Scripts Execution

Screenshots of the execution of the scripts written

1. Sqoop Data Ingestion from AWS RDS to HDFS:

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat data_ingestion.sh
#!/bin/bash
# Sqoop Import Card Member and Member Score data from AWS RDS to HDFS
# Execution Command: ./sqoop_data_ingestion.sh <AWS RDS Connection String> <database> <username> <password>

# AWS RDS Credentials
rds_connection=$1
database=$2
username=$3
password=$4


# Sqoop Import - Card Member table
sqoop import \
--connect jdbc:mysql://${rds_connection}/${database} \
--username ${username} \
--password ${password} \
--table card_member \
--warehouse-dir /user/hadoop/cred_financials_data \
--delete-target-dir \
--num-mappers 1 \
--compress

# Sqoop Import - Member Score table
sqoop import \
--connect jdbc:mysql://${rds_connection}/${database} \
--username ${username}  \
--password ${password} \
--table member_score \
--warehouse-dir /user/hadoop/cred_financials_data \
--delete-target-dir \
--num-mappers 1 \
--compress
```

Execution command:

/home/hadoop/cred_financials_data/script/data_ingestion.sh upgradawsrds1.cyaielc9bmnf.us-east-1.rds.amazonaws.com cred_financials_data upgraduser upgraduser

Card Member:

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=34628
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=281952
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=2937
                Total vcore-milliseconds taken by all map tasks=2937
                Total megabyte-milliseconds taken by all map tasks=9022464
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=68
                CPU time spent (ms)=2170
                Physical memory (bytes) snapshot=284667904
                Virtual memory (bytes) snapshot=4625903616
                Total committed heap usage (bytes)=243793920
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=34628
22/12/17 20:33:20 INFO mapreduce.ImportJobBase: Transferred 33.8164 KB in 16.6082 seconds (2.0361 KB/sec)
22/12/17 20:33:20 INFO mapreduce.ImportJobBase: Retrieved 999 records.
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
[hadoop@ip-172-31-80-45 script]$ hadoop fs -ls /user/hadoop/cred_financials_data/card_member
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2022-12-17 20:33 /user/hadoop/cred_financials_data/card_member/_SUCCESS
-rw-r--r--   1 hadoop hadoop      34628 2022-12-17 20:33 /user/hadoop/cred_financials_data/card_member/part-m-00000.gz
```

Member Score:

```
22/12/17 20:33:34 INFO mapreduce.Job: Job job_1671300494190_0020 running in uber mode : false
22/12/17 20:33:34 INFO mapreduce.Job:  map 0% reduce 0%
22/12/17 20:33:39 INFO mapreduce.Job:  map 100% reduce 0%
22/12/17 20:33:40 INFO mapreduce.Job: Job job_1671300494190_0020 completed successfully
22/12/17 20:33:40 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=189638
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=10186
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=276576
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=2881
                Total vcore-milliseconds taken by all map tasks=2881
                Total megabyte-milliseconds taken by all map tasks=8850432
        Map-Reduce Framework
                Map input records=999
                Map output records=999
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=67
                CPU time spent (ms)=1900
                Physical memory (bytes) snapshot=275959808
                Virtual memory (bytes) snapshot=4625100800
                Total committed heap usage (bytes)=243269632
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=10186
22/12/17 20:33:40 INFO mapreduce.ImportJobBase: Transferred 9.9473 KB in 14.5031 seconds (702.3317 bytes/sec)
22/12/17 20:33:40 INFO mapreduce.ImportJobBase: Retrieved 999 records.
[hadoop@ip-172-31-80-45 script]$
```

```
[hadoop@ip-172-31-80-45 script]$ hadoop fs -ls /user/hadoop/cred_financials_data/member_score
Found 2 items
-rw-r--r--   1 hadoop hadoop          0 2022-12-17 20:33 /user/hadoop/cred_financials_data/member_score/_SUCCESS
-rw-r--r--   1 hadoop hadoop      10186 2022-12-17 20:33 /user/hadoop/cred_financials_data/member_score/part-m-00000.gz
```

