Initial Setup:

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
hdfs dfs -mkdir /user/ec2-user/capstone
[hdfs@ip-10-0-0-183 ~]$ hdfs dfs -chown -R ec2-user:ec2-user /user/ec2-user/capstone
```

Thereafter create two sub-directories **card_member** and **member_score** within capstone

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -mkdir /user/ec2-user/capstone/card_member
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -mkdir /user/ec2-user/capstone/member_score
```

Download card_transactions and zipCodePosId csv's from resources section in the capstone project and transfer it to ec2 instance via WinSCP. Copy both the files to HDFS on the location created above.

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -copyFromLocal *.csv /user/ec2-user/capstone
```

```
[ec2-user@ip-10-0-0-243 ~]$ ls -l
total 7380
-rw-rw-r-- 1 ec2-user ec2-user   20202 Feb 12 17:54 card_member.java
-rw-rw-r-- 1 ec2-user ec2-user 4829520 Feb  5 11:30 card_transactions.csv
-rw-rw-r-- 1 ec2-user ec2-user 1673036 Aug 24 10:08 Key_indicator_districtwise.java
-rw-rw-r-- 1 ec2-user ec2-user   10985 Feb 12 18:06 member_score.java
drwxrwxr-x 3 ec2-user ec2-user      27 Jun 21  2019 spark_assignment
-rw-rw-r-- 1 ec2-user ec2-user 1013345 Feb  5 12:01 zipCodePosId.csv
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -copyFromLocal *.csv /user/ec2-user/capstone
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -chmod -R 777 /user/ec2-user/capstone
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -ls /user/ec2-user/capstone
Found 4 items
drwxrwxrwx   - ec2-user ec2-user          0 2020-02-15 02:51 /user/ec2-user/capstone/card_member
-rwxrwxrwx   3 ec2-user ec2-user    4829520 2020-02-15 04:47 /user/ec2-user/capstone/card_transactions.csv
drwxrwxrwx   - ec2-user ec2-user          0 2020-02-15 03:17 /user/ec2-user/capstone/member_score
-rwxrwxrwx   3 ec2-user ec2-user    1013345 2020-02-15 04:47 /user/ec2-user/capstone/zipCodePosId.csv
```

Sqoop command for ingesting card_member and member_score from AWS to HDFS.

```
[ec2-user@ip-10-0-0-243 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.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 '/user/ec2-user/capstone/card_member'
```

```
[ec2-user@ip-10-0-0-243 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.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 '/user/ec2-user/capstone/member_score'
```

```
[ec2-user@ip-10-0-0-243 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.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 '/user/ec2-user/capstone/card_member'
Warning: /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/bin/../lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/02/15 02:50:54 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
20/02/15 02:50:54 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/02/15 02:50:54 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/02/15 02:50:54 INFO tool.CodeGenTool: Beginning code generation
20/02/15 02:50:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
20/02/15 02:50:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
20/02/15 02:50:55 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
Note: /tmp/sqoop-ec2-user/compile/1b4ac0d765759c57ebde25bea360fe5a/card_member.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/02/15 02:50:57 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-ec2-user/compile/1b4ac0d765759c57ebde25bea360fe5a/card_member.jar
20/02/15 02:50:58 INFO tool.ImportTool: Destination directory /user/ec2-user/capstone/card_member deleted.
20/02/15 02:50:58 WARN manager.MySQLManager: It looks like you are importing from mysql.
20/02/15 02:50:58 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/02/15 02:50:58 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/02/15 02:50:58 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/02/15 02:50:58 INFO mapreduce.ImportJobBase: Beginning import of card_member
20/02/15 02:50:58 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/02/15 02:50:58 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/02/15 02:50:58 INFO client.RMProxy: Connecting to ResourceManager at ip-10-0-0-243.ec2.internal/10.0.0.243:8032
20/02/15 02:51:05 INFO db.DBInputFormat: Using read commited transaction isolation
20/02/15 02:51:05 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`card_id`), MAX(`card_id`) FROM `card_member`
20/02/15 02:51:05 WARN db.TextSplitter: Generating splits for a textual index column.
20/02/15 02:51:05 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.
20/02/15 02:51:05 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.
20/02/15 02:51:05 INFO mapreduce.JobSubmitter: number of splits:6
20/02/15 02:51:06 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1581734636733_0001
20/02/15 02:51:06 INFO impl.YarnClientImpl: Submitted application application_1581734636733_0001
20/02/15 02:51:06 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-243.ec2.internal:8088/proxy/application_1581734636733_0001/
20/02/15 02:51:06 INFO mapreduce.Job: Running job: job_1581734636733_0001
20/02/15 02:51:15 INFO mapreduce.Job: Job job_1581734636733_0001 running in uber mode : false
20/02/15 02:51:15 INFO mapreduce.Job:  map 0% reduce 0%
20/02/15 02:51:23 INFO mapreduce.Job:  map 33% reduce 0%
20/02/15 02:51:25 INFO mapreduce.Job:  map 50% reduce 0%
20/02/15 02:51:31 INFO mapreduce.Job:  map 67% reduce 0%
20/02/15 02:51:32 INFO mapreduce.Job:  map 83% reduce 0%
20/02/15 02:51:33 INFO mapreduce.Job:  map 100% reduce 0%
```

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -ls /user/ec2-user/capstone/card_member
Found 7 items
-rw-r--r--   3 ec2-user ec2-user          0 2020-02-15 02:51 /user/ec2-user/capstone/card_member/_SUCCESS
-rw-r--r--   3 ec2-user ec2-user          0 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00000
-rw-r--r--   3 ec2-user ec2-user      23080 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00001
-rw-r--r--   3 ec2-user ec2-user      20684 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00002
-rw-r--r--   3 ec2-user ec2-user      19608 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00003
-rw-r--r--   3 ec2-user ec2-user      21624 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00004
-rw-r--r--   3 ec2-user ec2-user         86 2020-02-15 02:51 /user/ec2-user/capstone/card_member/part-m-00005
```

```
[ec2-user@ip-10-0-0-243 ~]$ clear
[ec2-user@ip-10-0-0-243 ~]$ sqoop import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.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 '/user/ec2-user/capstone/member_score'
Warning: /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/bin/../lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/02/15 03:16:44 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
20/02/15 03:16:44 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/02/15 03:16:44 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/02/15 03:16:44 INFO tool.CodeGenTool: Beginning code generation
20/02/15 03:16:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
20/02/15 03:16:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
20/02/15 03:16:44 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
Note: /tmp/sqoop-ec2-user/compile/1f02c67efcc213cf9e483670b818aeed/member_score.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/02/15 03:16:46 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-ec2-user/compile/1f02c67efcc213cf9e483670b818aeed/member_score.jar
20/02/15 03:16:47 INFO tool.ImportTool: Destination directory /user/ec2-user/capstone/member_score deleted.
20/02/15 03:16:47 WARN manager.MySQLManager: It looks like you are importing from mysql.
20/02/15 03:16:47 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/02/15 03:16:47 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/02/15 03:16:47 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/02/15 03:16:47 INFO mapreduce.ImportJobBase: Beginning import of member_score
20/02/15 03:16:47 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/02/15 03:16:47 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/02/15 03:16:47 INFO client.RMProxy: Connecting to ResourceManager at ip-10-0-0-243.ec2.internal/10.0.0.243:8032
20/02/15 03:16:50 INFO db.DBInputFormat: Using read commited transaction isolation
20/02/15 03:16:50 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`member_id`), MAX(`member_id`) FROM `member_score`
20/02/15 03:16:50 WARN db.TextSplitter: Generating splits for a textual index column.
20/02/15 03:16:50 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.
20/02/15 03:16:50 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.
20/02/15 03:16:51 INFO mapreduce.JobSubmitter: number of splits:6
20/02/15 03:16:51 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1581734636733_0002
20/02/15 03:16:52 INFO impl.YarnClientImpl: Submitted application application_1581734636733_0002
20/02/15 03:16:52 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-243.ec2.internal:8088/proxy/application_1581734636733_0002/
20/02/15 03:16:52 INFO mapreduce.Job: Running job: job_1581734636733_0002
20/02/15 03:16:58 INFO mapreduce.Job: Job job_1581734636733_0002 running in uber mode : false
20/02/15 03:16:58 INFO mapreduce.Job:  map 0% reduce 0%
20/02/15 03:17:06 INFO mapreduce.Job:  map 33% reduce 0%
20/02/15 03:17:07 INFO mapreduce.Job:  map 50% reduce 0%
20/02/15 03:17:13 INFO mapreduce.Job:  map 67% reduce 0%
20/02/15 03:17:14 INFO mapreduce.Job:  map 100% reduce 0%
```

