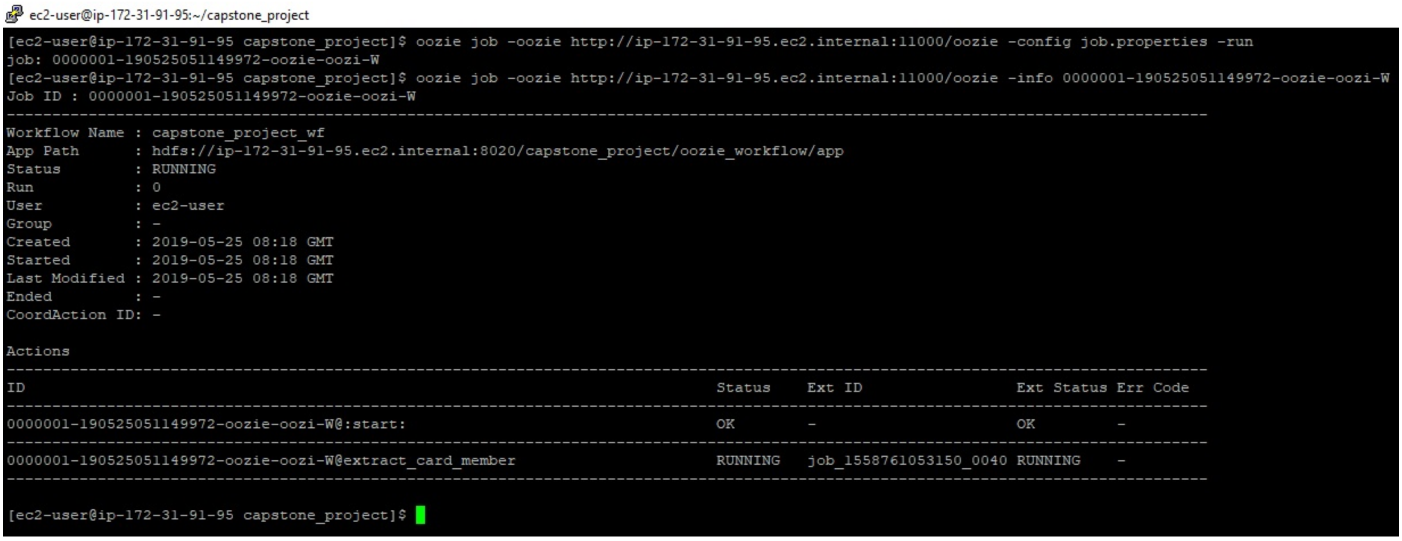
```
nameNode=hdfs://ip-172-31-91-95.ec2.internal:8020
jobTracker=ip-172-31-91-95.ec2.internal:8032
oozie.use.system.libpath=true
wfdir=${nameNode}/capstone_project/oozie_workflow
queueName=default
lookupScript=${wfdir}/app/lookupDataRefresh.hql
oozie.wf.application.path=${wfdir}/app
```

2. oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -config job.properties -run

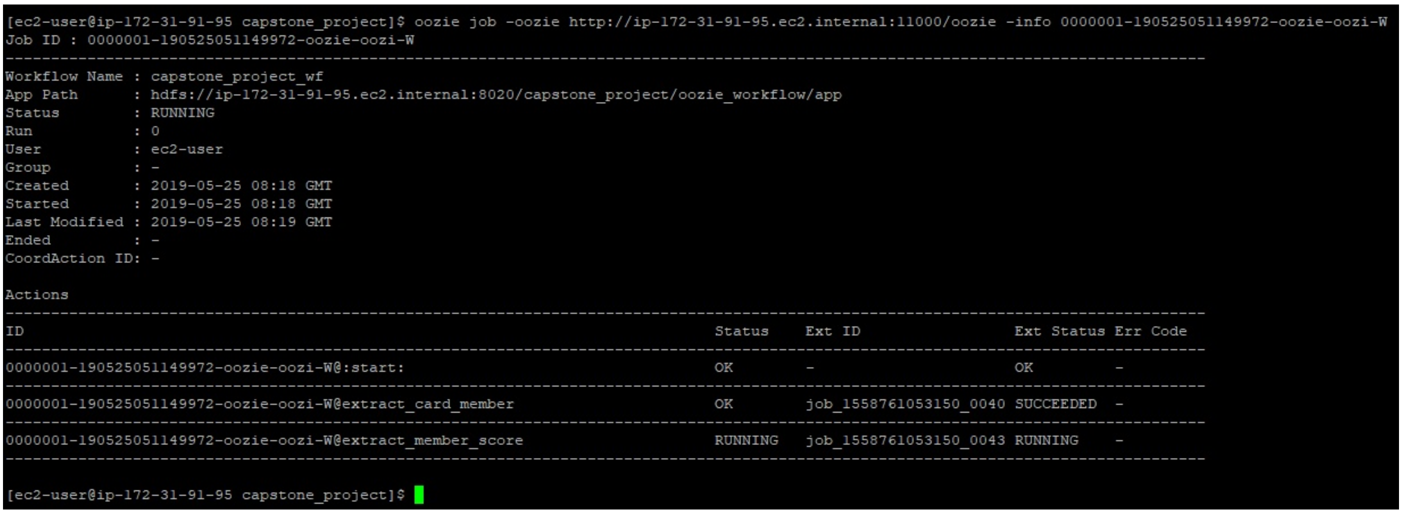
   oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -config job.properties -run