2. Hive database creation:

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat create_database.hql
-- Create a database
create database cred_financials_data;

exit;
[hadoop@ip-172-31-80-45 script]$
```

Execution command:

hive -f /home/hadoop/cred_financials_data/script/create_database.hql

```
[hadoop@ip-172-31-80-45 script]$ hive -f /home/hadoop/cred_financials_data/script/create_database.hql

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.762 seconds
[hadoop@ip-172-31-80-45 script]$
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
hive> show databases;
OK
database_name
cred_financials_data
default
Time taken: 0.006 seconds, Fetched: 2 row(s)
hive>
```

3. **Card Member hive table:**

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat card_member.hql
USE cred_financials_data;

-- Enforce Hive Bucketing
SET hive.enforce.bucketing=true;

-- Create card member staging table pointing to hdfs imported from MySQL
CREATE TABLE IF NOT EXISTS card_member_stg(
card_id BIGINT,
member_id BIGINT,
member_joining_dt TIMESTAMP,
card_purchase_dt STRING,
country STRING,
city STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/cred_financials_data/card_member';

-- Create Hive bucketing table (External) on card member for join optimization
CREATE EXTERNAL TABLE IF NOT EXISTS card_member(
card_id BIGINT,
member_id BIGINT,
member_joining_dt TIMESTAMP,
card_purchase_dt STRING,
country STRING,
city STRING
)
CLUSTERED BY (card_id, member_id) INTO 8 BUCKETS
STORED AS PARQUET
LOCATION '/user/hadoop/cred_financials_data/card_member_bucket';

-- Insert the card member data from staging to bucketing table
INSERT OVERWRITE TABLE card_member SELECT * FROM card_member_stg;

-- Drop staging table
DROP TABLE card_member_stg;

exit;
[hadoop@ip-172-31-80-45 script]$
```

Execution command:

hive -f /home/hadoop/cred_financials_data/script/card_member.hql

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
[hadoop@ip-172-31-80-45 script]$ hive -f /home/hadoop/cred_financials_data/script/card_member.hql

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.889 seconds
OK
Time taken: 0.207 seconds
OK
Time taken: 0.05 seconds
Query ID = hadoop_20221217204318_37037c89-9257-470f-bc8b-b2ad7e0116cc
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671300494190_0023)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      8          8        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 9.66 s
--------------------------------------------------------------------------------
Loading data to table cred_financials_data.card_member
OK
Time taken: 18.866 seconds
OK
Time taken: 0.116 seconds
[hadoop@ip-172-31-80-45 script]$
```

```
hive> show tables;
OK
tab_name
card_member
Time taken: 0.01 seconds, Fetched: 1 row(s)
hive> select count(*) from card_member;
OK
_c0
999
Time taken: 0.086 seconds, Fetched: 1 row(s)
hive>
```

4. Member Score Hive table:

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat member_score.hql
USE cred_financials_data;

-- Enforce Hive Bucketing
SET hive.enforce.bucketing=true;

-- Create member score staging table pointing to hdfs imported from MySQL
CREATE TABLE IF NOT EXISTS member_score_stg(
member_id BIGINT,
score INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/cred_financials_data/member_score';

-- Create Hive bucketing table (External) on member score for join optimization
CREATE EXTERNAL TABLE IF NOT EXISTS member_score(
member_id BIGINT,
score INT
)
CLUSTERED BY (member_id) INTO 8 BUCKETS
STORED AS PARQUET
LOCATION '/user/hadoop/cred_financials_data/member_score_bucket';

-- Insert the member score data from staging to bucketing table
INSERT OVERWRITE TABLE member_score SELECT * FROM member_score_stg;

-- Drop staging table
DROP TABLE member_score_stg;

exit;

[hadoop@ip-172-31-80-45 script]$
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

Execution command:

hive -f /home/hadoop/cred_financials_data/script/member_score.hql