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -ls /user/ec2-user/capstone/member_score
Found 7 items
-rw-r--r--   3 ec2-user ec2-user          0 2020-02-15 03:17 /user/ec2-user/capstone/member_score/_SUCCESS
-rw-r--r--   3 ec2-user ec2-user          0 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00000
-rw-r--r--   3 ec2-user ec2-user       5920 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00001
-rw-r--r--   3 ec2-user ec2-user       4220 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00002
-rw-r--r--   3 ec2-user ec2-user       4360 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00003
-rw-r--r--   3 ec2-user ec2-user       5460 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00004
-rw-r--r--   3 ec2-user ec2-user         20 2020-02-15 03:17 /user/ec2-user/capstone/member_score/part-m-00005
```

Read, write and execute permission given on capstone.

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -chmod -R 777 /user/ec2-user/capstone
```

```
[ec2-user@ip-10-0-0-243 ~]$ hdfs dfs -ls /user/ec2-user/capstone
Found 2 items
drwxrwxrwx   - ec2-user ec2-user          0 2020-02-15 02:51 /user/ec2-user/capstone/card_member
drwxrwxrwx   - ec2-user ec2-user          0 2020-02-15 03:17 /user/ec2-user/capstone/member_score
```

SECTION: 1 Script to load the data and create table/s in the NoSQL database. This includes all commands from file LoadCreateNoSQL.txt

Script to load data and create table/s in the No-sql database

a. **First create new database namely credit_card_fraud_detection**

```
CREATE database credit_card_fraud_detection;
USE credit_card_fraud_detection;
SHOW credit_card_fraud_detection;
```

```
1 show databases;
```

```
INFO  : Executing command(queryId=hive_20200215043939_6b298441-b317-4e8a-ab74-c83af6c5a45a): show databases
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215043939_6b298441-b317-4e8a-ab74-c83af6c5a45a); Time taken: 0.005 seconds
INFO  : OK
```

Query History | Saved Queries | Results (2)

	database_name
1	credit_card_fraud_detection
2	default

b. Set some parameters for 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;
```

```
✔ Success.
```

c. Create external table card_transactions_ext table which will point to HDFS location

```
CREATE EXTERNAL TABLE IF NOT EXISTS CARD_TRANSACTIONS_EXT(
'CARD_ID' STRING,
'MEMBER_ID' STRING,
'AMOUNT' DOUBLE,
'POSTCODE' STRING,
'POS_ID' STRING,
'TRANSACTION_DT' STRING,
'STATUS' STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/ec2-user/capstone/card_transactions'
TBLPROPERTIES ("skip.header.line.count"="1");
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS CARD_TRANSACTIONS_EXT(
`CARD_ID` STRING,
`MEMBER_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`POS_ID` STRING,
`TRANSACTION_DT` STRING,
`STATUS` STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/ec2-user/capstone/card_transactions'
TBLPROPERTIES ("skip.header.line.count"="1");
```