3. Wait for oozie job completion (job id was returned by previous command).

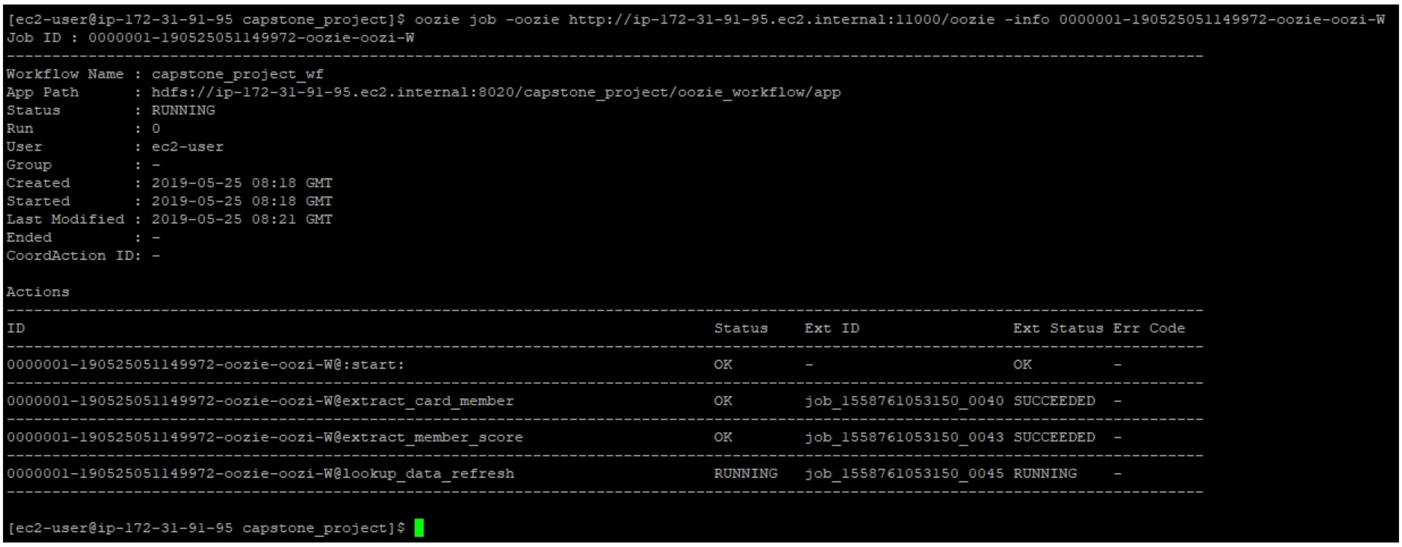
   oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000000-190525051149972-oozie-oozi-W



