Loading the Lookup Table

Commands to load the relevant data into the Lookup Table

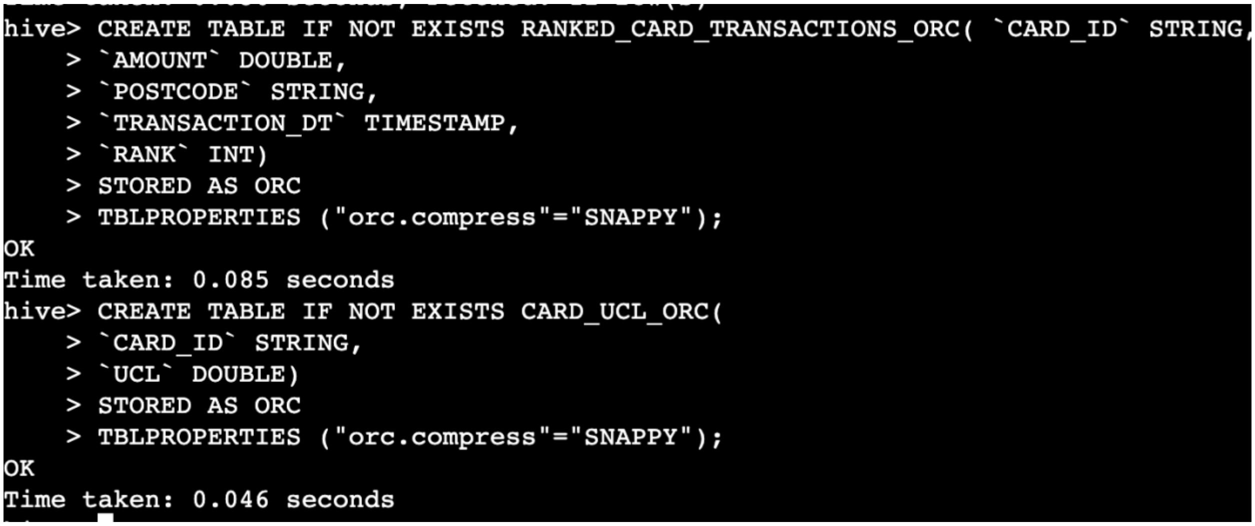
We have created 2 different tables for doing this task **–**

1. **RANKED_CARD_TRANSACTIONS_ORC –** this table stores last 10 transactions for each card ID
2. **CARD_UCL_ORC –** this table stores the upper card limit 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 CARD_UCL_ORC(
‘CARD_ID’ STRING,
‘UCL’ DOUBLE)
STORED AS ORC
TBLPROPERTIES ("orc.compress"="SNAPPY");

SCREENSHOTS :



```
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: 0.085 seconds
hive> CREATE TABLE IF NOT EXISTS CARD_UCL_ORC(
    > `CARD_ID` STRING,
    > `UCL` DOUBLE)
    > STORED AS ORC
    > TBLPROPERTIES ("orc.compress"="SNAPPY");
OK
Time taken: 0.046 seconds
```

LOADING DATA INTO THE ABOVE 2 TABLES

To fill **RANKED_CARD_TRANSACTIONS_ORC** , we will use the table **CARD_TRANSACTIONS_ORC**. For each genuine transaction, we find the latest 10 transactions and store them in **RANKED_CARD_TRANSACTIONS_ORC.**