```
TBLPROPERTIES ("skip.header.line.count"="1")
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215045151_9ff0da64-a895-4331-9653-ff516078fd94); Time taken: 0.044 se
conds
INFO  : OK
```

```
✔ Success.
```

d. Create table card_transactions_orc

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

```
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");
```

```
TBLPROPERTIES ( orc.compress = SNAPPY )
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215045353_f295db79-26a5-4a6f-ad8f-04242537c852); Time taken: 0.262 se
conds
INFO  : OK
```

Success.

e. Load data in card_transaction_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;
```

```
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')
FROM CARD_TRANSACTIONS_EXT;
```

```
INFO  : Stage-Stage-1: Map: 1  Cumulative CPU: 4.53 sec   HDFS Read: 5954 HDFS Write: 164 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 4 seconds 530 msec
INFO  : Completed executing command(queryId=hive_20200215045555_09fe76f6-7263-4fb8-bcce-8bd6c0b2a0b2); Time taken: 25.327 s
econds
INFO  : OK
```

job_1581737473027_0005

Success.

f. Verify transaction_dt and year in card_transactions_orc

```
select
      year(transaction_dt),
      transaction_dt
From card_transactions_orc limit 10;
```

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

```
INFO  : Completed executing command(queryId=hive_20200215050909_7b6f3882-9988-426c-a1f2-b8d05
econds
INFO  : OK
```

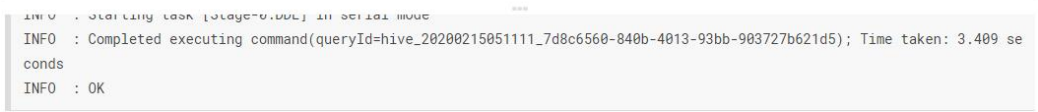
job_1581737473027_0009

Query History | Saved Queries | Results (10)

	_c0	transaction_dt
1	2018	2018-02-11 00:00:00.0
2	2018	2018-02-11 00:00:00.0
3	2018	2018-02-11 00:00:00.0
4	2018	2018-02-11 00:00:00.0
5	2018	2018-02-11 00:00:00.0
6	2018	2018-02-11 00:00:00.0
7	2018	2018-02-11 00:00:00.0

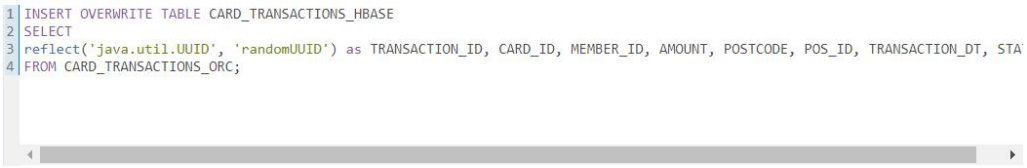
g. Create card_transactions_hbase as hive-hbase integrated table which will be visible from Hbase as well.

```
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");
```



h. Load data in card_transactions_hbase.

```
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;
```



i. Check for some data in card_transactions_hbase

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

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

```
INFO  : Completed executing command(queryId=hive_20200215051515_a6d74899-f4ae-4ca3-b3c5-339eeec6e900); Time taken: 0.001 se
conds
INFO  : OK
```

Query History | Saved Queries | Results (10)

	card_transactions_hbase.transaction_id	card_transactions_hbase.card_id	card_transactions_hbase.member_i
1	00006c54-05dd-452d-8e41-232eaeabba5b	340082915339645	512969555857346
2	00019b41-1432-4f95-a2b0-3124ffa19d28	348413196172048	001739553947511
3	0002ddb6-fa2d-45cf-add1-f82f86dae893	4851468805032068	493625663564055
4	00037835-94d6-4407-9eb0-8402fc98640d	4782879464621468	257134899293254
5	00039c97-79ee-4819-ad69-ea4bd5f31892	6510010051133634	162601897371597
6	0003ce3f-609c-4285-90f5-b094a04d0a36	5147189362741898	384113677556249
7	00052ca2-a307-40a1-9ceb-c0cf55a7ca64	5318585357625843	745833061184829

j. Create lookup_data_hbase as hive-hbase integrated table.

```
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");
```

```
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_fa
TBLPROPERTIES ("hbase.table.name" = "lookup_data_hive");
```

```
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215051717_a62ec867-e58c-4184-b940-9f160adc3697); Time taken: 1.422 se
conds
INFO  : OK
```

✔ Success.

k. In Hbase check details of card_transaction_hive

```
describe 'card_transactions_hive'
```

```
[ec2-user@ip-10-0-0-243 ~]$ hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
20/02/15 05:19:52 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.15.1, rUnknown, Thu Aug  9 09:07:41 PDT 2018

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 0.6370 seconds
```