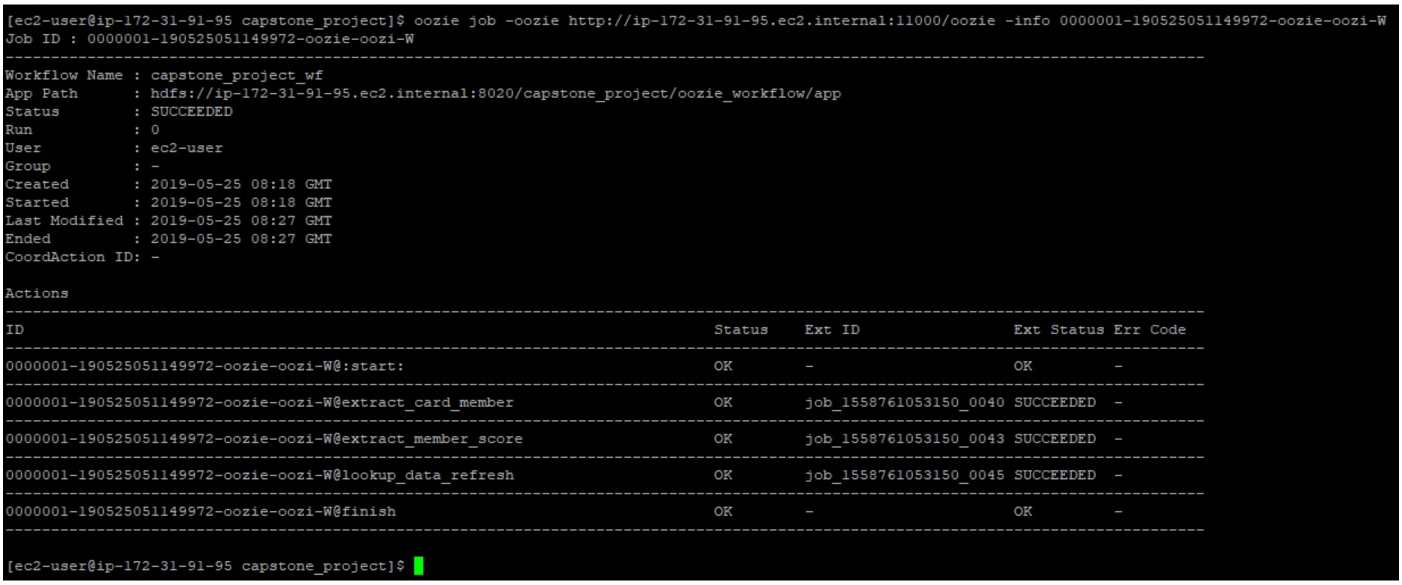
```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -config job.properties -run
job: 0000001-190525051149972-oozie-oozi-W
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000001-190525051149972-oozie-oozi-W
Job ID : 0000001-190525051149972-oozie-oozi-W
------------------------------------------------------------------------------------------------------------------------------------
Workflow Name : capstone_project_wf
App Path      : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/app
Status        : RUNNING
Run           : 0
User          : ec2-user
Group         : -
Created       : 2019-05-25 08:18 GMT
Started       : 2019-05-25 08:18 GMT
Last Modified : 2019-05-25 08:18 GMT
Ended         : -
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------------------------------------------
ID                                                                             Status    Ext ID                 Ext Status Err Code
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@:start:                                   OK        -                      OK         -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_card_member                       RUNNING   job_1558761053150_0040 RUNNING    -
------------------------------------------------------------------------------------------------------------------------------------

[ec2-user@ip-172-31-91-95 capstone_project]$
```



```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000001-190525051149972-oozie-oozi-W
Job ID : 0000001-190525051149972-oozie-oozi-W
------------------------------------------------------------------------------------------------------------------------------------
Workflow Name : capstone_project_wf
App Path      : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/app
Status        : RUNNING
Run           : 0
User          : ec2-user
Group         : -
Created       : 2019-05-25 08:18 GMT
Started       : 2019-05-25 08:18 GMT
Last Modified : 2019-05-25 08:19 GMT
Ended         : -
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------------------------------------------
ID                                                                             Status    Ext ID                 Ext Status Err Code
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@:start:                                   OK        -                      OK         -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_card_member                       OK        job_1558761053150_0040 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_member_score                      RUNNING   job_1558761053150_0043 RUNNING    -
------------------------------------------------------------------------------------------------------------------------------------

[ec2-user@ip-172-31-91-95 capstone_project]$
```



```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000001-190525051149972-oozie-oozi-W
Job ID : 0000001-190525051149972-oozie-oozi-W
------------------------------------------------------------------------------------------------------------------------------------
Workflow Name : capstone_project_wf
App Path      : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/app
Status        : RUNNING
Run           : 0
User          : ec2-user
Group         : -
Created       : 2019-05-25 08:18 GMT
Started       : 2019-05-25 08:18 GMT
Last Modified : 2019-05-25 08:21 GMT
Ended         : -
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------------------------------------------
ID                                                                             Status    Ext ID                 Ext Status Err Code
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@:start:                                   OK        -                      OK         -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_card_member                       OK        job_1558761053150_0040 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_member_score                      OK        job_1558761053150_0043 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@lookup_data_refresh                       RUNNING   job_1558761053150_0045 RUNNING    -
------------------------------------------------------------------------------------------------------------------------------------

[ec2-user@ip-172-31-91-95 capstone_project]$
```



```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000001-190525051149972-oozie-oozi-W
Job ID : 0000001-190525051149972-oozie-oozi-W
------------------------------------------------------------------------------------------------------------------------------------
Workflow Name : capstone_project_wf
App Path      : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/app
Status        : SUCCEEDED
Run           : 0
User          : ec2-user
Group         : -
Created       : 2019-05-25 08:18 GMT
Started       : 2019-05-25 08:18 GMT
Last Modified : 2019-05-25 08:27 GMT
Ended         : 2019-05-25 08:27 GMT
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------------------------------------------
ID                                                                             Status    Ext ID                 Ext Status Err Code
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@:start:                                   OK        -                      OK         -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_card_member                       OK        job_1558761053150_0040 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@extract_member_score                      OK        job_1558761053150_0043 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@lookup_data_refresh                       OK        job_1558761053150_0045 SUCCEEDED  -
------------------------------------------------------------------------------------------------------------------------------------
0000001-190525051149972-oozie-oozi-W@finish                                    OK        -                      OK         -
------------------------------------------------------------------------------------------------------------------------------------

[ec2-user@ip-172-31-91-95 capstone_project]$
```

4. Below are the contents of **job.properties**:

```
nameNode=hdfs://ip-172-31-91-95.ec2.internal:8020
jobTracker=ip-172-31-91-95.ec2.internal:8032
oozie.use.system.libpath=true
wfdir=${nameNode}/capstone_project/oozie_workflow
queueName=default
lookupScript=${wfdir}/app/lookupDataRefresh.hql

oozie.coord.application.path=${wfdir}/coordinator.xml
start=2019-05-25T08:40Z
end=2019-05-26T00:00Z
workflowpath=${wfdir}/app/workflow.xml
```

NOTE: Please change start time and end time appropriately whenever this is being executed for evaluation.

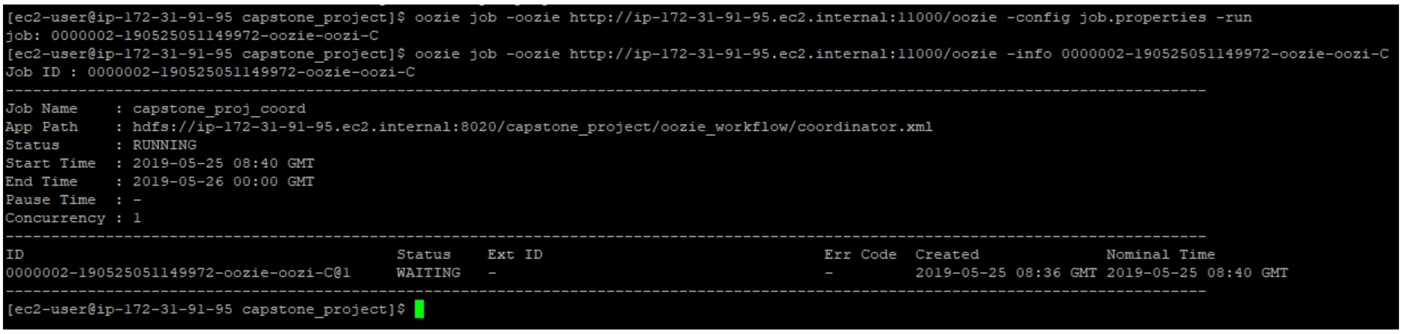
5. Run oozie job with coordinator.

```
oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -config job.properties -run
```

