Credit Card Fraud Detection System

Executed Script – Screenshots – Mid Submission

I. Scripts to load the transactions history data (card_transactions.csv) in a NoSQL database and to create a look-up table with columns specified earlier in the problem statement.

Creating the required directories:

hdfs@ip-10-0-0-87:~

```
[ec2-user@ip-10-0-0-87 ~]$ sudo -i
[root@ip-10-0-0-87 ~]# clear
[root@ip-10-0-0-87 ~]# su - hdfs
Last login: Sun Jul 12 18:13:40 UTC 2020 on pts/0
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -mkdir /user/root/creditcard
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -mkdir /user/root/creditcard/data
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -mkdir /user/root/creditcard/final
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -chown root /user/root/creditcard/data
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -chown root /user/root/creditcard/final
```

1. Script to create (INTERNAL) card_transactions temporary hive table to load the given CSV

```
[root@ip-10-0-0-87 ~]# hive

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/jars/hive-common-1.1.0-cdh5.15.1.jar!/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> CREATE TABLE card_transactions_tmp (card_id string, member_id string, amount double, postcode string, pos_id bigint, transaction_dt string, status string) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' LOCATION '/user/root/creditcard/data/card_transactions_tmp' TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 3.49 seconds
hive>
```

2. Script to load initially card_transactions data from csv

```
hive> load data local inpath '/home/ec2-user/card_transactions.csv' overwrite into table card_transactions_tmp;
Loading data to table default.card_transactions_tmp
Table default.card_transactions_tmp stats: [numFiles=1, totalSize=4829520]
OK
Time taken: 1.183 seconds
hive>
```

3. Check the data loaded

```
hive> select * from card_transactions_tmp limit 10;
OK
348702330256514 000037495066290 9084849.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 330148.0        33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 136052.0        33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 4310362.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 9097094.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 2291118.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 4900011.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 633447.0        33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 6259303.0       33946   614677375609919 11-02-2018 00:00:00     GENUINE
348702330256514 000037495066290 369067.0        33946   614677375609919 11-02-2018 00:00:00     GENUINE
Time taken: 0.695 seconds, Fetched: 10 row(s)
hive>
```

4. Create card_transactions table (HIVE-HABSE integrated table)

```
[root@ip-10-0-0-87 ~]# hbase shell
20/12/27 11:11:33 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> create 'card_transactions','cardtransactions';
hbase(main):002:0* create 'card_lookup','lookup';
hbase(main):003:0*
```

NOTE: For creating HIVE-HBASE integrated table, we need to first create a HBase table and then integrate it with hive table, HIVE external creation will work only on existing HBase table.

Creating external table in hive:

```
hive> CREATE EXTERNAL TABLE card_transaction (key struct<member_id:string, transaction_dt:string, amount:double>, card_id string, postcode string, pos_id bigint, status string) ROW FORMAT D
ELIMITED COLLECTION ITEMS TERMINATED BY '~' STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key, cardtransactions:card_id, car
dtransactions:postcode, cardtransactions:pos_id, cardtransactions:status") TBLPROPERTIES("hbase.table.name" = "card_transactions","hbase.mapred.output.outputtable" = "card_transactions");
OK
Time taken: 2.865 seconds
hive>
```

5. Insert data from card_transactions_tmp table to hive-hbase table

```
hive> INSERT OVERWRITE TABLE card_transaction SELECT named_struct('member_id',ct.member_id,'transaction_dt',ct.transaction_dt,'amount',ct.amount), ct.card_id, ct.postcode, ct.pos_id, ct.sta
tus FROM card_transactions_tmp ct;
Query ID = root_20201227113737_d8b8b979-398b-4849-be0e-481bdfb82b22
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_1609051870123_0001, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0001/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0001
Hadoop job information for Stage-0: number of mappers: 1; number of reducers: 0
2020-12-27 11:37:37,271 Stage-0 map = 0%,  reduce = 0%
2020-12-27 11:37:56,073 Stage-0 map = 100%,  reduce = 0%, Cumulative CPU 8.61 sec
MapReduce Total cumulative CPU time: 8 seconds 610 msec
Ended Job = job_1609051870123_0001
MapReduce Jobs Launched:
Stage-Stage-0: Map: 1   Cumulative CPU: 8.61 sec   HDFS Read: 4842565 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 610 msec
OK
Time taken: 37.28 seconds
hive>
```