l. Check count in card_transactions_hive

```
count 'card_transactions_hive'
```

```
hbase(main):002:0> count 'card_transactions_hive'
Current count: 1000, row: 04f7eaa7-7898-4777-905c-1d5491e5b5f3
Current count: 2000, row: 09a1ef8d-cacf-45a5-a219-c50203cc243b
Current count: 3000, row: 0e6802ad-dfed-462c-a994-e6f6800eb93c
Current count: 4000, row: 133c9c66-6d3e-4ae7-aec3-e981ff96124d
Current count: 5000, row: 17d538ea-dbde-4182-ad20-dc88005a3403
Current count: 6000, row: 1c71c54d-b582-41c3-9ecc-18aa91ebed0f
Current count: 7000, row: 2149afd4-c3fc-4045-b696-90bcec78f72b
Current count: 8000, row: 263ae1b2-53e7-46ba-a719-694ed9bc9ae8
Current count: 9000, row: 2af7ff2c-4eaf-46ee-b754-4a1124a1f8b5
Current count: 10000, row: 2fd9b685-3fdc-4368-ada4-303212a75381
Current count: 11000, row: 345ec60e-dc27-4dfe-b84e-dd591d19f788
Current count: 12000, row: 38f2724f-936f-48c0-bd2a-4617c8d62d5c
Current count: 13000, row: 3dec6d4f-edc6-4883-bfcf-bbe489997a46
Current count: 14000, row: 42b8e496-1874-4c74-bcb7-f170af0d8d25
Current count: 15000, row: 4782cec3-5948-422d-bd3a-b3b8b74f7aee
Current count: 16000, row: 4c879e9e-ce14-4325-8781-a38dfca69cad
Current count: 17000, row: 51766304-a279-4df8-9b9f-0f06c0b85c08
Current count: 18000, row: 5691dedc-d1de-42d6-9f78-e1c358226a1e
Current count: 19000, row: 5b665c48-d2f4-47cd-9471-c4817a09ff1e
Current count: 20000, row: 60507d84-b939-4bfc-a477-1c1df17a2c85
Current count: 21000, row: 6532f9a3-6a2c-46f0-8bbf-7bdbb152e159
Current count: 22000, row: 6a330cbf-dd21-49d9-b1b5-b07d04f60853
Current count: 23000, row: 6f33c1b3-02c8-475b-828f-b20bdecebcfc
Current count: 24000, row: 7415c760-c334-4201-8b87-91b89b53ff53
Current count: 25000, row: 7897256b-9f8a-4b18-baab-0254972b7673
Current count: 26000, row: 7d4d341e-dd35-4faf-a47e-0fd013d86b8e
Current count: 27000, row: 820d4acc-63fc-4a2f-80c4-e0aa3ec45d04
Current count: 28000, row: 8709dcd9-cc17-4fe2-8749-1ce4530213e0
Current count: 29000, row: 8bc736c3-7367-47d7-a4d9-391df7012bc8
Current count: 30000, row: 907c70c5-3c02-48d9-ad1d-ab60119c7d00
Current count: 31000, row: 952d010e-57b3-4e6b-80ae-cdfbd320bb4b
Current count: 32000, row: 9a144e87-79a0-483d-9c53-5b2a6d0036b1
Current count: 33000, row: 9ed56917-97ad-4afa-9df8-5ed1d3374896
Current count: 34000, row: a3984dbd-7b36-415c-a92c-00713dc6d532
Current count: 35000, row: a838efa7-f53a-4f66-8fa4-56a554d995fa
Current count: 36000, row: ad02864c-e341-498d-897d-52209f043bee
Current count: 37000, row: b1e85dcf-de47-42b7-9d4f-4ba38585f6eb
Current count: 38000, row: b683870e-e2b0-4b66-ab2d-4b43928f4bb2
Current count: 39000, row: bb360d38-f415-43c3-bcb9-ea0314e92990
```

m. In Hbase check details of lookup_data_hive integrated tables

```
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 => 'N
ONE', 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', COMPRESSIO
N => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0350 seconds
```

n. In Hbase, alter lookup_data_hive table and set VERSIONS to 10 for lookup_transaction_family

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

```
hbase(main):006:0> alter 'lookup_data_hive', {NAME => 'lookup_transaction_family', VERSIONS => 10}
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 2.4760 seconds
```

o. Check details of lookup_data_hive and confirm that VERSIONS is set to 10 for lookup_transaction_family

```
describe 'lookup_data_hive'
```

```
hbase(main):007:0> 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 => 'N
ONE', 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', COMPRESSI
ON => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0250 seconds
```

SECTION: 2 Script to ingest the relevant data from AWS RDS to Hadoop. This is from file DataIngestion.txt

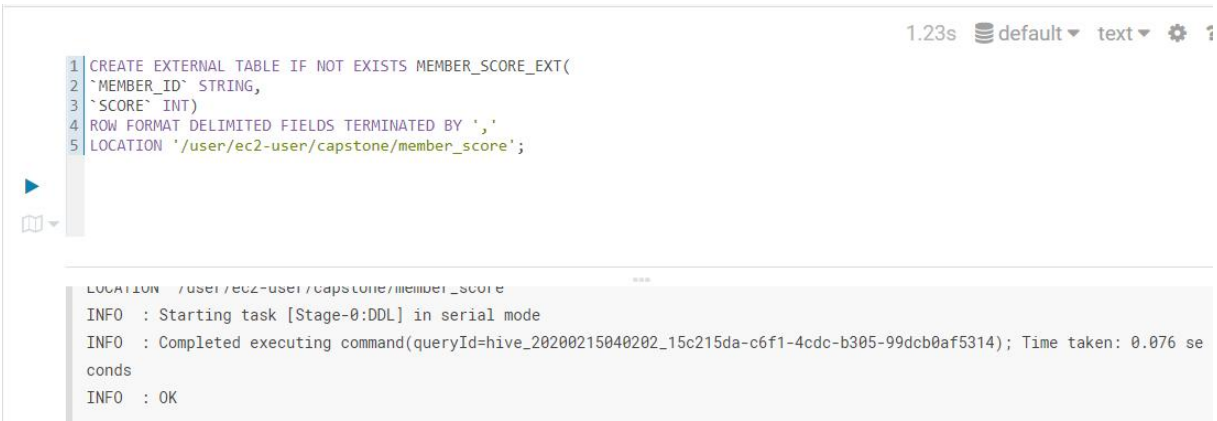
Script to ingest the relevant data from AWS RDS to Hadoop.

a. Create external table card_member_ext which will point to HDFS 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 '/user/ec2-user/capstone/card_member';
```

b. Create external table member_score_ext which will point to HDFS location

```
CREATE EXTERNAL TABLE IF NOT EXISTS MEMBER_SCORE_EXT(
'MEMBER_ID' STRING,
'SCORE' INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/ec2-user/capstone/member_score';
```



c. Create card_member_orc table

```
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");
```

d. Create member_score_orc table