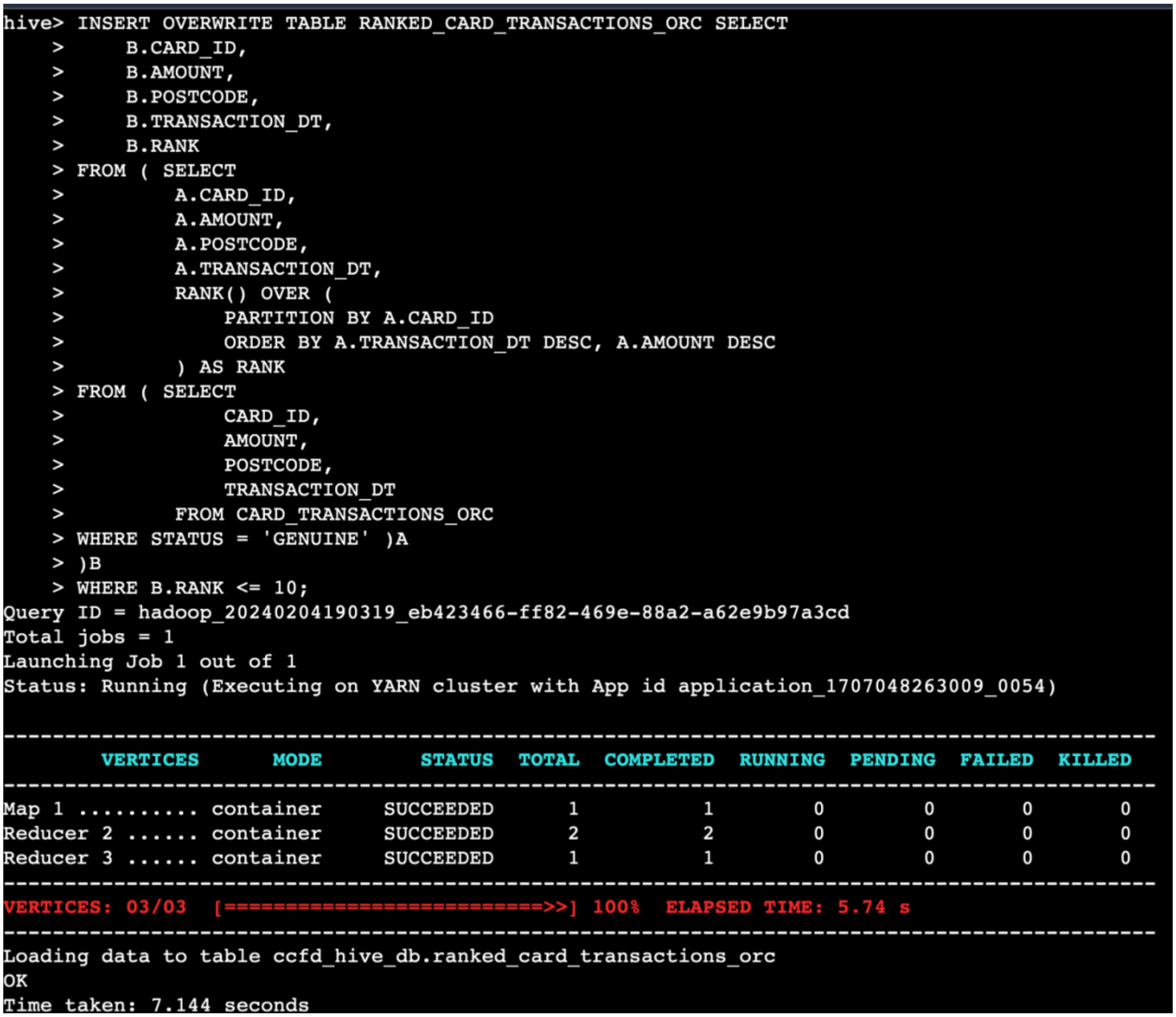
```
INSERT OVERWRITE TABLE RANKED_CARD_TRANSACTIONS_ORC SELECT
  B.CARD_ID,
  B.AMOUNT,
  B.POSTCODE,
  B.TRANSACTION_DT,
  B.RANK
FROM ( SELECT
    A.CARD_ID,
    A.AMOUNT,
    A.POSTCODE,
    A.TRANSACTION_DT,
    RANK() OVER (
      PARTITION BY A.CARD_ID
      ORDER BY A.TRANSACTION_DT DESC, A.AMOUNT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

To fill **CARD_UCL_ORC**, we will use the **RANKED_CARD_TRANSACTIONS_ORC**. For each card_id, we first calculate average amount and standard deviation. Then use this to calculate the Upper card limit(UCL) for each card.