6. Check if the data is inserted in hive

```
hive> select * from card_transaction limit 10;
OK
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":1193207.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":136052.0}        348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":1611089.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":1799290.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":1914315.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":2148850.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":217221.0}        348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":2241736.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":2259393.0}       348702330256514 33946   614677375609919 GENUINE
{"member_id":"000037495066290","transaction_dt":"11-02-2018 00:00:00","amount":2291118.0}       348702330256514 33946   614677375609919 GENUINE
Time taken: 0.163 seconds, Fetched: 10 row(s)
```

7. Check if the data is inserted in hbase

```
hbase(main):001:0> scan 'card_transactions', {'LIMIT' => 5}
ROW                                  COLUMN+CELL
 000037495066290~11-02-2018 00:      column=cardtransactions:card_id, timestamp=1609069070604, value=348702330256514
 00:00~1193207.0
 000037495066290~11-02-2018 00:      column=cardtransactions:pos_id, timestamp=1609069070604, value=614677375609919
 00:00~1193207.0
 000037495066290~11-02-2018 00:      column=cardtransactions:postcode, timestamp=1609069070604, value=33946
 00:00~1193207.0
 000037495066290~11-02-2018 00:      column=cardtransactions:status, timestamp=1609069070604, value=GENUINE
 00:00~1193207.0
 000037495066290~11-02-2018 00:      column=cardtransactions:card_id, timestamp=1609069070604, value=348702330256514
 00:00~136052.0
 000037495066290~11-02-2018 00:      column=cardtransactions:pos_id, timestamp=1609069070604, value=614677375609919
 00:00~136052.0
 000037495066290~11-02-2018 00:      column=cardtransactions:postcode, timestamp=1609069070604, value=33946
 00:00~136052.0
 000037495066290~11-02-2018 00:      column=cardtransactions:status, timestamp=1609069070604, value=GENUINE
 00:00~136052.0
 000037495066290~11-02-2018 00:      column=cardtransactions:card_id, timestamp=1609069070604, value=348702330256514
 00:00~1611089.0
 000037495066290~11-02-2018 00:      column=cardtransactions:pos_id, timestamp=1609069070604, value=614677375609919
 00:00~1611089.0
 000037495066290~11-02-2018 00:      column=cardtransactions:postcode, timestamp=1609069070604, value=33946
 00:00~1611089.0
 000037495066290~11-02-2018 00:      column=cardtransactions:status, timestamp=1609069070604, value=GENUINE
 00:00~1611089.0
 000037495066290~11-02-2018 00:      column=cardtransactions:card_id, timestamp=1609069070604, value=348702330256514
 00:00~1799290.0
 000037495066290~11-02-2018 00:      column=cardtransactions:pos_id, timestamp=1609069070604, value=614677375609919
 00:00~1799290.0
 000037495066290~11-02-2018 00:      column=cardtransactions:postcode, timestamp=1609069070604, value=33946
 00:00~1799290.0
 000037495066290~11-02-2018 00:      column=cardtransactions:status, timestamp=1609069070604, value=GENUINE
 00:00~1799290.0
 000037495066290~11-02-2018 00:      column=cardtransactions:card_id, timestamp=1609069070604, value=348702330256514
 00:00~1914315.0
 000037495066290~11-02-2018 00:      column=cardtransactions:pos_id, timestamp=1609069070604, value=614677375609919
 00:00~1914315.0
 000037495066290~11-02-2018 00:      column=cardtransactions:postcode, timestamp=1609069070604, value=33946
 00:00~1914315.0
 000037495066290~11-02-2018 00:      column=cardtransactions:status, timestamp=1609069070604, value=GENUINE
 00:00~1914315.0
5 row(s) in 0.3230 seconds
```

8. Script to create final lookup table (HIVE-HABSE integrated)

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS card_lookup (
    > card_id string,
    > UCL decimal,
    > postcode string,
    > transaction_dt string,
    > score bigint)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,
    > lookup:UCL,
    > lookup:postcode,
    > lookup:transaction_dt,
    > lookup:score")
    > TBLPROPERTIES("hbase.table.name" = "card_lookup" ,
    > "hbase.mapred.output.outputtable" = "card_lookup" );
OK
Time taken: 0.149 seconds
```

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

1. Switching to hdfs user