```
CREATE TABLE IF NOT EXISTS MEMBER_SCORE_ORC(
'MEMBER_ID' STRING,
'SCORE' INT)
```

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

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

```
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215040505_aa73c021-5ae3-4b12-91e1-3ad501732bc5); Time taken: 0.064 se
conds
INFO  : OK
```

Success.

e. Load data into card_member_orc

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

```
INFO  : Total MapReduce CPU Time Spent: 10 seconds 410 msec
INFO  : Completed executing command(queryId=hive_20200215040606_79660fbf-9e55-4d59-bb80-13f1d
conds
INFO  : OK
```

job_1581737473027_0001
job_1581737473027_0002

Success.

f. Load data into member_score_orc

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

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

```
INFO  : Total MapReduce CPU Time Spent: 10 seconds 510 msec
INFO  : Completed executing command(queryId=hive_20200215040808_a2b65b43-5c55-4ffc-9e3a-033e2
econds
INFO  : OK
```

job_1581737473027_0003
job_1581737473027_0004

Success.

g. Verify some data in card_member_orc table

```
SELECT * FROM CARD_MEMBER_ORC LIMIT 10;
```

```
SELECT * FROM CARD_MEMBER_ORC LIMIT 10;
```

```
INFO  : Executing command(queryId=hive_20200215041313_493ff34b-790c-4ca0-bf79-58980a6fa66c): SELECT * FROM CARD_MEMBER_ORC
LIMIT 10
INFO  : Completed executing command(queryId=hive_20200215041313_493ff34b-790c-4ca0-bf79-58980a6fa66c); Time taken: 0.001 se
conds
INFO  : OK
```

Query History | Saved Queries | Results (10)

	card_member_orc.card_id	card_member_orc.member_id	card_member_orc.member_joining_dt	card_mem
1	340028465709212	009250698176266	2012-02-08 06:04:13.0	05/13
2	340054675199675	835873341185231	2017-03-10 09:24:44.0	03/17
3	340082915339645	512969555857346	2014-02-15 06:30:30.0	07/14
4	340134186926007	887711945571282	2012-02-05 01:21:58.0	02/13
5	340265728490548	680324265406190	2014-03-29 07:49:14.0	11/14
6	340268219434811	929799084911715	2012-07-08 02:46:08.0	08/12
7	340379737226464	089615510858348	2010-03-10 00:06:42.0	09/10
8	340383645652108	181180599313885	2012-02-24 05:32:44.0	10/16

h. Verify some data in member_score_orc table

```
SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;
```

```
SELECT * FROM MEMBER_SCORE_ORC LIMIT 10;
```

```
INFO  : Executing command(queryId=hive_20200215041818_cd1cff9d-9429-47c0-a513-66e450217d5e): SELECT * FROM MEMBER_SCORE_ORC
LIMIT 10
INFO  : Completed executing command(queryId=hive_20200215041818_cd1cff9d-9429-47c0-a513-66e450217d5e); Time taken: 0.001 se
conds
INFO  : OK
```

Query History | Saved Queries | Results (10)

	member_score_orc.member_id	member_score_orc.score
1	000037495066290	339
2	000117826301530	289
3	001147922084344	393
4	001314074991813	225
5	001739553947511	642
6	003761426295463	413
7	004494068832701	217
8	006836124210484	504

SECTION: 3 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. The commands for this is from file PreAnalysis.txt

a. Create table ranked_card_transactions_orc to store last 10 transaction for each card_id

```
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");
```

```
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");
```

```
TBLPROPERTIES ( orc.compress = SNAPPY )
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215100808_c454882b-6f4e-47e2-8c66-a896453f209a); Time taken: 0.112 se
conds
INFO  : OK
```

✔ Success.

b. Create table card_ucl_orc to store UCL values for each card_id

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

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

```
TBLPROPERTIES ( orc.compress = SNAPPY )
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20200215101010_6f05d3ac-5184-435d-a83f-580e75e8388c); Time taken: 0.144 se
conds
INFO  : OK
```

✔ Success.

c. Load data in ranked_card_transactions_orc table.

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

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

```
INFO  : Total MapReduce CPU Time Spent: 18 seconds 120 msec
INFO  : Completed executing command(queryId=hive_20200215101111_eeca3b01-2e4b-4354-8cbb-263a0...); Time taken: 56.147 s
econds
INFO  : OK
```

job_1581760742561_0001

Success.

d. Load data in card_ucl_orc table

```
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 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;
```

```
INFO  : Total MapReduce CPU Time Spent: 8 seconds 840 msec
INFO  : Completed executing command(queryId=hive_20200215101414_48c305ba-cb89-4ecf-8aad-4ab65...); Time taken: 29.66 se
conds
INFO  : OK
```

job_1581760742561_0002

Success.

e. Load data in lookup_data_hbase table.

```
INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE
SELECT
      RCTO.CARD_ID,
      CUO.UCL,
      CMS.SCORE,
      RCTO.POSTCODE,
      RCTO.TRANSACTION_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;
```

```
1
2  INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE
3  SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE, RCTO.TRANSACTION_DT
4  FROM  RANKED_CARD_TRANSACTIONS_ORC RCTO
5  JOIN CARD_UCL_ORC CUO
6  ON CUO.CARD_ID = RCTO.CARD_ID
7  JOIN (
8  SELECT DISTINCT CARD.CARD_ID, SCORE.SCORE
9  FROM CARD_MEMBER_ORC CARD
10 JOIN MEMBER_SCORE_ORC SCORE
11 ON CARD.MEMBER_ID = SCORE.MEMBER_ID) AS CMS
12 ON RCTO.CARD_ID = CMS.CARD_ID
13 WHERE RCTO.RANK = 1;
```