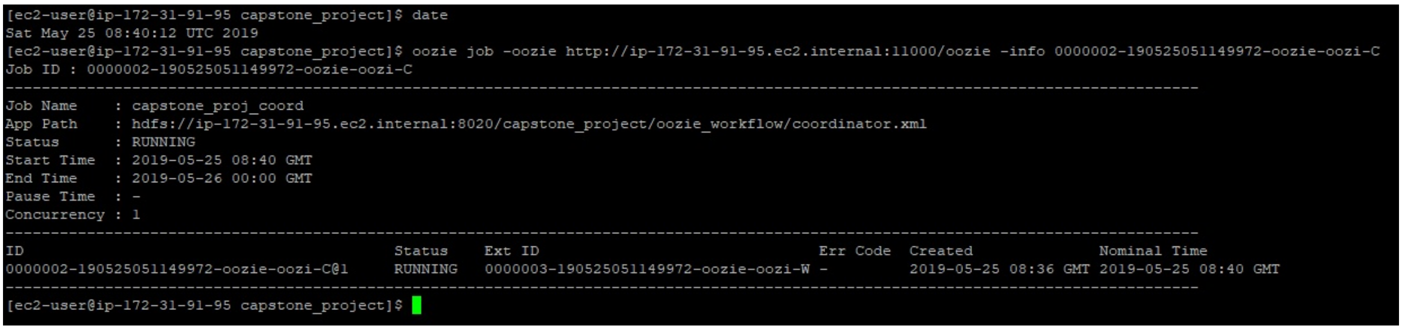
6. Verify oozie job (job id was returned by previous command).

   oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000002-190525051149972-oozie-oozi-C

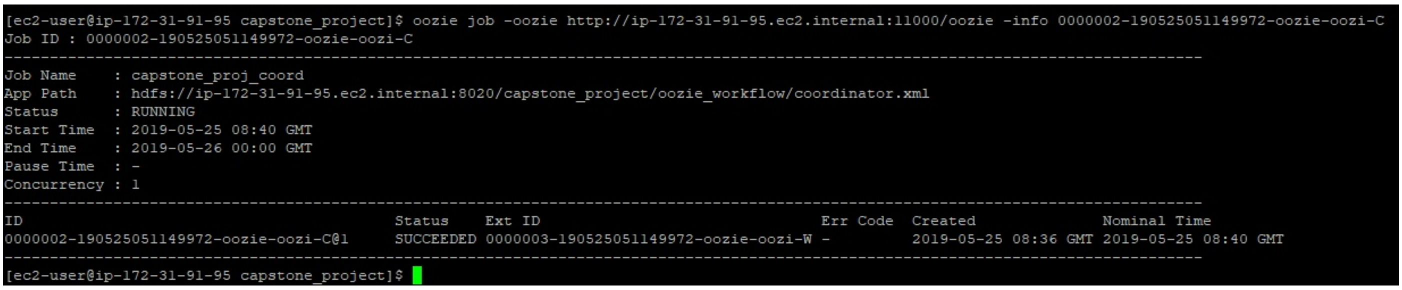
   NOTE: You will notice that there is a 'C' at the end of job id which tells us that it is a co-ordinator job.



```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -config job.properties -run
job: 0000002-190525051149972-oozie-oozi-C
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000002-190525051149972-oozie-oozi-C
Job ID : 0000002-190525051149972-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 08:40 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                          Status    Ext ID                                 Err Code  Created              Nominal Time
0000002-190525051149972-oozie-oozi-C@1      WAITING   -                                      -         2019-05-25 08:36 GMT 2019-05-25 08:40 GMT
------------------------------------------------------------------------------------------------------------------------------------
[ec2-user@ip-172-31-91-95 capstone_project]$
```