```
[ec2-user@ip-10-0-0-87 ~]$ sudo -i su - hdfs
Last login: Sun Dec 27 07:08:21 UTC 2020 on pts/0
[hdfs@ip-10-0-0-87 ~]$ set sqoop.metastore.client.record.password=true;
```

2. Create Sqoop job to import data incrementally from AWS RDS

```
[hdfs@ip-10-0-0-87 ~]$ sqoop job --create job_card_member_incremental -- import --connect jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data --username upgraduser --password upgraduser --table card_member --incremental append --check-column member_joining_dt --last-value "2000-01-01 00:00:00" --target-dir /user/root/creditcard/final/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 $ACCUMULO_HOME to the root of your Accumulo installation.
20/12/27 16:04:09 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
20/12/27 16:04:09 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
[hdfs@ip-10-0-0-87 ~]$
```

3. Look for the listed Sqoop job

```
[hdfs@ip-10-0-0-87 ~]$ sqoop job --list
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 $ACCUMULO_HOME to the root of your Accumulo installation.
20/12/27 16:05:20 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
Available jobs:
  job_card_member_incremental
[hdfs@ip-10-0-0-87 ~]$
```

4. View the configuration of the job

```
[hdfs@ip-10-0-0-87 ~]$ sqoop job --show job_card_member_incremental
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.
20/12/27 16:07:09 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
Enter password:
Job: job_card_member_incremental
Tool: import
Options:
----------------------------
relaxed.isolation = false
import.direct.split.size = 0
export.new.update = UpdateOnly
sqoop.oracle.escaping.disabled = true
import.max.inline.lob.size = 16777216
enable.compression = false
customtool.options.jsonmap = {}
direct.import = false
codegen.input.delimiters.enclose.required = false
verbose = false
db.connect.string = jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data
mapreduce.num.mappers = 4
db.clear.staging.table = false
codegen.input.delimiters.field = 0
codegen.input.delimiters.record = 0
codegen.output.dir = .
hdfs.append.dir = true
hcatalog.create.table = false
hcatalog.drop.and.create.table = false
incremental.last.value = 2000-01-01 00:00:00
reset.onemapper = false
incremental.col = member_joining_dt
sqoop.throwOnError = false
db.username = upgraduser
codegen.output.delimiters.escape = 0
accumulo.batch.size = 10240000
hbase.create.table = false
db.batch = false
db.require.password = true
codegen.output.delimiters.enclose = 0
import.fetch.size = null
incremental.mode = AppendRows
temporary.dirRoot = _sqoop
mainframe.input.dataset.type = p
accumulo.max.latency = 5000
hive.drop.delims = false
db.table = card_member
hive.import = false
hive.overwrite.table = false
codegen.output.delimiters.record = 10
split.limit = null
```

```
hbase.bulk.load.enabled = false
codegen.compile.dir = /tmp/sqoop-hdfs/compile/c58997c9f22c69af1b584e6294e8c210
hive.fail.table.exists = false
accumulo.create.table = false
hdfs.delete-target.dir = false
codegen.output.delimiters.enclose.required = false
sqlconnection.metadata.transaction.isolation.level = 2
hdfs.file.format = TextFile
hdfs.target.dir = /user/root/creditcard/final/card_member
codegen.input.delimiters.escape = 0
codegen.input.delimiters.enclose = 0
codegen.auto.compile.dir = true
codegen.output.delimiters.field = 44
[hdfs@ip-10-0-0-87 ~]$
```

5. Execute the job to load initial data