```
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
```

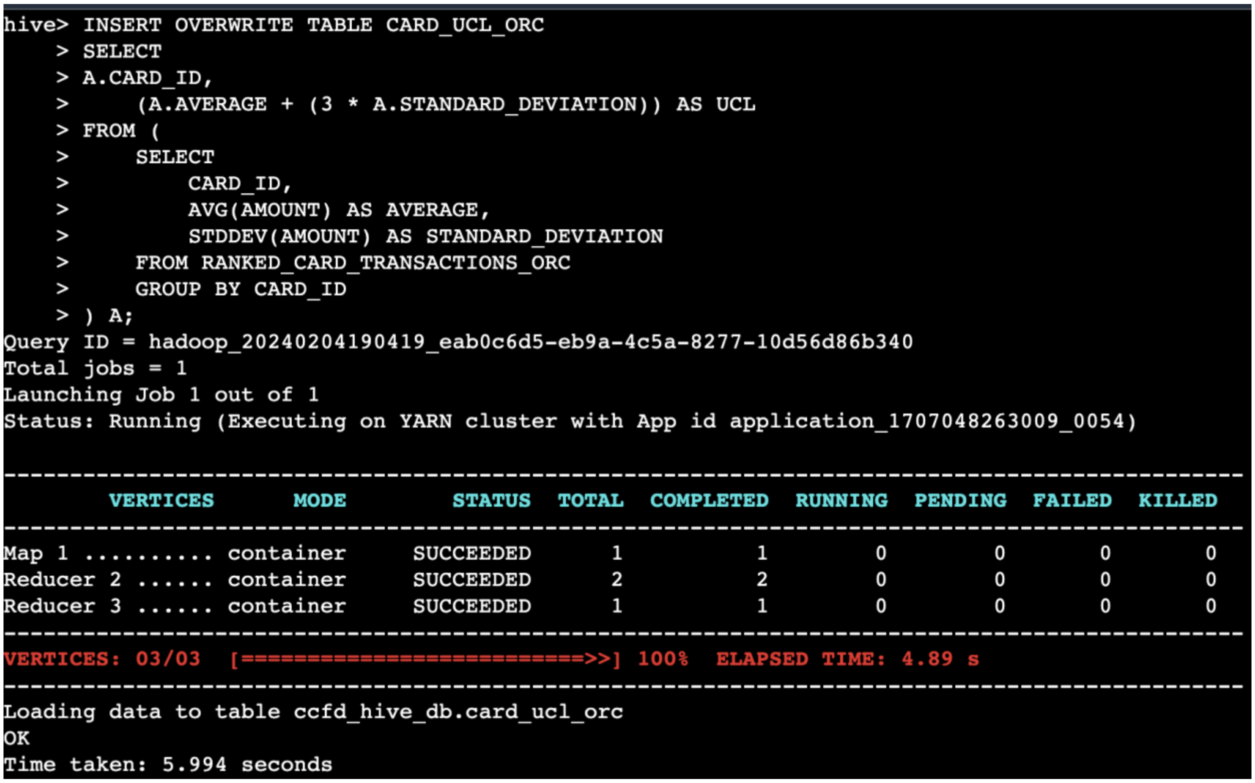
```
        FROM RANKED_CARD_TRANSACTIONS_ORC
        GROUP BY CARD_ID
) A;
```

SCREENSHOTS:



```
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, A.AMOUNT DESC
    >         ) AS RANK
    > FROM ( SELECT
    >             CARD_ID,
    >             AMOUNT,
    >             POSTCODE,
    >             TRANSACTION_DT
    >         FROM CARD_TRANSACTIONS_ORC
    > WHERE STATUS = 'GENUINE' )A
    > )B
    > WHERE B.RANK <= 10;
Query ID = hadoop_20240204190319_eb423466-ff82-469e-88a2-a62e9b97a3cd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1707048263009_0054)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.74 s
----------------------------------------------------------------------------------------------
Loading data to table ccfd_hive_db.ranked_card_transactions_orc
OK
Time taken: 7.144 seconds
```