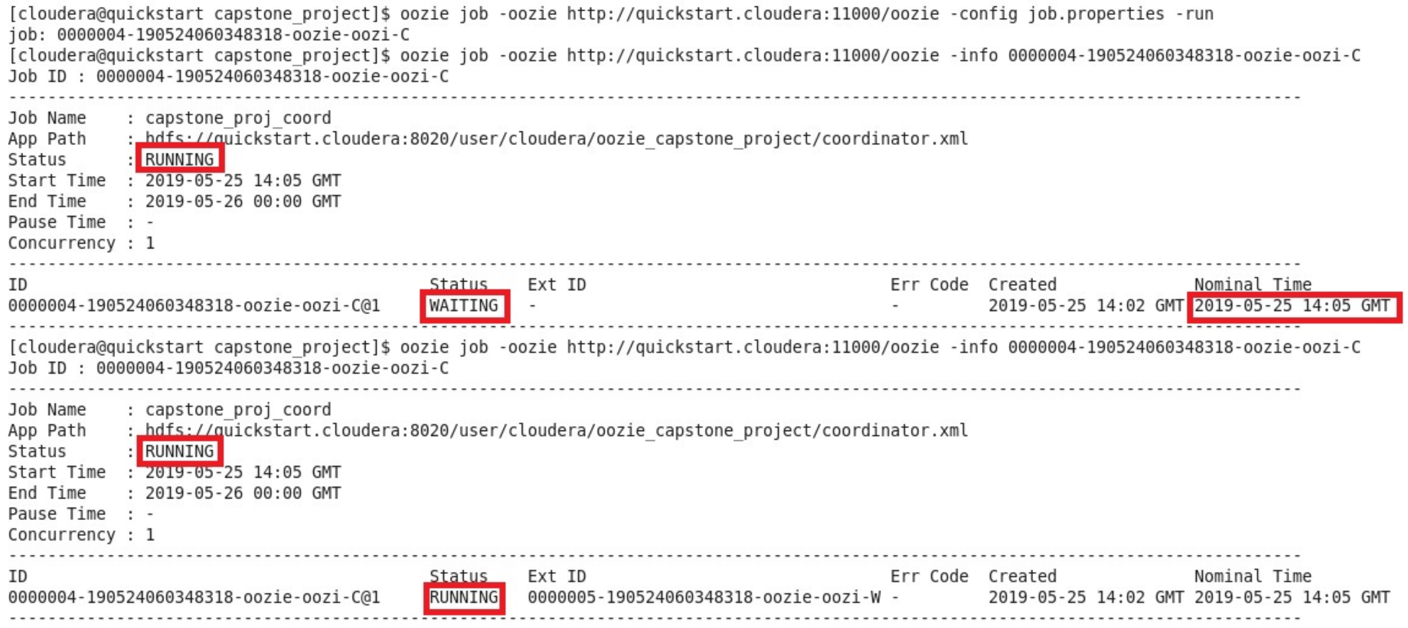


```
[ec2-user@ip-172-31-91-95 capstone_project]$ date
Sat May 25 08:40:12 UTC 2019
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000002-190525051149972-oozie-oozi-C
Job ID : 0000002-190525051149972-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 08:40 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                          Status    Ext ID                                 Err Code  Created              Nominal Time
0000002-190525051149972-oozie-oozi-C@1      RUNNING   0000003-190525051149972-oozie-oozi-W -         2019-05-25 08:36 GMT 2019-05-25 08:40 GMT
------------------------------------------------------------------------------------------------------------------------------------
[ec2-user@ip-172-31-91-95 capstone_project]$
```



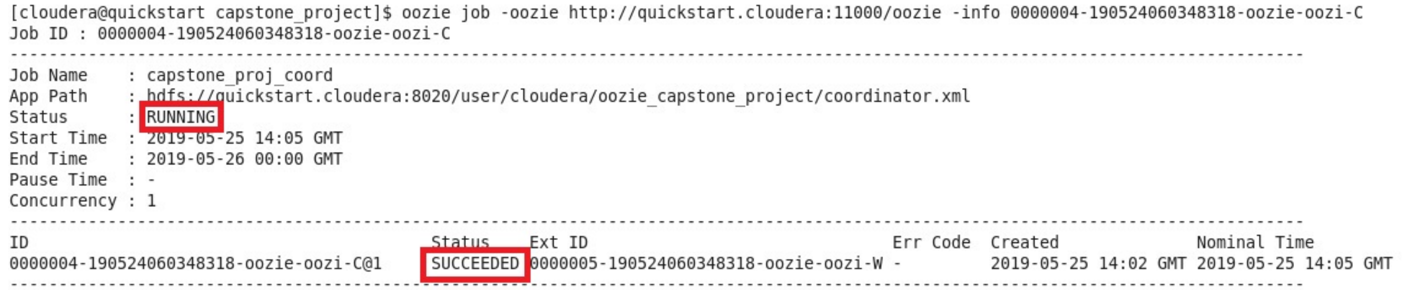
```
[ec2-user@ip-172-31-91-95 capstone_project]$ oozie job -oozie http://ip-172-31-91-95.ec2.internal:11000/oozie -info 0000002-190525051149972-oozie-oozi-C
Job ID : 0000002-190525051149972-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://ip-172-31-91-95.ec2.internal:8020/capstone_project/oozie_workflow/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 08:40 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                          Status    Ext ID                                 Err Code  Created              Nominal Time
0000002-190525051149972-oozie-oozi-C@1      SUCCEEDED 0000003-190525051149972-oozie-oozi-W -         2019-05-25 08:36 GMT 2019-05-25 08:40 GMT
------------------------------------------------------------------------------------------------------------------------------------
[ec2-user@ip-172-31-91-95 capstone_project]$
```

After 4 hours (before 1 minute), a new job will come in WAITING status and will start RUNNING after exactly 4 hours from the start time. I deliberately did not test this in ec2 instance because I was supposed to keep up the instance for another 4 hours to see that execution. It would have incurred more charges. So, in order to save cost associated with use of ec2 instance, I tested this part in cloudera VM. Everything is same there. I just change IP address to quickstart.cloudera, wherever was needed. Below are the screenshots from cloudera VM just to prove that I have setup a job successfully using oozie coordinator which will run every 4 hours.