```
[hdfs@ip-10-0-0-87 ~]$ sqoop job --exec job_card_member_incremental
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.
20/12/27 16:12:09 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
Enter password:
20/12/27 16:12:28 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/12/27 16:12:28 INFO tool.CodeGenTool: Beginning code generation
20/12/27 16:12:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
20/12/27 16:12:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `card_member` AS t LIMIT 1
20/12/27 16:12:28 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
Note: /tmp/sqoop-hdfs/compile/95c669f4ea2151d5fe35a5c242820b34/card_member.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/12/27 16:12:30 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-hdfs/compile/95c669f4ea2151d5fe35a5c242820b34/card_member.jar
20/12/27 16:12:31 INFO tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`member_joining_dt`) FROM `card_member`
20/12/27 16:12:31 INFO tool.ImportTool: Incremental import based on column `member_joining_dt`
20/12/27 16:12:31 INFO tool.ImportTool: Lower bound value: '2000-01-01 00:00:00'
20/12/27 16:12:31 INFO tool.ImportTool: Upper bound value: '2018-02-01 08:02:34.0'
20/12/27 16:12:31 WARN manager.MySQLManager: It looks like you are importing from mysql.
20/12/27 16:12:31 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/12/27 16:12:31 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/12/27 16:12:31 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/12/27 16:12:31 INFO mapreduce.ImportJobBase: Beginning import of card_member
20/12/27 16:12:31 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/12/27 16:12:31 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/12/27 16:12:31 INFO client.RMProxy: Connecting to ResourceManager at ip-10-0-0-87.ec2.internal/10.0.0.87:8032
20/12/27 16:12:34 INFO db.DBInputFormat: Using read commited transaction isolation
20/12/27 16:12:34 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`card_id`), MAX(`card_id`) FROM `card_member` WHERE ( `member_joining_dt` > '2000-01-01 00:00:00' AND `membe
r_joining_dt` <= '2018-02-01 08:02:34.0' )
20/12/27 16:12:34 WARN db.TextSplitter: Generating splits for a textual index column.
20/12/27 16:12:34 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.
20/12/27 16:12:34 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.
20/12/27 16:12:35 INFO mapreduce.JobSubmitter: number of splits:6
20/12/27 16:12:35 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1609051870123_0003
20/12/27 16:12:35 INFO impl.YarnClientImpl: Submitted application application_1609051870123_0003
20/12/27 16:12:35 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0003/
20/12/27 16:12:35 INFO mapreduce.Job: Running job: job_1609051870123_0003
20/12/27 16:12:42 INFO mapreduce.Job: Job job_1609051870123_0003 running in uber mode : false
20/12/27 16:12:42 INFO mapreduce.Job:  map 0% reduce 0%
20/12/27 16:12:53 INFO mapreduce.Job:  map 50% reduce 0%
20/12/27 16:13:00 INFO mapreduce.Job:  map 67% reduce 0%
20/12/27 16:13:01 INFO mapreduce.Job:  map 100% reduce 0%
20/12/27 16:13:01 INFO mapreduce.Job: Job job_1609051870123_0003 completed successfully
20/12/27 16:13:01 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=1066806
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=832
                HDFS: Number of bytes written=85082
                HDFS: Number of read operations=24
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=12
        Job Counters
                Launched map tasks=6
                Other local map tasks=6
                Total time spent by all maps in occupied slots (ms)=40413
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=40413
                Total vcore-milliseconds taken by all map tasks=40413
                Total megabyte-milliseconds taken by all map tasks=41382912
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Input split bytes=832
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=486
                CPU time spent (ms)=7340
                Physical memory (bytes) snapshot=1463484416
                Virtual memory (bytes) snapshot=9542955008
                Total committed heap usage (bytes)=1896873984
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=85082
20/12/27 16:13:01 INFO mapreduce.ImportJobBase: Transferred 83.0879 KB in 30.0941 seconds (2.7609 KB/sec)
20/12/27 16:13:01 INFO mapreduce.ImportJobBase: Retrieved 999 records.
20/12/27 16:13:01 INFO util.AppendUtils: Appending to directory card_member
20/12/27 16:13:01 INFO util.AppendUtils: Using found partition 6
20/12/27 16:13:01 INFO tool.ImportTool: Saving incremental import state to the metastore
20/12/27 16:13:01 INFO tool.ImportTool: Updated data for job: job_card_member_incremental
[hdfs@ip-10-0-0-87 ~]$
```

6. List contents of /user/root/creditcard/card_member, to check if the import is working

```
[hdfs@ip-10-0-0-87 ~]$ hadoop fs -ls /user/root/creditcard/final/card_member
Found 12 items
-rw-r--r--   3 hdfs supergroup          0 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00000
-rw-r--r--   3 hdfs supergroup      23080 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00001
-rw-r--r--   3 hdfs supergroup      20684 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00002
-rw-r--r--   3 hdfs supergroup      19608 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00003
-rw-r--r--   3 hdfs supergroup      21624 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00004
-rw-r--r--   3 hdfs supergroup         86 2020-12-27 13:50 /user/root/creditcard/final/card_member/part-m-00005
-rw-r--r--   3 hdfs supergroup          0 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00006
-rw-r--r--   3 hdfs supergroup      23080 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00007
-rw-r--r--   3 hdfs supergroup      20684 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00008
-rw-r--r--   3 hdfs supergroup      19608 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00009
-rw-r--r--   3 hdfs supergroup      21624 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00010
-rw-r--r--   3 hdfs supergroup         86 2020-12-27 16:12 /user/root/creditcard/final/card_member/part-m-00011
```

7. Create card_member table

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS card_member (card_id string, member_id string, member_joining_dt timestamp, card_purchase_dt string, country string, city string) row format delimi
ted fields terminated by ',' location '/user/root/creditcard/final/card_member';
OK
Time taken: 1.863 seconds
hive>
```