```
[hadoop@ip-172-31-80-45 script]$ hive -f /home/hadoop/cred_financials_data/script/member_score.hql

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.86 seconds
OK
Time taken: 0.19 seconds
OK
Time taken: 0.046 seconds
Query ID = hadoop_20221217204629_7ffdf39d-3cc9-4f47-a89a-67bc7ae1e527
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671300494190_0025)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      8          8        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 10.48 s
--------------------------------------------------------------------------------
Loading data to table cred_financials_data.member_score
OK
Time taken: 21.108 seconds
OK
Time taken: 0.056 seconds
[hadoop@ip-172-31-80-45 script]$
```

```
hive> select count(*) from member_score;
OK
_c0
999
Time taken: 0.067 seconds, Fetched: 1 row(s)
hive>
```

5. **Load Card Transactions History – Hive and HBase tables:**

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat load_transactions_nosql.sh
#!/bin/bash
# Create Hive and HBase tables for card transactions historical data and load data in it

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

hive -f /home/hadoop/cred_financials_data/script/card_transactions_history.hql
[hadoop@ip-172-31-80-45 script]$
```

```
[hadoop@ip-172-31-80-45 script]$ cat card_transactions_history.hql
USE cred_financials_data;

-- Enforce Hive Bucketing
SET hive.enforce.bucketing=true;

-- Create card transactions source table and load the data from the CSV file provided (historical data)
CREATE TABLE IF NOT EXISTS card_transactions_src(
card_id BIGINT,
member_id BIGINT,
amount INT,
postcode INT,
pos_id BIGINT,
transaction_dt STRING,
status STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/cred_financials_data/card_transactions_src'
TBLPROPERTIES ('skip.header.line.count'='1');

LOAD DATA LOCAL INPATH '/home/hadoop/cred_financials_data/data/card_transactions.csv' INTO TABLE card_transactions_src;

-- Perform type conversion for transaction date attribute and load the data in staging table
CREATE TABLE IF NOT EXISTS card_transactions_stg(
card_id BIGINT,
member_id BIGINT,
amount INT,
postcode INT,
pos_id BIGINT,
transaction_dt TIMESTAMP,
status STRING
)
location '/user/hadoop/cred_financials_data/card_transactions_stg';
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
INSERT INTO TABLE card_transactions_stg
SELECT card_id,
member_id,
amount,
postcode,
pos_id,
CAST(FROM_UNIXTIME(UNIX_TIMESTAMP(transaction_dt, 'dd-MM-yyyy HH:mm:ss'),'yyyy-MM-dd HH:mm:ss')AS TIMESTAMP) AS transaction_dt,
status
FROM card_transactions_src;

-- Create Card Transactions Hive table (External) with HBase Integration
CREATE EXTERNAL TABLE IF NOT EXISTS card_transactions(
transaction_key STRING,
card_id BIGINT,
member_id BIGINT,
amount INT,
postcode INT,
pos_id BIGINT,
transaction_dt TIMESTAMP,
status STRING
)
CLUSTERED BY (card_id, member_id)INTO 16 BUCKETS
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ('hbase.columns.mapping'=':key, transaction_data:card_id, transaction_data:member_id, transaction_data:amount, transaction_data:postcode, transa
ction_data:pos_id, transaction_data:transaction_dt, transaction_data:status')
TBLPROPERTIES ('hbase.table.name' = 'card_transactions');

-- Insert the staging data to Hive-HBase table
INSERT INTO TABLE card_transactions
SELECT CONCAT_WS('|', CAST(card_id AS STRING), CAST(member_id AS STRING), CAST(amount AS STRING), CAST(postcode AS STRING), CAST(pos_id AS STRING), CAST(transaction_
dt AS STRING), status) AS transaction_key,
card_id,
member_id,
amount,
postcode,
pos_id,
transaction_dt,
status
FROM card_transactions_stg;
```

```
-- Drop all intermediate hive tables
DROP TABLE card_transactions_stg;
DROP TABLE card_transactions_src;

exit;
[hadoop@ip-172-31-80-45 script]$
```