```
INFO  : Stage-Stage-6: Map: 1  Cumulative CPU: 5.8 sec   HDFS Read: 52366 HDFS Write: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 13 seconds 720 msec
INFO  : Completed executing command(queryId=hive_20200215101717_31807ad0-53fd-41af-9956-cbd9d...); Time taken: 63.904 seconds
INFO  : OK
```

job_1581760742561_0003
job_1581760742561_0004

Success.

f. Verify count in lookup_data_hbase table

SELECT count(*) FROM lookup_data_hbase;

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

```
INFO  : Stage-Stage-1: Map: 1  Reduce: 1  Cumulative CPU: 7.21 sec  HDFS Read: 16268 HDFS Write: 4 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 7 seconds 210 msec
INFO  : Completed executing command(queryId=hive_20200215101919_a1c55ce4-18eb-40dc-b162-d4545...); Time taken: 32.453 seconds
INFO  : OK
```

job_1581760742561_0005

Query History | Saved Queries | Results (1)

	_c0
1	999

g. Verify some data in lookup_data_hbase table

SELECT * FROM lookup_data_hbase limit 10;

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

```
INFO  : Executing command(queryId=hive_20200215102222_e5438d95-4526-46bb-892d-7a0c9c8444a7): select * from lookup_data_hbase limit 10
INFO  : Completed executing command(queryId=hive_20200215102222_e5438d95-4526-46bb-892d-7a0c9c8444a7); Time taken: 0.001 seconds
INFO  : OK
```

Query History | Saved Queries | Results (10)

	lookup_data_hbase.card_id	lookup_data_hbase.ucl	lookup_data_hbase.score	lookup_data_hbase.postcod
1	340028465709212	16331555.548882348	233	24658
2	340054675199675	14156079.786189131	631	50140
3	340082915339645	15285685.330791473	407	17844
4	340134186926007	15239767.522438556	614	67576
5	340265728490548	16084916.71255562	202	72435
6	340268219434811	12507323.937605347	415	62513

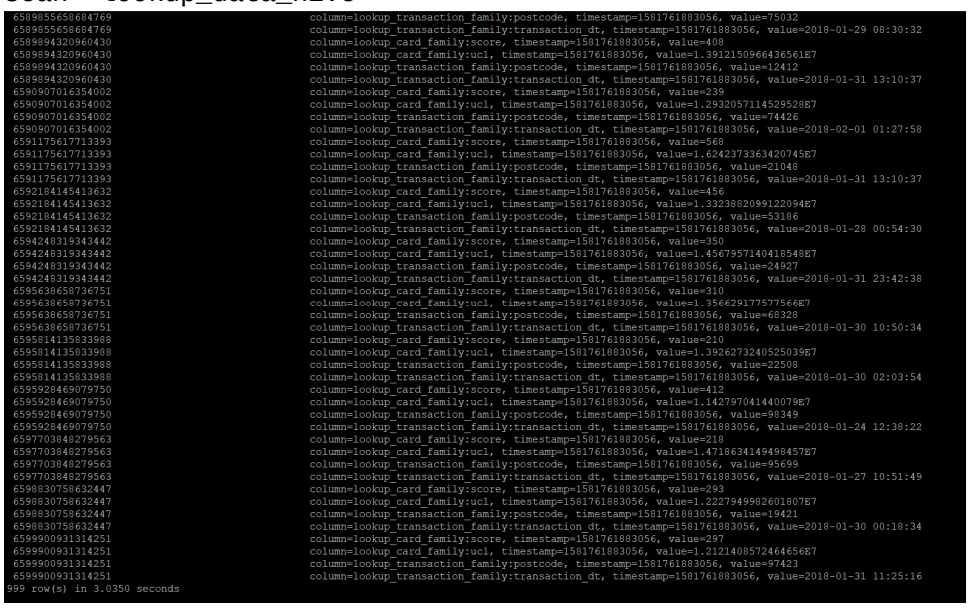
h. Check count in lookup_data_hive table

count 'lookup_data_hive'

```
hbase(main):001:0> count 'lookup_data_hive'
999 row(s) in 0.6330 seconds

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

i. Check data in lookup_data_hive table

scan 'lookup_data_hive'



SECTION 4. SCHEDULING USING OOZIE 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. Commands relevant to this are in file sqoop_oozie_hbase.txt

a. Start Sqoop metastore

sudo -u sqoop sqoop-metastore

b. Setup sqoop job to import card_member data incrementally from RDS into HDFS

sqoop job --create extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop -- import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data --username upgraduser --password upgraduser --table card_member --null-string 'NA' --null-non-string '\\N' --incremental lastmodified --check-column member_joining_dt --last-value 0 --merge-key card_id --target-dir '/user/ec2-user/capstone/card_member'

sqoop job --create extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop -- import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data--username upgraduser --password upgraduser --table member_score --null-string 'NA' --null-non-string '\\N' --delete-target-dir --target-dir '/user/ec2-user/capstone/member_score'

c. Verify Sqoop Jobs

sqoop job --list --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop

sqoop job --show extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop

sqoop job --show extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop

d. Check if the Sqoop jobs are getting executed

sqoop job --exec extract_card_member --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop

sqoop job --exec extract_member_score --meta-connect jdbc:hsqldb:hsql://ip-10-0-0-243.ec2.internal:16000/sqoop

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

e. **Login as root**
   sudo su –

f. **Login as hdfs**
   su – hdfs

g. **Export OOZIE_URL**
   export OOZIE_URL=http://ip-10-0-0-243.ec2.internal:11000/oozie

h. **Check oozie shared library for sqoop**
   oozie admin -shareliblist sqoop

i. **Start updating oozie shared library**
   oozie admin -sharelibupdate
   [ShareLib update status]
   sharelibDirOld = hdfs://ip-10-0-0-243.ec2.internal:8020/user/oozie/share/lib/lib_20190620223511
   host = http://ec2-54-208-194-25.compute-1.amazonaws.com:11000/oozie

   sharelibDirNew = hdfs://ip-10-0-0-243.ec2.internal:8020/user/oozie/share/lib/lib_20190620223511
   status = Successful

j. **Find mysql connector jar**
   find / -name mysql*jar
   Above command found mysql connector jar at this location - /var/lib/oozie/mysql-connector-java.jar

k. **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_20190620223511/sqoop/.

   hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/sqoop/mysql-connector-java.jar

   hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/sqoop/mysql-connector-java.jar

l. **Check oozie shared library for hive**
   oozie admin -shareliblist hive

m. **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_20190620223511/hive/.

   hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/hive/hive-site.xml

   hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/hive/hive-site.xml

n. **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_20190620223511/hive/.

   hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/hive/hbase-site.xml

   hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/hive/hbase-site.xml

o. **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_20190620223511/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/hive/metrics-core-2.2.0.jar

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/hive/metrics-core-2.2.0.jar

p. 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-5.15.1-1.cdh5.15.1.p0.4/jars/hive-hbase-handler-1.1.0-cdh5.15.1.jar /user/oozie/share/lib/lib_20190620223511/hive/.

hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/hive/hive-hbase-handler-1.1.0-cdh5.15.1.jar

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/hive/hive-hbase-handler-1.1.0-cdh5.15.1.jar

q. 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_20190620223511/hive/.; done

hadoop fs -chown oozie /user/oozie/share/lib/lib_20190620223511/hive/hbase*

hadoop fs -chmod 775 /user/oozie/share/lib/lib_20190620223511/hive/hbase*

r. Finish updating oozie shared library

oozie admin -sharelibupdate

[ShareLib update status]

sharelibDirOld = hdfs://ip-10-0-0-243.ec2.internal:8020/user/oozie/share/lib/lib_20190620223511
host = http://ec2-34-230-47-250.compute-1.amazonaws.com:11000/oozie

sharelibDirNew = hdfs://ip-10-0-0-243.ec2.internal:8020/user/oozie/share/lib/lib_20190620223511
status = Successful

s. Update sqoop-site.xml

/etc/sqoop/conf/sqoop-site.xml

```
<configuration>
        <property>
                <name>sqoop.metastore.client.autoconnect.url</name>
                <value>jdbc:hsqldb:hsql://ip-10-0-0-
243.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>
```

t. Create directory in HDFS for oozie workflow. Put sqoop-site.xml in oozie workflow application location. Put workflow.xml in oozie workflow application location. Put lookupDataRefresh.hql in oozie workflow application location. Put coordinator.xml in oozie workflow location

hadoop fs -mkdir -p /capstone/oozie_workflow/app

```
hadoop fs -put /etc/sqoop/conf/sqoop-site.xml /user/ec2-user/capstone/oozie_workflow/app/.

hadoop fs -put workflow.xml /user/ec2-user/capstone/oozie_workflow/app/.

hadoop fs -put lookupDataRefresh.hql /user/ec2-user/capstone/oozie_workflow/app/.
hadoop fs -put coordinator.xml /user/ec2-user/capstone/oozie_workflow/.
```

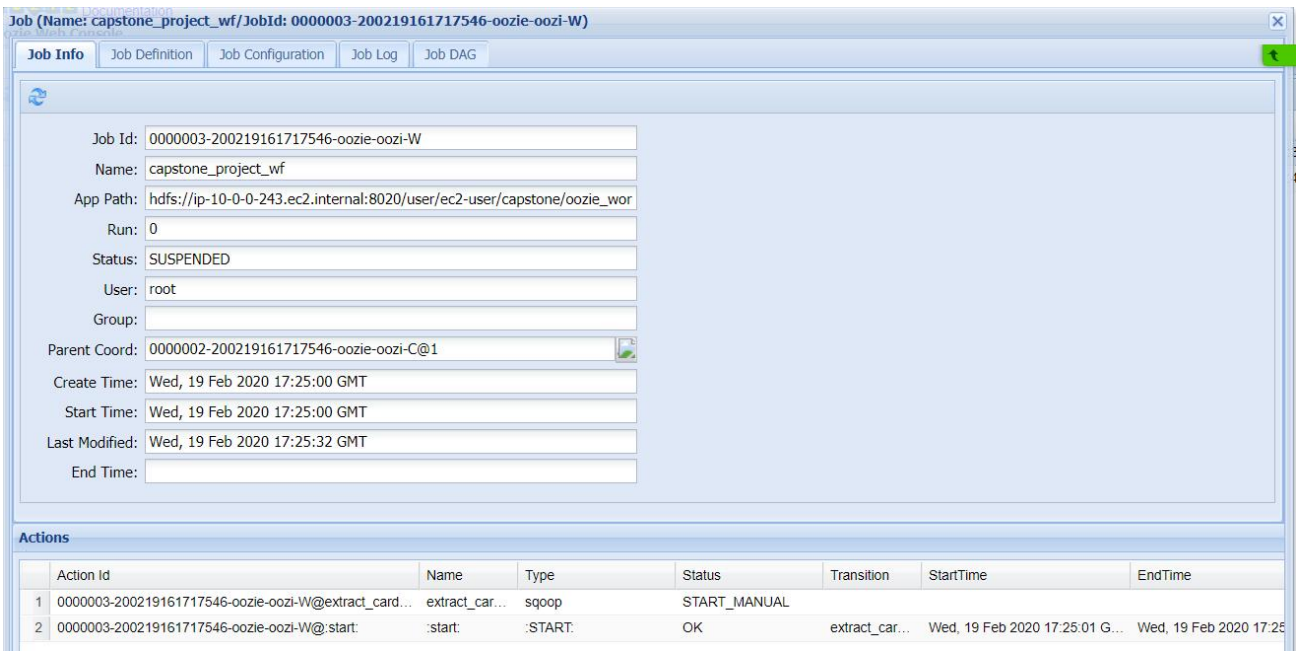
u. Copy job.properties.withoutcoordinator as job.properties. Run oozie job without coordinator. Wait for oozie job completion (job id was returned by previous command). Copy job.properties.withcoordinator as job.properties. Run oozie job with coordinator. Verify oozie job (job id was returned by previous command)