8. Check if the data is inserted (Data automatically loaded into table)

```
hive> select * from card_member limit 10;
OK
340028465709212 009250698176266 2012-02-08 06:04:13     05/13   United States   Barberton
340054675199675 835873341185231 2017-03-10 09:24:44     03/17   United States   Fort Dodge
340082915339645 512969555857346 2014-02-15 06:30:30     07/14   United States   Graham
340134186926007 887711945571282 2012-02-05 01:21:58     02/13   United States   Dix Hills
340265728490548 680324265406190 2014-03-29 07:49:14     11/14   United States   Rancho Cucamonga
340268219434811 929799084911715 2012-07-08 02:46:08     08/12   United States   San Francisco
340379737226464 089615510858348 2010-03-10 00:06:42     09/10   United States   Clinton
340383645652108 181180599313885 2012-02-24 05:32:44     10/16   United States   West New York
340803866934451 417664728506297 2015-05-21 04:30:45     08/17   United States   Beaverton
340889618969736 459292914761635 2013-04-23 08:40:11     11/15   United States   West Palm Beach
Time taken: 0.545 seconds, Fetched: 10 row(s)
hive>
```

9. Script to insert/load data from hadoop into card_member table (Load data is not required)

```
hive> load data inpath '/user/root/creditcard/final/card_member/part*' into table card_member;
Loading data to table default.card_member
Table default.card_member stats: [numFiles=12, numRows=0, totalSize=170164, rawDataSize=0]
OK
Time taken: 2.168 seconds
hive>
```

10. Create Sqoop job to import member_score data from AWS RDS (It need to be refreshed in 4 hours)

```
[hdfs@ip-10-0-0-87 ~]$ set sqoop.metastore.client.record.password=true;
[hdfs@ip-10-0-0-87 ~]$ sqoop job --create job_member_score_complete -- import --connect jdbc:mysql://upgradawsrds1.cyaie1c9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data -username up
graduser --password upgraduser --table member_score --target-dir /user/root/creditcard/temporary/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.
20/12/27 17:43:56 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
20/12/27 17:43:57 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
```