Execution command:

/home/hadoop/cred_financials_data/script/load_transactions_nosql.sh

```
Hbase::Table - card_transactions

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.808 seconds
OK
Time taken: 0.22 seconds
Loading data to table cred_financials_data.card_transactions_src
OK
Time taken: 0.8 seconds
OK
Time taken: 0.05 seconds
Query ID = hadoop_20221217205136_ec39414e-9968-4070-bdc9-8c78291394e5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671300494190_0027)

--------------------------------------------------------------------------------
        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: 5.17 s
--------------------------------------------------------------------------------
Loading data to table cred_financials_data.card_transactions_stg
OK
Time taken: 14.411 seconds
OK
Time taken: 0.916 seconds
Query ID = hadoop_20221217205151_779495a6-be43-4001-a211-031d88d93156
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671300494190_0027)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     16         16        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.62 s
--------------------------------------------------------------------------------
OK
Time taken: 13.505 seconds
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

Hive table

```
hive> select count(*) from card_transactions;
Query ID = hadoop_20221217205257_1d4fae6e-620d-45e7-908e-619f55db7ea7
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_1671300494190_0028)

--------------------------------------------------------------------------------
        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
_c0
53292
Time taken: 14.215 seconds, Fetched: 1 row(s)
hive>
```

HBase table

```
Current count: 41000, row: 6011525010455848|5385552135011981|9400038|28719|1461984749453281|2017-11-17 22:28:00|GENUINE
Current count: 42000, row: 6011782857327719|969966222267715|1982856|17061|841171694848680|2017-09-26 11:42:09|GENUINE
Current count: 43000, row: 6221796595498984|617313952879129|1551821|35674|834075578964661|2017-01-10 18:09:06|GENUINE
Current count: 44000, row: 6224271253849917|815187737253702|140177|83849|7131165095637777|2018-01-25 18:29:05|GENUINE
Current count: 45000, row: 6225606551069826|284643419452603|7500507|22901|606202448444893|2017-12-05 03:01:57|GENUINE
Current count: 46000, row: 6228733641419063|610330639594612|3051758|39169|309527290827592|2016-07-14 03:34:45|GENUINE
Current count: 47000, row: 6447877814927926|907972949998745|923252|56685|992747968210744|2018-01-11 00:00:00|GENUINE
Current count: 48000, row: 6461356425954109|739325996860756|8515416|10801|531820328204383|2017-12-25 04:03:59|GENUINE
Current count: 49000, row: 6480152634975473|439083998526821|9304601|71047|181711798421306|2018-01-08 19:11:10|GENUINE
Current count: 50000, row: 6505080237250161|615754567307150|8801408|27341|486982167852820|2017-12-05 17:04:37|GENUINE
Current count: 51000, row: 6544876671165176|269098610760255|4604408|14510|536497882467098|2018-01-21 07:23:36|GENUINE
Current count: 52000, row: 6574255180086418|891702243060747|5991679|33097|891586971848958|2016-04-22 01:13:11|GENUINE
Current count: 53000, row: 6595814135833988|236864426408837|3243199|56162|834307885260185|2016-10-08 23:28:28|GENUINE
53292 row(s) in 1.6040 seconds