Kill an oozie running job

```
[root@ip-10-0-0-243 ec2-user]# oozie job -kill  0000000-200219161717546-oozie-oozi-C  -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie
```

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

oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -config job.properties -run

oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -info 0000002-200219161717546-oozie-oozi-C
```

```
[root@ip-10-0-0-243 ec2-user]# oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -config job.properties -run
job: 0000002-200219161717546-oozie-oozi-C
[root@ip-10-0-0-243 ec2-user]# oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -info 0000002-200219161717546-oozie-oozi-C
Job ID : 0000002-200219161717546-oozie-oozi-C
------------------------------------------------------------------------------------------------------------------------------------
Job Name      : capstone_proj_coord
App Path      : hdfs://ip-10-0-0-243.ec2.internal:8020/user/ec2-user/capstone/oozie_workflow/coordinator.xml
Status        : RUNNING
Start Time    : 2020-02-19 17:25 GMT
End Time      : 2020-02-19 21:25 GMT
Pause Time    : -
Concurrency : 1
------------------------------------------------------------------------------------------------------------------------------------
ID                                            Status    Ext ID                 Err Code  Created              Nominal Time
0000002-200219161717546-oozie-oozi-C@1        WAITING   -                      -         2020-02-19 17:22 GMT 2020-02-19 17:25 GMT
------------------------------------------------------------------------------------------------------------------------------------
```

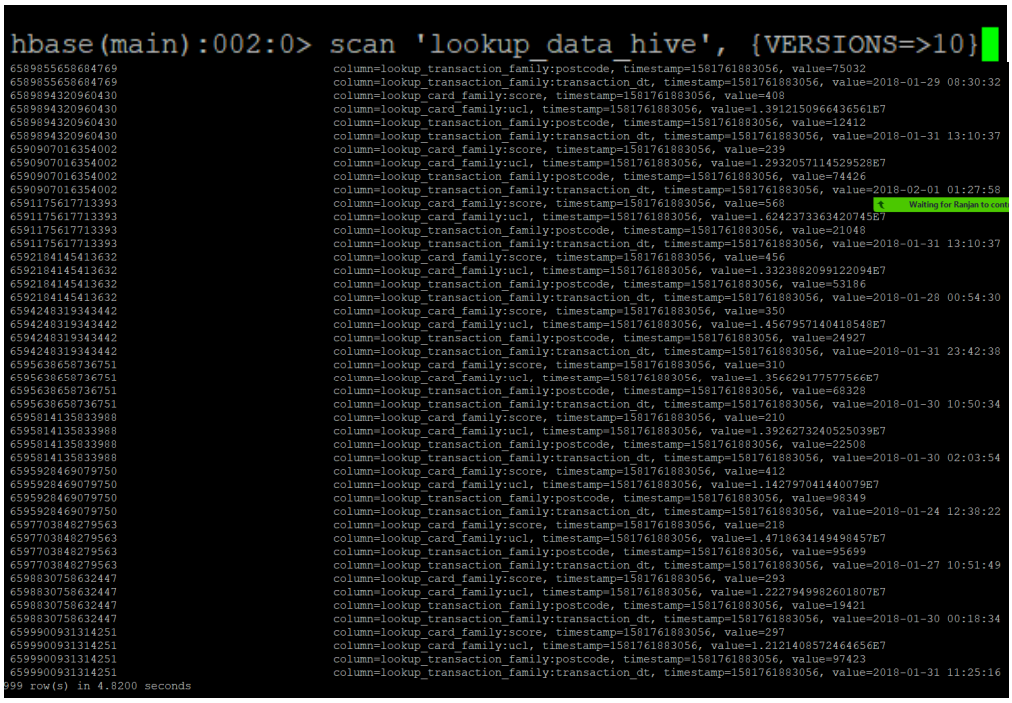



```
cp job.properties.withcoordinator job.properties

oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -config job.properties -run

oozie job -oozie http://ip-10-0-0-243.ec2.internal:11000/oozie -info 0000002-200219161717546-oozie-oozi-C
```

Once oozie jobs are successful, check data in HBase lookup_data_hive table

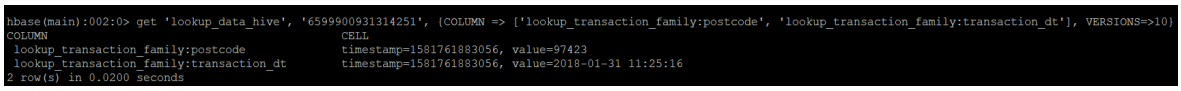
a. Check data in Hbase lookup_data_hive table

```
scan 'lookup_data_hive', {VERSIONS=>10}
```



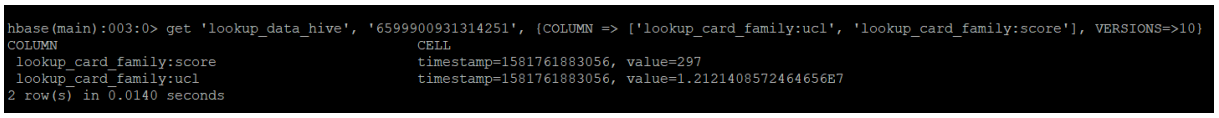
b. Check data for a particular card_id, see multiple versions for postcode and transaction_dt

```
get 'lookup_data_hive', '6599900931314251', {COLUMN =>
['lookup_transaction_family:postcode', 'lookup_transaction_family:transaction_dt'],
VERSIONS=>10}
```



c. 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}
```



d. Check data for a particular card_id

```
get 'lookup_data_hive', '6599900931314251'
```