11. Execute to load the initial data

```
[hdfs@ip-10-0-0-87 ~]$ sqoop job --exec job_member_score_complete
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.
20/12/27 17:48:32 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.1
Enter password:
20/12/27 17:48:42 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/12/27 17:48:42 INFO tool.CodeGenTool: Beginning code generation
20/12/27 17:48:42 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
20/12/27 17:48:42 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `member_score` AS t LIMIT 1
20/12/27 17:48:42 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
Note: /tmp/sqoop-hdfs/compile/d430fa7db2561eb00269200acd0023a4/member_score.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/12/27 17:48:43 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-hdfs/compile/d430fa7db2561eb00269200acd0023a4/member_score.jar
20/12/27 17:48:43 WARN manager.MySQLManager: It looks like you are importing from mysql.
20/12/27 17:48:43 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/12/27 17:48:43 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/12/27 17:48:43 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/12/27 17:48:43 INFO mapreduce.ImportJobBase: Beginning import of member_score
20/12/27 17:48:44 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/12/27 17:48:44 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/12/27 17:48:45 INFO client.RMProxy: Connecting to ResourceManager at ip-10-0-0-87.ec2.internal/10.0.0.87:8032
20/12/27 17:48:46 INFO db.DBInputFormat: Using read commited transaction isolation
20/12/27 17:48:46 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`member_id`), MAX(`member_id`) FROM `member_score`
20/12/27 17:48:46 WARN db.TextSplitter: Generating splits for a textual index column.
20/12/27 17:48:46 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.
20/12/27 17:48:46 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.
20/12/27 17:48:47 INFO mapreduce.JobSubmitter: number of splits:6
20/12/27 17:48:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1609051870123_0004
20/12/27 17:48:48 INFO impl.YarnClientImpl: Submitted application application_1609051870123_0004
20/12/27 17:48:48 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0004/
20/12/27 17:48:48 INFO mapreduce.Job: Running job: job_1609051870123_0004
20/12/27 17:48:55 INFO mapreduce.Job: Job job_1609051870123_0004 running in uber mode : false
20/12/27 17:48:55 INFO mapreduce.Job:  map 0% reduce 0%
20/12/27 17:49:04 INFO mapreduce.Job:  map 33% reduce 0%
20/12/27 17:49:05 INFO mapreduce.Job:  map 50% reduce 0%
20/12/27 17:49:12 INFO mapreduce.Job:  map 67% reduce 0%
20/12/27 17:49:13 INFO mapreduce.Job:  map 83% reduce 0%
20/12/27 17:49:14 INFO mapreduce.Job:  map 100% reduce 0%
20/12/27 17:49:14 INFO mapreduce.Job: Job job_1609051870123_0004 completed successfully
20/12/27 17:49:14 INFO mapreduce.Job: Counters: 30
```

```
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=1061220
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=879
                HDFS: Number of bytes written=19980
                HDFS: Number of read operations=24
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=12
        Job Counters
                Launched map tasks=6
                Other local map tasks=6
                Total time spent by all maps in occupied slots (ms)=40377
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=40377
                Total vcore-milliseconds taken by all map tasks=40377
                Total megabyte-milliseconds taken by all map tasks=41346048
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Input split bytes=879
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=576
                CPU time spent (ms)=7290
                Physical memory (bytes) snapshot=1391534080
                Virtual memory (bytes) snapshot=9518235648
                Total committed heap usage (bytes)=1896873984
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=19980
20/12/27 17:49:14 INFO mapreduce.ImportJobBase: Transferred 19.5117 KB in 30.0653 seconds (664.553 bytes/sec)
20/12/27 17:49:14 INFO mapreduce.ImportJobBase: Retrieved 999 records.
```

12. Create member_score table

```
[hdfs@ip-10-0-0-87 ~]$ hive

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/jars/hive-common-1.1.0-cdh5.15.1.jar!/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> CREATE EXTERNAL TABLE IF NOT EXISTS member_score (member_id string, score bigint) row format delimited fields terminated by ',' location '/user/root/creditcard/final/member_score';
OK
Time taken: 1.829 seconds
hive>
```

13. Load member_score table

```
hive> load data inpath '/user/root/creditcard/temporary/member_score/part*' overwrite into table member_score;
Loading data to table default.member_score
Table default.member_score stats: [numFiles=6, totalSize=19980]
OK
Time taken: 1.188 seconds
hive>
```

14. Check if the data is inserted

```
hive> select * from member_score 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: 0.274 seconds, Fetched: 10 row(s)
hive>
```

III. 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.

1. Create raw lookup table

```
hive> CREATE TABLE IF NOT EXISTS card_lookup_raw (card_id string, member_id string, ucl decimal) LOCATION '/user/root/creditcard/temporary/card_lookup_raw';
OK
Time taken: 0.119 seconds
hive>
```

2. Insert data into card_lookup_raw

```
hive> INSERT OVERWRITE TABLE card_lookup_raw SELECT card_id, member_id, (AVG(amount) + (3 * STDDEV_POP(amount))) as ucl FROM (SELECT card_id, key.member_id as member_id, key.amount as amoun
t, row_number() OVER (PARTITION BY card_id order by UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank FROM card_transaction WHERE status = "GENUINE") a WHERE rank <= 1
0 GROUP BY card_id, member_id;
Query ID = hdfs_20201227182121_584180dd-10f6-4784-970f-b05dcd9f47fa
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 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_1609051870123_0005, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0005/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-12-27 18:21:24,553 Stage-1 map = 0%,  reduce = 0%
2020-12-27 18:21:38,116 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.43 sec
2020-12-27 18:21:48,602 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.53 sec
MapReduce Total cumulative CPU time: 12 seconds 530 msec
Ended Job = job_1609051870123_0005
Launching Job 2 out of 2
Number of reduce tasks not specified. Estimated from input data size: 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_1609051870123_0006, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0006/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0006
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-12-27 18:21:58,322 Stage-2 map = 0%,  reduce = 0%
2020-12-27 18:22:04,539 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.54 sec
2020-12-27 18:22:11,817 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.2 sec
MapReduce Total cumulative CPU time: 4 seconds 200 msec
Ended Job = job_1609051870123_0006
Loading data to table default.card_lookup_raw
Table default.card_lookup_raw stats: [numFiles=1, numRows=999, totalSize=41645, rawDataSize=40646]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.53 sec   HDFS Read: 19992 HDFS Write: 86557 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.2 sec   HDFS Read: 95146 HDFS Write: 41728 SUCCESS
Total MapReduce CPU Time Spent: 16 seconds 730 msec
OK
Time taken: 64.638 seconds
hive>
```