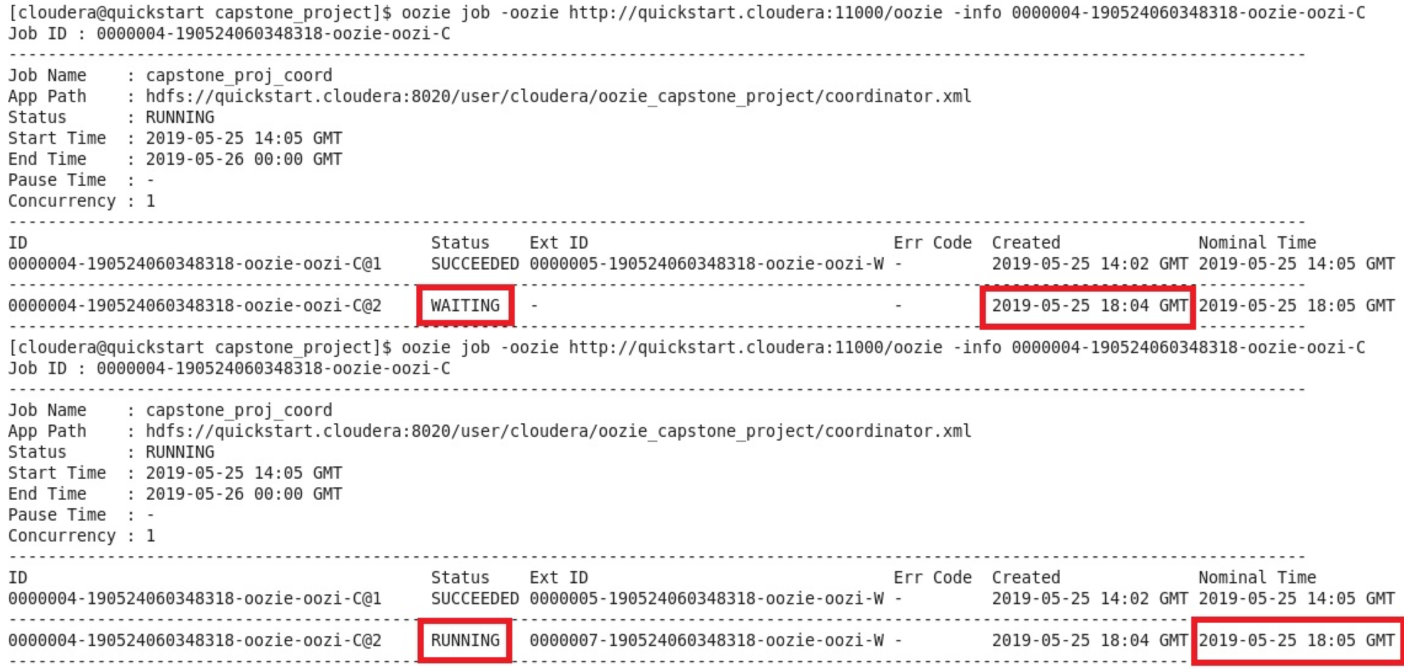
```
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -config job.properties -run
job: 0000004-190524060348318-oozie-oozi-C
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                         Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1     WAITING   -                                    -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                         Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1     RUNNING   0000005-190524060348318-oozie-oozi-W -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
```

```
============================================================================================================================================
```



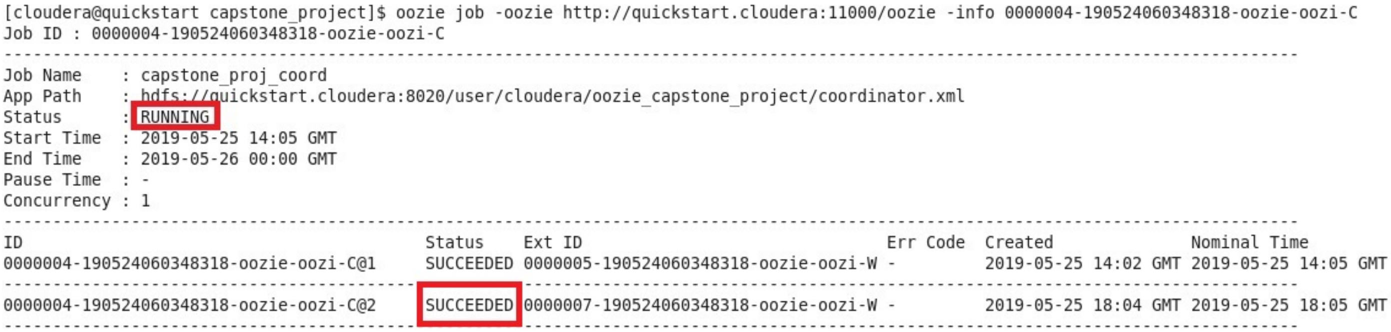
```
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                         Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1     SUCCEEDED 0000005-190524060348318-oozie-oozi-W -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
```

```
============================================================================================================================================
```



```
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                         Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1     SUCCEEDED 0000005-190524060348318-oozie-oozi-W -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
0000004-190524060348318-oozie-oozi-C@2     WAITING   -                                    -         2019-05-25 18:04 GMT 2019-05-25 18:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                         Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1     SUCCEEDED 0000005-190524060348318-oozie-oozi-W -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
0000004-190524060348318-oozie-oozi-C@2     RUNNING   0000007-190524060348318-oozie-oozi-W -         2019-05-25 18:04 GMT 2019-05-25 18:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
```

```
============================================================================================================================================
```



```
[cloudera@quickstart capstone_project]$ oozie job -oozie http://quickstart.cloudera:11000/oozie -info 0000004-190524060348318-oozie-oozi-C
Job ID : 0000004-190524060348318-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name    : capstone_proj_coord
App Path    : hdfs://quickstart.cloudera:8020/user/cloudera/oozie_capstone_project/coordinator.xml
Status      : RUNNING
Start Time  : 2019-05-25 14:05 GMT
End Time    : 2019-05-26 00:00 GMT
Pause Time  : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                        Status    Ext ID                               Err Code  Created              Nominal Time
0000004-190524060348318-oozie-oozi-C@1    SUCCEEDED 0000005-190524060348318-oozie-oozi-W -         2019-05-25 14:02 GMT 2019-05-25 14:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
0000004-190524060348318-oozie-oozi-C@2    SUCCEEDED 0000007-190524060348318-oozie-oozi-W -         2019-05-25 18:04 GMT 2019-05-25 18:05 GMT
------------------------------------------------------------------------------------------------------------------------------------
```