```
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_20240204190419_eab0c6d5-eb9a-4c5a-8277-10d56d86b340
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1707048263009_0054)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 4.89 s
----------------------------------------------------------------------------------------------
Loading data to table ccfd_hive_db.card_ucl_orc
OK
Time taken: 5.994 seconds
```

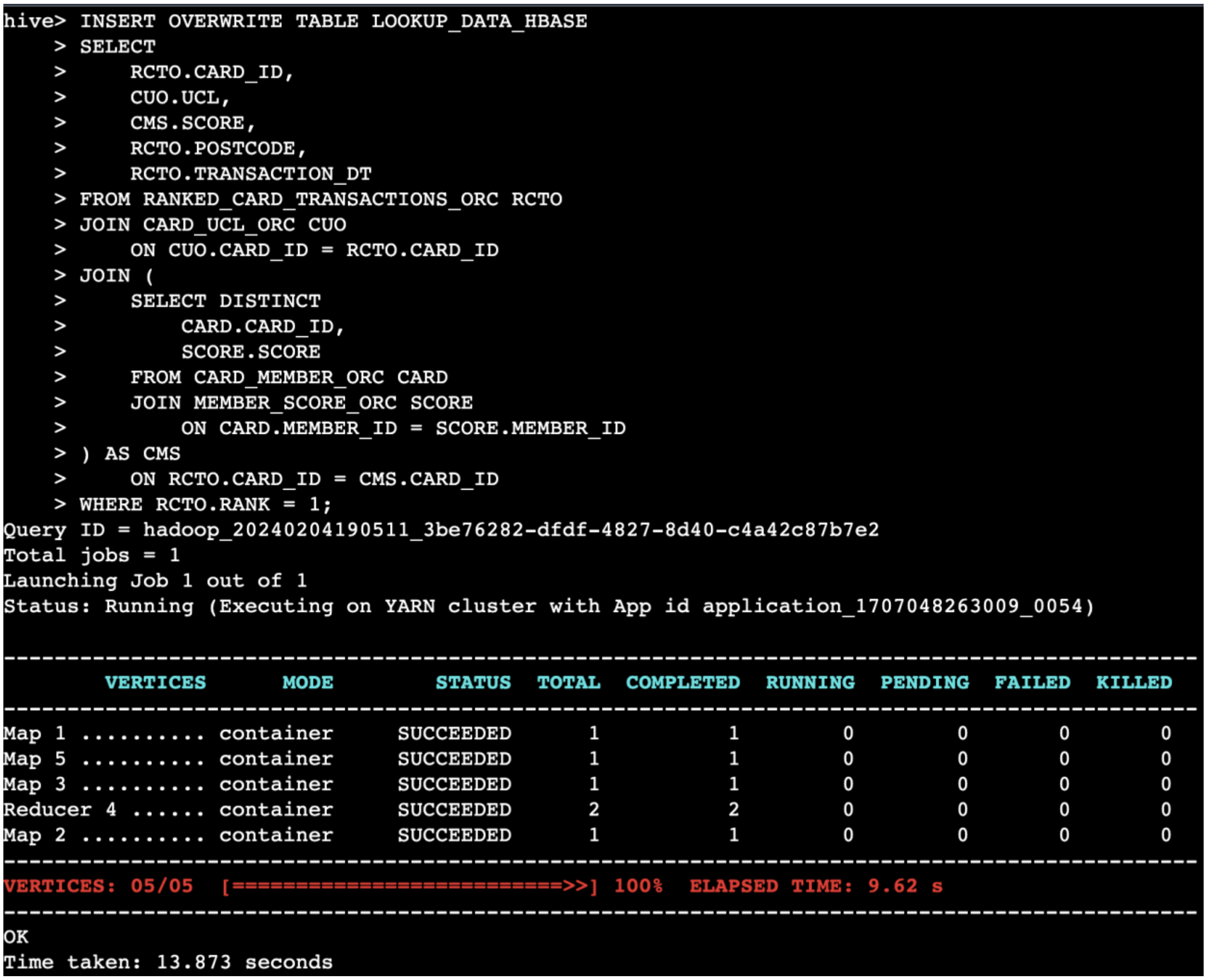
LOADING DATA INTO LOOKUP TABLE:

For lookup table data loading, we use the above 2 created tables, join them on card_id and then join with card_member table and member_score 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;

SCREENSHOT:

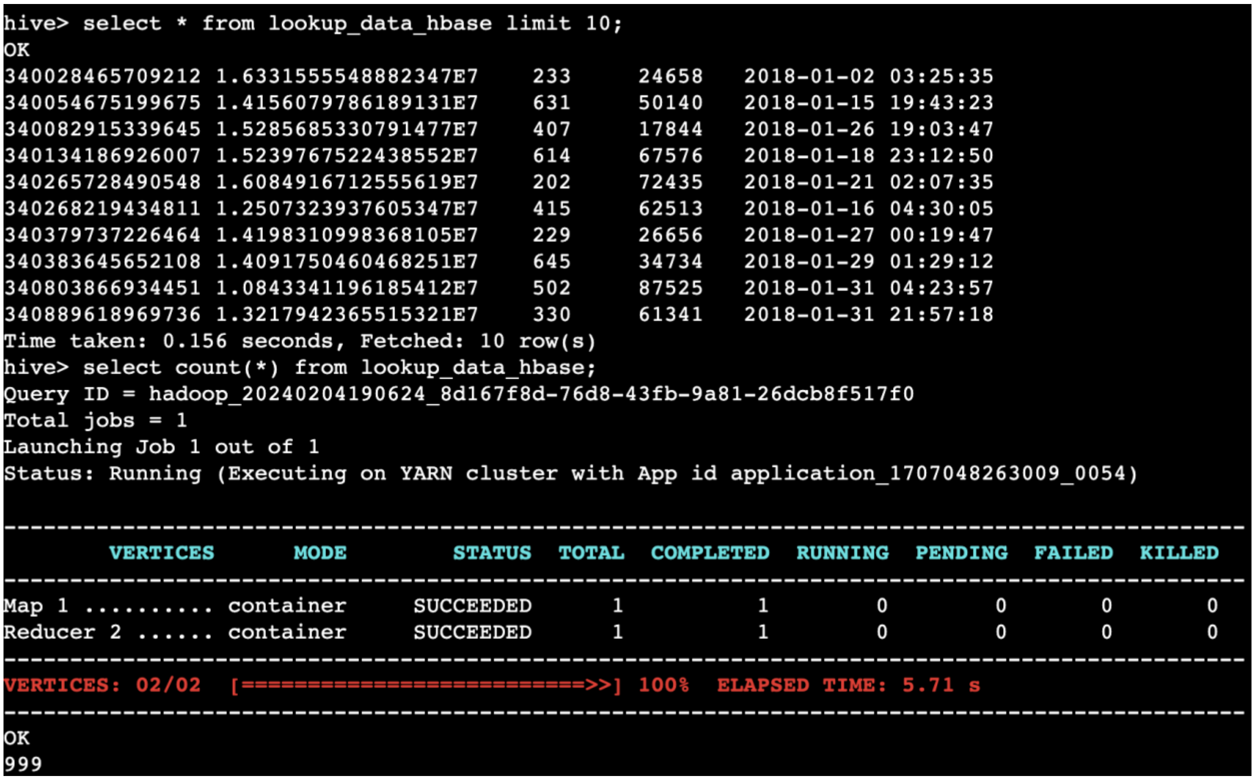


