Scripts Execution

Screenshots of the execution of the scripts written

Purpose of document:

- This pdf document intent to contain all the screenshots of the execution of the scripts written.
- The scripts should, after loading the data and creating the look-up table, 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

Attach screenshots with a brief explanation of each step:

Execution of scripts & screenshots:

Step 1: Script execution for “card_transaction” and “look_up_table” creation

- Copy the card_transactions.csv to hive folder.

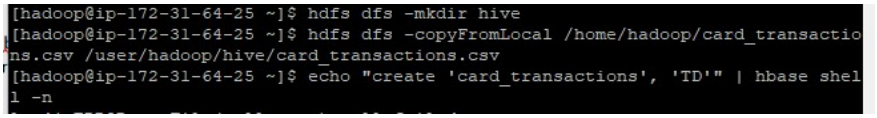
  Hbase command line scripts to create the hive directory and copy the card_transaciotns.csv file:

  hdfs dfs -mkdir hive
  hdfs dfs -copyFromLocal /home/hadoop/card_transactions.csv /user/hadoop/hive/card_transactions.csv

- Hbase shell commands to create card_transacitons table in hbase:

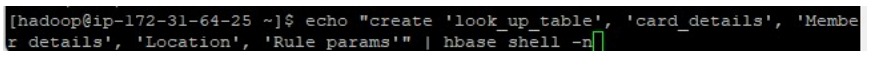
  echo "create 'card_transactions', 'TD'" | hbase shell -n

- Execution screenshots:



- Hbase shell commands to create look_up_table in hbase:

  echo "create 'look_up_table', 'card_details', 'Member_details', 'Location', 'Rule_params'" | hbase shell -n



- Validating tables created:

```
hbase(main):001:0> list
TABLE
card_transactions
look_up_table
2 row(s) in 0.2760 seconds

=> ["card_transactions", "look_up_table"]
```

- Table creation scripts for card_transacitons_stg table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS card_transactions_stg (
card_id STRING,
member_id STRING,
  amount DOUBLE,
  postcode STRING,
  pos_id STRING,
  transaction_dt STRING,
  status STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/historical/card_transactions'
tblproperties("skip.header.line.count"="1");
```

Note: Staging table created to copy the data as is from card_tranactions.csv file.

Execution Screenshots:

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS card_transactions_stg (
    >   card_id STRING,
    >   member_id STRING,
    >   amount DOUBLE,
    >   postcode STRING,
    >   pos_id STRING,
    >   transaction_dt STRING,
    >   status STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LOCATION '/user/hadoop/historical/card_transactions'
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 1.317 seconds
hive>
```

- Table creation scripts for look_up_table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS look_up_table (
 card_id STRING,
 card_purchase_dt STRING,
 transaction_dt STRING,
 member_id STRING,
 member_joining_dt STRING,
 country STRING,
 city STRING,
```

```
 UCL DOUBLE,
 postcode STRING,
 score INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED BY "org.apache.hadoop.hive.hbase.HBaseStorageHandler"
WITH SERDEPROPERTIES ("hbase.columns.mapping" = "card_details:card_purchase_dt,
card_details:transaction_dt,Member_details:member_id, Member_details:member_joining_dt,
Location:country,Location:city,Rule_params:UCL,Rule_params:postcode, Rule_params:score")
TBLPROPERTIES ("hbase.table.name" = "look_up_table");
```

Execution Screenshot:

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS look_up_table (
    >   card_id STRING,
    >   card_purchase_dt STRING,
    >   transaction_dt STRING,
    >   member_id STRING,
    >   member_joining_dt STRING,
    >   country STRING,
    >   city STRING,
    >   UCL DOUBLE,
    >   postcode STRING,
    >   score INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED BY "org.apache.hadoop.hive.hbase.HBaseStorageHandler"
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" = "card_details:card_purchase_dt, card_details:transaction_dt,Member_details:m
ember_id, Member_details:member_joining_dt, Location:country,Location:city,Rule_params:UCL,Rule_params:postcode, Rule_params:score
")
    > TBLPROPERTIES ("hbase.table.name" = "look_up_table");
OK
Time taken: 2.453 seconds
hive>
```