================== OOZIE Workflow Execution: End =======================

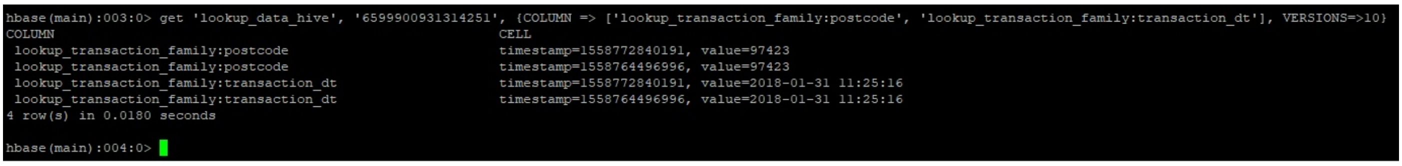
================== HBase Commands: Start ======================

1. Once oozie workflow is successfully setup and executed, you can check data in HBase lookup_data_hive table using below command:

   scan 'lookup_data_hive', {VERSIONS=>10}

2. Check data for a particular card_id, see multiple versions for postcode and transaction_dt.

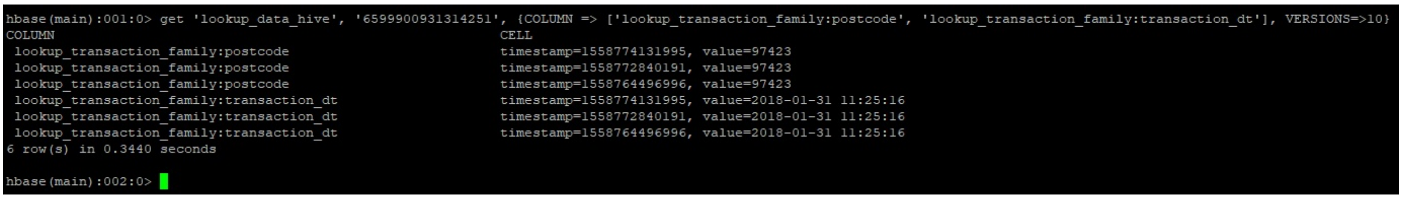
   get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_transaction_family:postcode', 'lookup_transaction_family:transaction_dt'], VERSIONS=>10}
   Below is the screenshot after I have run oozie workflow without coordinator:



```
hbase(main):003:0> get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_transaction_family:postcode', 'lookup_transaction_family:transaction_dt'], VERSIONS=>10}
COLUMN                                                      CELL
 lookup_transaction_family:postcode                         timestamp=1558772840191, value=97423
 lookup_transaction_family:postcode                         timestamp=1558764496996, value=97423
 lookup_transaction_family:transaction_dt                   timestamp=1558772840191, value=2018-01-31 11:25:16
 lookup_transaction_family:transaction_dt                   timestamp=1558764496996, value=2018-01-31 11:25:16
4 row(s) in 0.0180 seconds

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

You will notice that there are 2 versions of postcode and transaction_dt in lookup_transaction_family. 1st version came when I loaded the data first time in task 3. And 2nd version came when I loaded the data through oozie workflow without coordinator. Colum family lookup_transaction_family is set to have 10 VERSIONS. Data has not changed because we have not setup streaming layer yet and no changes have happened in card_member and member_score data as well.

Below is the screenshot after I have run oozie workflow with coordinator after its 1st successful execution:



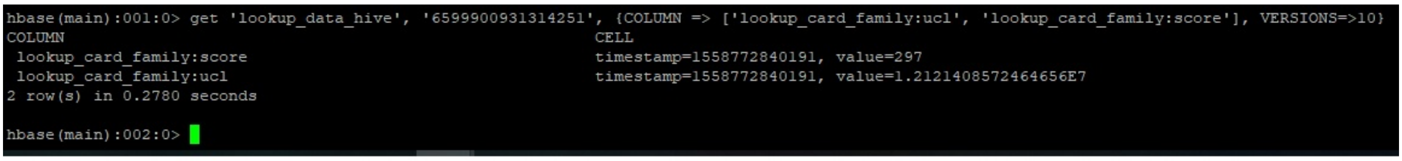
```
hbase(main):001:0> get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_transaction_family:postcode', 'lookup_transaction_family:transaction_dt'], VERSIONS=>10}
COLUMN                                                      CELL
 lookup_transaction_family:postcode                         timestamp=1558774131995, value=97423
 lookup_transaction_family:postcode                         timestamp=1558772840191, value=97423
 lookup_transaction_family:postcode                         timestamp=1558764496996, value=97423
 lookup_transaction_family:transaction_dt                   timestamp=1558774131995, value=2018-01-31 11:25:16
 lookup_transaction_family:transaction_dt                   timestamp=1558772840191, value=2018-01-31 11:25:16
 lookup_transaction_family:transaction_dt                   timestamp=1558764496996, value=2018-01-31 11:25:16
6 row(s) in 0.3440 seconds

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

You will notice that there are now 3 versions of postcode and transaction_dt in lookup_transaction_family. 3rd version came when I loaded the data through oozie workflow with coordinator. Colum family lookup_transaction_family is set to have 10 VERSIONS. Data has not changed because we have not setup streaming layer yet and no changes have happened in card_member and member_score data as well.