```
hive> INSERT OVERWRITE TABLE LOOKUP_DATA_HBASE
    > SELECT
    >     RCTO.CARD_ID,
    >     CUO.UCL,
    >     CMS.SCORE,
    >     RCTO.POSTCODE,
    >     RCTO.TRANSACTION_DT
    > FROM RANKED_CARD_TRANSACTIONS_ORC RCTO
    > JOIN CARD_UCL_ORC CUO
    >     ON CUO.CARD_ID = RCTO.CARD_ID
    > 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;
Query ID = hadoop_20240204190511_3be76282-dfdf-4827-8d40-c4a42c87b7e2
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1707048263009_0054)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      2          2        0        0       0       0
Map 2 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 9.62 s
----------------------------------------------------------------------------------------------
OK
Time taken: 13.873 seconds
```

Table content in hive:

Select * from lookup_data_hbase limit 10;



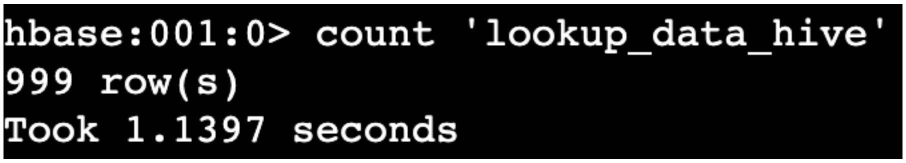
```
hive> select * from lookup_data_hbase limit 10;
OK
340028465709212 1.6331555548882347E7    233     24658   2018-01-02 03:25:35
340054675199675 1.4156079786189131E7    631     50140   2018-01-15 19:43:23
340082915339645 1.5285685330791477E7    407     17844   2018-01-26 19:03:47
340134186926007 1.5239767522438552E7    614     67576   2018-01-18 23:12:50
340265728490548 1.6084916712555619E7    202     72435   2018-01-21 02:07:35
340268219434811 1.2507323937605347E7    415     62513   2018-01-16 04:30:05
340379737226464 1.4198310998368105E7    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: 0.156 seconds, Fetched: 10 row(s)
hive> select count(*) from lookup_data_hbase;
Query ID = hadoop_20240204190624_8d167f8d-76d8-43fb-9a81-26dcb8f517f0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1707048263009_0054)