3. Check if the data is inserted

```
hive> select * from card_lookup_raw limit 10;
OK
340028465709212 009250698176266 16331556
340054675199675 835873341185231 14156080
340082915339645 512969555857346 15285685
340134186926007 887711945571282 15239768
340265728490548 680324265406190 16084917
340268219434811 929799084911715 12507324
340379737226464 089615510858348 14198311
340383645652108 181180599313885 14091750
340803866934451 417664728506297 10843341
340889618969736 459292914761635 13217942
Time taken: 0.082 seconds, Fetched: 10 row(s)
hive>
```

4. Create temporary lookup table

```
hive> CREATE TABLE IF NOT EXISTS card_lookup_tmp (card_id string, member_id string, ucl decimal, score bigint) LOCATION '/user/root/creditcard/temporary/card_lookup_tmp';
OK
Time taken: 0.106 seconds
hive>
```

5. Insert UCL, score data into card_lookup_tmp by joining member_score and card_lookup_raw tables

```
hive> INSERT OVERWRITE TABLE card_lookup_tmp SELECT r.card_id, r.member_id, r.ucl, m.score FROM card_lookup_raw r JOIN member_score m ON (r.member_id = m.member_id);
Query ID = hdfs_20201227183434_1a4443fa-ae55-421e-9aaa-1fca50a21675
Total jobs = 1
2020-12-27 06:34:34     Starting to launch local task to process map join;      maximum memory = 1908932608
2020-12-27 06:34:35     Uploaded 1 File to: file:/tmp/hdfs/f04cfc13-d634-4ca8-859d-0df79376c77b/hive_2020-12-27_18-34-27_692_3387764313373013801-1/-local-10002/HashTable-Stage-4/MapJoin-map
file01--.hashtable (37277 bytes)
2020-12-27 06:34:35     End of local task; Time Taken: 1.275 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1609051870123_0007, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0007/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0007
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2020-12-27 18:34:49,533 Stage-4 map = 0%,  reduce = 0%
2020-12-27 18:34:56,860 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 2.13 sec
MapReduce Total cumulative CPU time: 2 seconds 130 msec
Ended Job = job_1609051870123_0007
Loading data to table default.card_lookup_tmp
Table default.card_lookup_tmp stats: [numFiles=1, numRows=999, totalSize=45641, rawDataSize=44642]
MapReduce Jobs Launched:
Stage-Stage-4: Map: 1   Cumulative CPU: 2.13 sec   HDFS Read: 48686 HDFS Write: 45724 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 130 msec
OK
Time taken: 30.943 seconds
hive>
```

6. Check if the data is inserted

```
hive> select * from card_lookup_tmp limit 10;
OK
340028465709212 009250698176266 16331556        233
340054675199675 835873341185231 14156080        631
340082915339645 512969555857346 15285685        407
340134186926007 887711945571282 15239768        614
340265728490548 680324265406190 16084917        202
340268219434811 9297990849117l5 12507324        415
340379737226464 089615510858348 14198311        229
340383645652108 181180599313885 14091750        645
340803866934451 417664728506297 10843341        502
340889618969736 459292914761635 13217942        330
Time taken: 0.065 seconds, Fetched: 10 row(s)
hive>
```

7. Create lookup temporary table for card transaction

```
hive> CREATE TABLE IF NOT EXISTS lookup_tmp_trans (card_id string, postcode string, transaction_dt string) LOCATION '/user/root/creditcard/temporary/ lookup_tmp_trans';
OK
Time taken: 0.057 seconds
hive>
```