3. Check data for a particular card_id, verify that there should not be any multiple versions for ucl and score.

get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_card_family:ucl', 'lookup_card_family:score'], VERSIONS=>10}

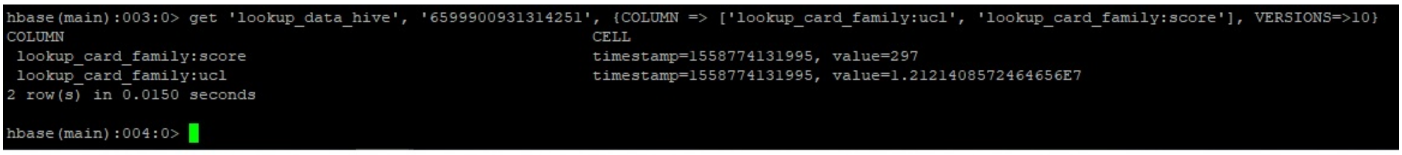
Below is the screenshot after I have run oozie workflow without coordinator:



```
hbase(main):001:0> get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_card_family:ucl', 'lookup_card_family:score'], VERSIONS=>10}
COLUMN                                                      CELL
 lookup_card_family:score                                   timestamp=1558772840191, value=297
 lookup_card_family:ucl                                     timestamp=1558772840191, value=1.2121408572464656E7
2 row(s) in 0.2780 seconds

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

You will notice that there is only 1 version of ucl and score in card_lookup_family as it is set to have only 1 VERSION.

Below is the screenshot after I have run oozie workflow with coordinator after its 1st successful execution:



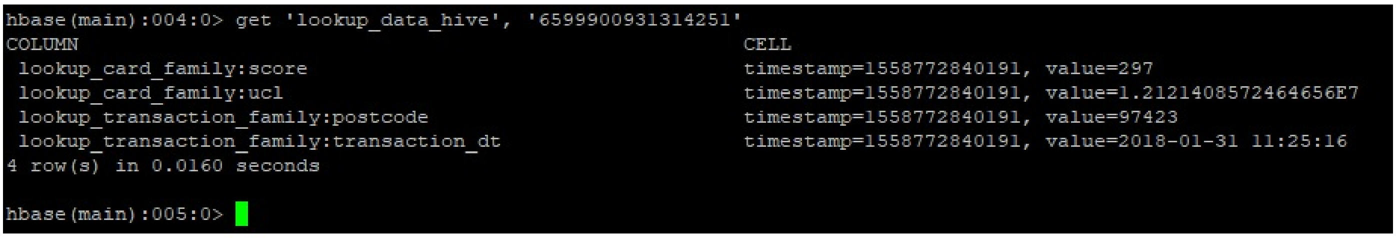
```
hbase(main):003:0> get 'lookup_data_hive', '6599900931314251', {COLUMN => ['lookup_card_family:ucl', 'lookup_card_family:score'], VERSIONS=>10}
COLUMN                                                      CELL
 lookup_card_family:score                                   timestamp=1558774131995, value=297
 lookup_card_family:ucl                                     timestamp=1558774131995, value=1.2121408572464656E7
2 row(s) in 0.0150 seconds

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

You will notice that there is still only 1 version of ucl and score in card_lookup_family as it is set to have only 1 VERSION. Please notice the change in timestamp from previous screenshot. Data has not changed because we have not setup streaming layer yet and no changes have happened in card_member and member_score data as well.

4. Check data for a particular card_id. This command shows only 1 version (latest) of the data.

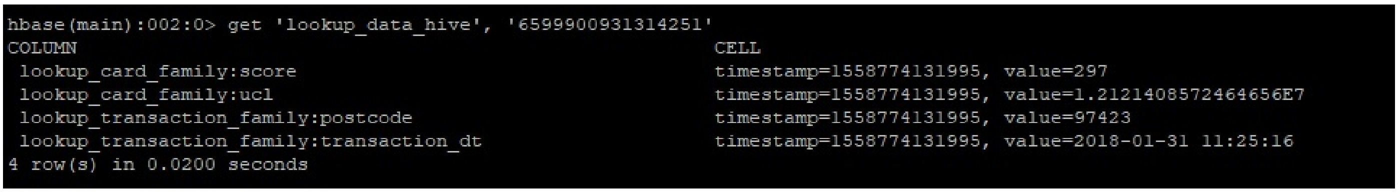
   get 'lookup_data_hive', '6599900931314251'

   Below is the screenshot after I have run oozie workflow without coordinator:



```
hbase(main):004:0> get 'lookup_data_hive', '6599900931314251'
COLUMN                                                      CELL
 lookup_card_family:score                                   timestamp=1558772840191, value=297
 lookup_card_family:ucl                                     timestamp=1558772840191, value=1.2121408572464656E7
 lookup_transaction_family:postcode                         timestamp=1558772840191, value=97423
 lookup_transaction_family:transaction_dt                   timestamp=1558772840191, value=2018-01-31 11:25:16
4 row(s) in 0.0160 seconds

hbase(main):005:0>
```

Below is the screenshot after I have run oozie workflow with coordinator after its 1st successful execution:



```
hbase(main):002:0> get 'lookup_data_hive', '6599900931314251'
COLUMN                                                      CELL
 lookup_card_family:score                                   timestamp=1558774131995, value=297
 lookup_card_family:ucl                                     timestamp=1558774131995, value=1.2121408572464656E7
 lookup_transaction_family:postcode                         timestamp=1558774131995, value=97423
 lookup_transaction_family:transaction_dt                   timestamp=1558774131995, value=2018-01-31 11:25:16
4 row(s) in 0.0200 seconds
```

Please notice the change in timestamp from previous screenshot. Data has not changed because we have not setup streaming layer yet and no changes have happened in card_member and member_score data as well.

================== HBase Commands: End ======================

==================================Mid-Submission – END==