----------------------------------------------------------------------------------------------
        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: 5.71 s
----------------------------------------------------------------------------------------------
OK
999
```

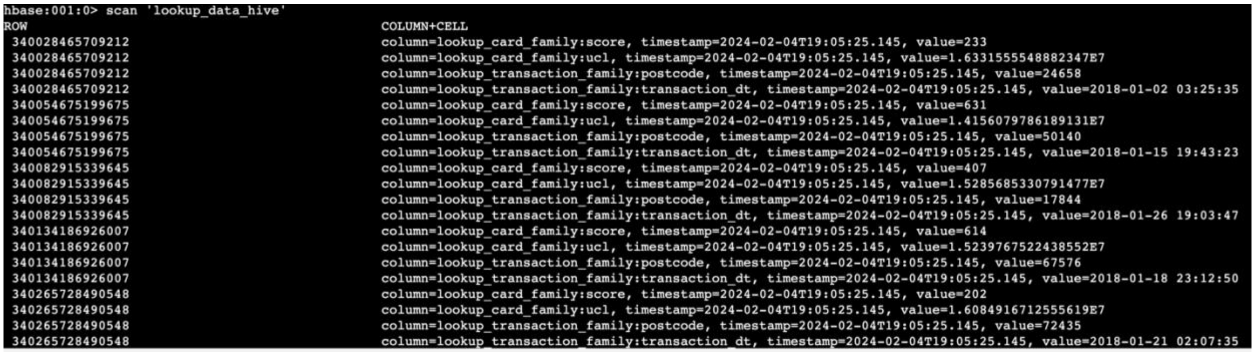
Table content in HBase:

count ‘select * from lookup_data_hive’ => to give row count

scan ‘lookup_data_hive’ => to give rows from HBase table



```
hbase:001:0> count 'lookup_data_hive'
999 row(s)
Took 1.1397 seconds
```



```
hbase:001:0> scan 'lookup_data_hive'
ROW                                 COLUMN+CELL
 340028465709212                    column=lookup_card_family:score, timestamp=2024-02-04T19:05:25.145, value=233
 340028465709212                    column=lookup_card_family:ucl, timestamp=2024-02-04T19:05:25.145, value=1.6331555548882347E7
 340028465709212                    column=lookup_transaction_family:postcode, timestamp=2024-02-04T19:05:25.145, value=24658
 340028465709212                    column=lookup_transaction_family:transaction_dt, timestamp=2024-02-04T19:05:25.145, value=2018-01-02 03:25:35
 340054675199675                    column=lookup_card_family:score, timestamp=2024-02-04T19:05:25.145, value=631
 340054675199675                    column=lookup_card_family:ucl, timestamp=2024-02-04T19:05:25.145, value=1.4156079786189131E7
 340054675199675                    column=lookup_transaction_family:postcode, timestamp=2024-02-04T19:05:25.145, value=50140
 340054675199675                    column=lookup_transaction_family:transaction_dt, timestamp=2024-02-04T19:05:25.145, value=2018-01-15 19:43:23
 340082915339645                    column=lookup_card_family:score, timestamp=2024-02-04T19:05:25.145, value=407
 340082915339645                    column=lookup_card_family:ucl, timestamp=2024-02-04T19:05:25.145, value=1.5285685330791477E7
 340082915339645                    column=lookup_transaction_family:postcode, timestamp=2024-02-04T19:05:25.145, value=17844
 340082915339645                    column=lookup_transaction_family:transaction_dt, timestamp=2024-02-04T19:05:25.145, value=2018-01-26 19:03:47
 340134186926007                    column=lookup_card_family:score, timestamp=2024-02-04T19:05:25.145, value=614
 340134186926007                    column=lookup_card_family:ucl, timestamp=2024-02-04T19:05:25.145, value=1.5239767522438552E7
 340134186926007                    column=lookup_transaction_family:postcode, timestamp=2024-02-04T19:05:25.145, value=67576
 340134186926007                    column=lookup_transaction_family:transaction_dt, timestamp=2024-02-04T19:05:25.145, value=2018-01-18 23:12:50
 340265728490548                    column=lookup_card_family:score, timestamp=2024-02-04T19:05:25.145, value=202
 340265728490548                    column=lookup_card_family:ucl, timestamp=2024-02-04T19:05:25.145, value=1.6084916712555619E7
 340265728490548                    column=lookup_transaction_family:postcode, timestamp=2024-02-04T19:05:25.145, value=72435
 340265728490548                    column=lookup_transaction_family:transaction_dt, timestamp=2024-02-04T19:05:25.145, value=2018-01-21 02:07:35
```