- ### Scripts for Loading data onto card transaction staging table

```
LOAD DATA INPATH 'hdfs:/user/hadoop/hive/card_transactions.csv' OVERWRITE INTO TABLE
card_transactions_stg;
```

Execution Screenshots

```
hive> LOAD DATA INPATH 'hdfs:/user/hadoop/hive/card_transactions.csv' OVERWRITE INTO TABLE card_transactions_stg;
Loading data to table default.card_transactions_stg
OK
Time taken: 1.259 seconds
hive>
```

- ### Creating hive integrated hbase table for card_transactions:

```
CREATE EXTERNAL TABLE IF NOT EXISTS card_transactions (row_key struct<card_id:string,
pos_id:string, transaction_dt:string, amount:double>, card_id STRING, pos_id STRING,
transaction_dt STRING, member_id STRING,amount DOUBLE, postcode STRING,status STRING)
ROW FORMAT DELIMITED
COLLECTION ITEMS TERMINATED BY '~'
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ('hbase.columns.mapping'='TD:card_id, TD:pos_id,
TD:transaction_dt,TD:member_id,TD:amount, TD:postcode, TD:status')
```

TBLPROPERTIES ("hbase.table.name" = "card_transactions");

Execution Screenshots:

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS card_transactions (row_key struct<card_id:string, pos_id:string, transaction_dt:string,
amount:double>, card_id STRING, pos_id STRING, transaction_dt STRING, member_id STRING,amount DOUBLE, postcode STRING,status STRIN
G)
    > ROW FORMAT DELIMITED
    > COLLECTION ITEMS TERMINATED BY '~'
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ('hbase.columns.mapping'='TD:card_id, TD:pos_id, TD:transaction_dt,TD:member_id,TD:amount, TD:postcode,
 TD:status')
    > TBLPROPERTIES ("hbase.table.name" = "card_transactions");
OK
Time taken: 0.161 seconds
hive>
```

- ### Inserting data from staging table to main card_transactions table:

Insert into card_transactions select NAMED_STRUCT('card_id',card_id,'pos_id',pos_id,'transaction_dt', transaction_dt, 'amount', amount) as row_key,card_id,pos_id,transaction_dt, member_id, amount, postcode, status from card_transactions_stg;

Execution Screenshots:

```
hive> Insert into card_transactions  select NAMED_STRUCT('card_id',card_id,'pos_id',pos_id,'transaction_dt', transaction_dt, 'amou
nt', amount) as row_key,card_id,pos_id,transaction_dt, member_id, amount, postcode, status from card_transactions_stg;
Query ID = hadoop_20230729110403_d8413718-673c-4c95-b14e-5b9dc0cb8644
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1690626913936_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 10.68 s
--------------------------------------------------------------------------------
OK
Time taken: 24.095 seconds
```

- ### Validate the rows in card_transactions table:

```
hive> select count(*) from card_transactions;
Query ID = hadoop_20230729111028_28eaabf3-8efa-47f1-a96c-6fac02f6f614
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1690626913936_0003)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 8.20 s
--------------------------------------------------------------------------------
OK
53292
Time taken: 17.63 seconds, Fetched: 1 row(s)
hive>
```

Row count: 53292

```
hive> select count(*) from card_transactions_stg;
Query ID = hadoop_20230729111205_0f13518c-ab12-4155-80c8-83026lb64fla
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1690626913936_0003)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.44 s
--------------------------------------------------------------------------------
OK
53292
Time taken: 6.078 seconds, Fetched: 1 row(s)
hive>
```

Row count: 53292

Step 2 - Getting AWS RDS data into HDFS. This is required to populate look_up_table (NOSQL)

hdfs dfs -mkdir /user/hadoop/sqoop
hdfs dfs -mkdir /user/hadoop/sqoop/import
hdfs dfs -mkdir /user/hadoop/sqoop/import/cred_financials_data

Execution Screenshots:

```
[hadoop@ip-172-31-64-25 ~]$ hdfs dfs -mkdir /user/hadoop/sqoop
[hadoop@ip-172-31-64-25 ~]$ hdfs dfs -mkdir /user/hadoop/sqoop/import
[hadoop@ip-172-31-64-25 ~]$ hdfs dfs -mkdir /user/hadoop/sqoop/import/cred_financials_data
[hadoop@ip-172-31-64-25 ~]$
```

Card Member data from RDS to HDFS:

sqoop import --connect "jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data?connectionCollation=latin1_swedish_ci" --table card_member --target-dir /user/hadoop/sqoop/import/cred_financials_data/card_member --username upgraduser -m 1 --password upgraduser

Execution Screenshots:

```
[hadoop@ip-172-31-64-25 ~]$ sqoop import --connect "jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data?connectionCollation=latin1_swedish_ci" --table card_member --target-dir /user/hadoop/sqoop/import/cred_financials_data/card_member --username upgraduser -m 1 --password upgraduser
```

```
        File Output Format Counters
                Bytes Written=85081
23/07/29 11:20:39 INFO mapreduce.ImportJobBase: Transferred 83.0869 KB in 20.2988 seconds (4.0932 KB/sec)
23/07/29 11:20:39 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[hadoop@ip-172-31-64-25 ~]$
```

Member Score data from RDS to HDFS:

sqoop import --connect "jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data?connectionCollation=latin1_swedish_ci" --table member_score --target-dir /user/hadoop/sqoop/import/cred_financials_data/member_score --username upgraduser -m 1 --password upgraduser

Execution Screenshots:

```
[hadoop@ip-172-31-64-25 ~]$ sqoop import --connect "jdbc:mysql://upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com/cred_financials_data?connectionCollation=latin1_swedish_ci" --table member_score --target-dir /user/hadoop/sqoop/import/cred_financials_data/member_score --username upgraduser -m 1 --password upgraduser
```

```
                Total committed heap usage (bytes)=242221056
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=19980
23/07/29 11:23:44 INFO mapreduce.ImportJobBase: Transferred 19.5117 KB in 18.0934 seconds (1.0784 KB/sec)
23/07/29 11:23:44 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[hadoop@ip-172-31-64-25 ~]$
```

Card_member table:

Creating card member table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS card_member (
 card_id STRING,
 member_id STRING,
 member_joining_dt STRING,
 card_purchase_dt STRING,
 country STRING,
 city STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/RDS/card_member';
```

Execution Screenshots:

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS card_member (
    >   card_id STRING,
    >   member_id STRING,
    >   member_joining_dt STRING,
    >   card_purchase_dt STRING,
    >   country STRING,
    >   city STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LOCATION '/user/hadoop/RDS/card_member';
OK
Time taken: 0.8 seconds
hive>
```

Creating Member score table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS member_score (
 member_id STRING,
 score INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/RDS/member_score';
```

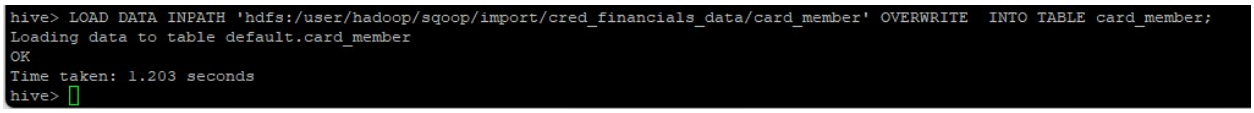
Execution Screenshots:

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS member_score (
    >   member_id STRING,
    >   score INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LOCATION '/user/hadoop/RDS/member_score';
OK
Time taken: 0.057 seconds
hive>
```

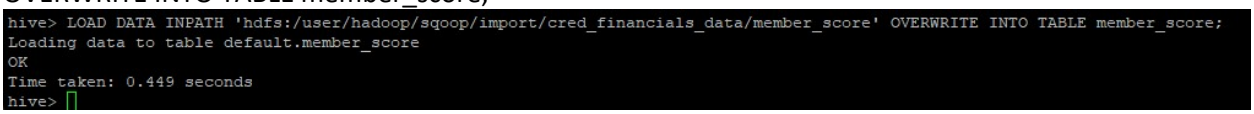
- Loading data on top card member and member score tables:

LOAD DATA INPATH 'hdfs:/user/hadoop/sqoop/import/cred_financials_data/card_member' OVERWRITE INTO TABLE card_member;

Execution Screenshots:



LOAD DATA INPATH 'hdfs:/user/hadoop/sqoop/import/cred_financials_data/member_score' OVERWRITE INTO TABLE member_score;



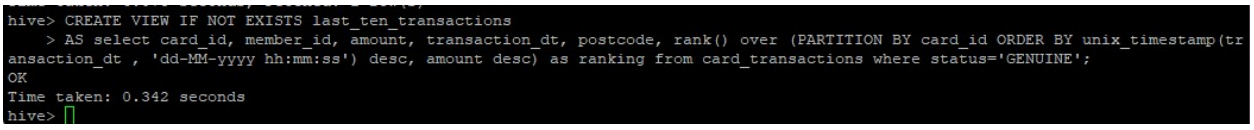
Step 3 - Creating a view for last 10 transactions for each card id.

- Script for view for latest genuine transactions rank wise for each card :

CREATE VIEW IF NOT EXISTS last_ten_transactions
AS select card_id, member_id, amount, transaction_dt, postcode, rank() over (PARTITION BY card_id ORDER BY unix_timestamp(transaction_dt , 'dd-MM-yyyy hh:mm:ss') desc, amount desc) as ranking from card_transactions where status='GENUINE';

This will help to populate lookup table. Required for calculating UCL - Upper control limit value for last 10 transactions.

Execution Screenshots:



Step 4: Script execution for loading data on “look_up_table”:

- Script for inserting data into look_up_table:

```
Insert into look_up_table
select ltt.card_id, cm.card_purchase_dt,ltt.transaction_dt, ltt.member_id, member_joining_dt,
country, city, UCL,ltt.postcode, score
from last_ten_transactions ltt
        inner join member_score ms on ltt.member_id=ms.member_id and ltt.ranking=1
        inner join card_member cm on cm.member_id=ltt.member_id and ltt.ranking=1
        inner join (select card_id, avg(amount)+ (3* stddev(amount)) as UCL
                        from last_ten_transactions
                        where ranking<=10 group by card_id) as ucl
                                on ltt.card_id=ucl.card_id ;
```

Execution Screenshots:

```
hive> Insert into look_up_table select ltt.card_id, cm.card_purchase_dt,ltt.transaction_dt, ltt.member_id, member_joining_dt, coun
try, city, UCL,ltt.postcode, score from last_ten_transactions ltt inner join member_score ms on ltt.member_id=ms.member_id and ltt
.ranking=1 inner join card_member cm on cm.member_id=ltt.member_id and ltt.ranking=1 inner join (select card_id, avg(amount)+ (3*
stddev(amount)) as UCL from last_ten_transactions where ranking<=10 group by card_id) as ucl on ltt.card_id=ucl.card_id ;
No Stats for default@card_transactions, Columns: member_id, amount, postcode, transaction_dt, card_id, status
No Stats for default@member_score, Columns: member_id, score
No Stats for default@card_member, Columns: member_id, country, member_joining_dt, city, card_purchase_dt
No Stats for default@card_transactions, Columns: amount, transaction_dt, card_id, status
Query ID = hadoop_20230729113122_ca40f2a6-11be-46c0-a2a1-682ad54742fc
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1690626913936_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       1
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 25.30 s
--------------------------------------------------------------------------------
OK
Time taken: 41.888 seconds
hive>
```

Capstone project – Credit Card Fraud Detection (mid submission)

by

Susil Patro, Krishna Mohan & Ashmeet Singh Deol