Loading the Lookup Table

Commands to load the relevant data in the Lookup Table

1. Start hive from command prompt. Create table ranked_card_transactions_orc to store last 10 transactions for each card_id. ORC format will help in better performance.

```
CREATE TABLE IF NOT EXISTS RANKED_CARD_TRANSACTIONS_ORC(
`CARD_ID` STRING,
`AMOUNT` DOUBLE,
`POSTCODE` STRING,
`TRANSACTION_DT` TIMESTAMP,
`RANK` INT)
STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");
```

```
hive> 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");
OK
Time taken: 0.052 seconds
```

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

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

```
hive> CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
    > `CARD_ID` STRING,
    > `UCL` DOUBLE)
    > STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.051 seconds
```

3. Load data in ranked_card_transactions_orc table. Here for each card id get top 10 transactions based on the amount column. This is done with SQL using Rank() function partition by card_id sorted by amount in descending order with max # of transactions <= 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, 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;
```

```
hive>
    > 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, AMOU
NT DESC) AS RANK FROM
    > (SELECT CARD_ID, AMOUNT, POSTCODE, TRANSACTION_DT FROM CARD_TRANSACTIONS_HBASE WHERE
    > STATUS = 'GENUINE') A ) B WHERE B.RANK <= 10;
Query ID = hadoop_20230521080530_07c8a249-1e60-4471-91fe-5a84b56e3d9e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1684652299850_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.55 s
--------------------------------------------------------------------------------
Loading data to table capstone_project.ranked_card_transactions_orc
OK
Time taken: 15.746 seconds
hive>
```

4. Load data in card_ucl_orc table. In innermost query, select card_id, average of amount and standard deviation of amount from card_transactions_orc. In outermost query, select card_id and compute UCL using average and standard deviation with formula (avg + (3 * stddev)). Insert all this data in card_ucl_orc.

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

```
hive>
    >
    > 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;
Query ID = hadoop_20230521081015_dcf978e9-667c-4cc8-80bc-df2c0ffc81a7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1684652299850_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.43 s
--------------------------------------------------------------------------------
Loading data to table capstone_project.card_ucl_orc
OK
Time taken: 7.568 seconds
```

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

```
INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE
SELECT RCTO.CARD_ID, CUO.UCL, CMS.SCORE, RCTO.POSTCODE, RCTO.TRANSACTION_DT
FROM RANKED_CARD_TRANSACTIONS_ORC RCTO
JOIN CARD_UCL_ORC CUO
ON CUO.CARD_ID = RCTO.CARD_ID
JOIN (
SELECT DISTINCT CARD.CARD_ID, SCORE.SCORE
FROM CARD_MEMBER_ORC CARD
JOIN MEMBER_SCORE_ORC SCORE
ON CARD.MEMBER_ID = SCORE.MEMBER_ID) AS CMS
ON RCTO.CARD_ID = CMS.CARD_ID
WHERE RCTO.RANK = 1;
```

```
hive> 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;
No Stats for capstone_project@ranked_card_transactions_orc, Columns: postcode, rank, transaction_dt, card_id
No Stats for capstone_project@card_ucl_orc, Columns: card_id, ucl
No Stats for capstone_project@card_member_orc, Columns: member_id, card_id
No Stats for capstone_project@member_score_orc, Columns: member_id, score
Query ID = hadoop_20230521081303_7c55ebf6-e939-46cd-b1e7-c1e5927a5d6e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1684652299850_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 2 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 11.78 s
--------------------------------------------------------------------------------
OK
Time taken: 17.342 seconds
```

6. Verify count in lookup_data_hbase table.

select count(*) from lookup_data_hbase;

```
hive> select count(*) from lookup_data_hbase;
Query ID = hadoop_20230521090038_7f2fb1fb-8a5b-4310-bd20-66b208b350e0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1684652299850_0008)

--------------------------------------------------------------------------------
        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: 6.28 s
--------------------------------------------------------------------------------
OK
999
Time taken: 10.841 seconds, Fetched: 1 row(s)
```

7. Verify some data in lookup_data_hbase table.

select * from lookup_data_hbase limit 10;

```
hive> select * from lookup_data_hbase limit 10;
OK
340028465709212 1.6331555548882348E7    233     24658   2018-01-02 03:25:35
340054675199675 1.4156079786189131E7    631     50140   2018-01-15 19:43:23
340082915339645 1.5285685330791473E7    407     17844   2018-01-26 19:03:47
340134186926007 1.5239767522438556E7    614     67576   2018-01-18 23:12:50
340265728490548 1.608491671255562E7     202     72435   2018-01-21 02:07:35
340268219434811 1.2507323937605347E7    415     62513   2018-01-16 04:30:05
340379737226464 1.4198310998368107E7    229     26656   2018-01-27 00:19:47
340383645652108 1.4091750460468251E7    645     34734   2018-01-29 01:29:12
340803866934451 1.0843341196185412E7    502     87525   2018-01-31 04:23:57
340889618969736 1.3217942365515321E7    330     61341   2018-01-31 21:57:18
Time taken: 0.176 seconds, Fetched: 10 row(s)
```

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

count 'lookup_data_hive';

```
hbase(main):004:0> count 'lookup_data_hive'
999 row(s) in 0.2340 seconds

=> 999
```

9. In HBase, check data in lookup_data_hive table.

scan 'lookup_data_hive'

```
6591175617713393                column=lookup_card_family:score, timestamp=1684659024744, value=568
6591175617713393                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.624237336342O745E7
6591175617713393                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=21048
6591175617713393                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-31 13:10:37
6592184145413632                column=lookup_card_family:score, timestamp=1684659024744, value=456
6592184145413632                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.3323882099122094E7
6592184145413632                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=53186
6592184145413632                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-28 00:54:30
6594248319343442                column=lookup_card_family:score, timestamp=1684659024744, value=350
6594248319343442                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.4567957140418548E7
6594248319343442                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=24927
6594248319343442                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-31 23:42:38
6595638658736751                column=lookup_card_family:score, timestamp=1684659024744, value=310
6595638658736751                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.356629177577566E7
6595638658736751                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=68328
6595638658736751                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-30 10:50:34
6595814135833988                column=lookup_card_family:score, timestamp=1684659024744, value=210
6595814135833988                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.3926273240525039E7
6595814135833988                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=22508
6595814135833988                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-30 02:03:54
6595928469079750                column=lookup_card_family:score, timestamp=1684659024744, value=412
6595928469079750                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.142797041440079E7
6595928469079750                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=98349
6595928469079750                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-24 12:38:22
6597703848279563                column=lookup_card_family:score, timestamp=1684659024744, value=218
6597703848279563                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.4718634149498457E7
6597703848279563                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=95699
6597703848279563                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-27 10:51:49
6598830758632447                column=lookup_card_family:score, timestamp=1684659024744, value=293
6598830758632447                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.2227949982601807E7
6598830758632447                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=19421
6598830758632447                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-30 00:18:34
6599900931314251                column=lookup_card_family:score, timestamp=1684659024744, value=297
6599900931314251                column=lookup_card_family:ucl, timestamp=1684659024744, value=1.2121408572464656E7
6599900931314251                column=lookup_transaction_family:postcode, timestamp=1684659024744, value=97423
6599900931314251                column=lookup_transaction_family:transaction_dt, timestamp=1684659024744, value=2018-01-31 11:25:16
999 row(s) in 2.4440 seconds

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