8. Insert into lookup_tmp_trans, a record with latest GENUINE transaction date and postcode

```
hive> INSERT OVERWRITE TABLE lookup_tmp_trans SELECT card_id, postcode, transaction_dt FROM (SELECT card_id , postcode, key.transaction_dt as transaction_dt, row_number() OVER (PARTITION BY
 card_id ORDER BY UNIX_TIMESTAMP(key.transaction_dt, 'dd-MM-yyyy HH:mm:ss') desc) as rank FROM card_transaction WHERE status = 'GENUINE') a WHERE rank == 1;
Query ID = hdfs_20201227184949_fb8e490c-4679-41e0-9c62-c9ff723c9680
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 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_1609051870123_0008, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0008/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-12-27 18:49:51,966 Stage-1 map = 0%,  reduce = 0%
2020-12-27 18:50:02,354 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.24 sec
2020-12-27 18:50:10,705 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.51 sec
MapReduce Total cumulative CPU time: 12 seconds 510 msec
Ended Job = job_1609051870123_0008
Loading data to table default.lookup_tmp_trans
Table default.lookup_tmp_trans stats: [numFiles=1, numRows=999, totalSize=42684, rawDataSize=41685]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.51 sec   HDFS Read: 13138 HDFS Write: 42768 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 510 msec
OK
Time taken: 29.688 seconds
hive>
```

9. Check if the data is inserted correctly

```
hive> select * from lookup_tmp_trans limit 10;
OK
340028465709212 24658   02-01-2018 03:25:35
340054675199675 50140   15-01-2018 19:43:23
340082915339645 17844   26-01-2018 19:03:47
340134186926007 67576   18-01-2018 23:12:50
340265728490548 72435   21-01-2018 02:07:35
340268219434811 62513   16-01-2018 04:30:05
340379737226464 26656   27-01-2018 00:19:47
340383645652108 34734   29-01-2018 01:29:12
340803866934451 87525   31-01-2018 04:23:57
340889618969736 61341   31-01-2018 21:57:18
Time taken: 0.063 seconds, Fetched: 10 row(s)
hive>
```

10. Insert into final lookup table (hive-hbase integrated)

```
hive> INSERT OVERWRITE TABLE card_lookup SELECT l.card_id, l.ucl, t.postcode, t.transaction_dt, l.score FROM card_lookup_tmp l JOIN lookup_tmp_trans t ON l.card_id = t.card_id;
Query ID = hdfs_20201227185555_ceec0b52-f9f3-41c5-82ad-ba167d772558
Total jobs = 1
2020-12-27 06:55:32     Starting to launch local task to process map join;      maximum memory = 1908932608
2020-12-27 06:55:33     Dump the side-table for tag: 1 with group count: 999 into file: file:/tmp/hdfs/f04cfc13-d634-4ca8-859d-0df79376c77b/hive_2020-12-27_18-55-27_504_5658796978031840437-
1/-local-10001/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1609051870123_0009, Tracking URL = http://ip-10-0-0-87.ec2.internal:8088/proxy/application_1609051870123_0009/
Kill Command = /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1609051870123_0009
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 0
2020-12-27 18:55:46,302 Stage-2 map = 0%,  reduce = 0%
2020-12-27 18:55:53,536 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.56 sec
MapReduce Total cumulative CPU time: 3 seconds 560 msec
Ended Job = job_1609051870123_0009
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1   Cumulative CPU: 3.56 sec   HDFS Read: 53367 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 560 msec
OK
Time taken: 27.197 seconds
hive>
```

11. Check if the data is inserted correctly

```
hive> select * from card_lookup limit 10;
OK
340028465709212 16331556        24658   02-01-2018 03:25:35     233
340054675199675 14156080        50140   15-01-2018 19:43:23     631
340082915339645 15285685        17844   26-01-2018 19:03:47     407
340134186926007 15239768        67576   18-01-2018 23:12:50     614
340265728490548 16084917        72435   21-01-2018 02:07:35     202
340268219434811 12507324        62513   16-01-2018 04:30:05     415
340379737226464 14198311        26656   27-01-2018 00:19:47     229
340383645652108 14091750        34734   29-01-2018 01:29:12     645
340803866934451 10843341        87525   31-01-2018 04:23:57     502
340889618969736 13217942        61341   31-01-2018 21:57:18     330
Time taken: 0.111 seconds, Fetched: 10 row(s)
hive>
```