=> 53292
hbase(main):023:0>
```

6. Create Card Lookup – Hive and HBase tables:

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat create_lookup_nosql.sh
#!/bin/bash
# Create Hive and HBase tables of Card Lookup

echo "create 'card_lookup', 'lkp_info'" | hbase shell -n

hive -f /home/hadoop/cred_financials_data/script/card_lookup_ddl.hql
[hadoop@ip-172-31-80-45 script]$
```

```
[hadoop@ip-172-31-80-45 script]$ cat card_lookup_ddl.hql
USE cred_financials_data;

-- Enforce Hive Bucketing
SET hive.enforce.bucketing=true;

-- Create Card Lookup Hive table (External) with HBase Integration
CREATE EXTERNAL TABLE IF NOT EXISTS card_lookup(
card_id BIGINT,
ucl DOUBLE,
postcode INT,
transaction_dt TIMESTAMP,
credit_score INT
)
CLUSTERED BY (card_id)INTO 8 BUCKETS
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ('hbase.columns.mapping'=':key, lkp_info:ucl, lkp_info:postcode, lkp_info:transaction_dt, lkp_info:credit_score')
TBLPROPERTIES ('hbase.table.name' = 'card_lookup');

exit;
[hadoop@ip-172-31-80-45 script]$
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

Execution command:

/home/hadoop/cred_financials_data/script/create_lookup_nosql.sh

```
Hbase::Table - card_lookup

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.771 seconds
OK
Time taken: 1.697 seconds
[hadoop@ip-172-31-80-45 script]$
```

Hive table

```
hive> describe formatted card_lookup;
OK
col_name        data_type       comment
# col_name              data_type               comment

card_id                 bigint
ucl                     double
postcode                int
transaction_dt          timestamp
credit_score            int

# Detailed Table Information
Database:               cred_financials_data
Owner:                  hadoop
CreateTime:             Sat Dec 17 20:56:58 UTC 2022
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://ip-172-31-80-45.ec2.internal:8020/user/hive/warehouse/cred_financials_data.db/card_lookup
Table Type:             EXTERNAL_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        EXTERNAL                TRUE
        hbase.table.name        card_lookup
        numFiles                0
        numRows                 0
        rawDataSize             0
        storage_handler         org.apache.hadoop.hive.hbase.HBaseStorageHandler
        totalSize               0
        transient_lastDdlTime   1671310618

# Storage Information
SerDe Library:          org.apache.hadoop.hive.hbase.HBaseSerDe
InputFormat:            null
OutputFormat:           null
Compressed:             No
Num Buckets:            8
Bucket Columns:         [card_id]
Sort Columns:           []
Storage Desc Params:
        hbase.columns.mapping   :key, lkp_info:ucl, lkp_info:postcode, lkp_info:transaction_dt, lkp_info:credit_score
        serialization.format    1
Time taken: 0.023 seconds, Fetched: 38 row(s)
```

HBase table

```
hbase(main):024:0> describe 'card_lookup'
Table card_lookup is ENABLED
card_lookup
COLUMN FAMILIES DESCRIPTION
{NAME => 'lkp_info', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COM
PRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.0080 seconds

hbase(main):025:0>
```

7. **Load Card Lookup metrics – Hive and HBase tables:**

Script:

```
[hadoop@ip-172-31-80-45 script]$ cat lookup_metrics_calculate_nosql.sh
#!/bin/bash
# Prepare card lookup data using Spark SQL and load it to card lookup Hive and HBase tables

spark-submit /home/hadoop/cred_financials_data/script/card_lookup_preprocessing.py

hive -f /home/hadoop/cred_financials_data/script/card_lookup_insert.hql
[hadoop@ip-172-31-80-45 script]$
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
[hadoop@ip-172-31-80-45 script]$ cat card_lookup_preprocessing.py
# Import necessary PySpark libraries
import pyspark
from pyspark.sql import SparkSession

# Create a Spark Session with Hive support
spark = SparkSession \
        .builder \
        .appName('Credit Card Lookup data preparation') \
        .enableHiveSupport() \
        .getOrCreate()

# Set log level to ERROR
spark.sparkContext.setLogLevel('ERROR')

# Prepare the card lookup data from Card Transactions and Member Score Hive tables using Spark SQL and store the results in a temporary view
spark.sql('''
WITH transaction_details AS
(
    SELECT
        card_id,
        member_id,
        amount,
        postcode,
        transaction_dt,
        RANK() OVER(PARTITION BY card_id ORDER BY transaction_dt DESC) AS txn_rank
    FROM
        cred_financials_data.card_transactions
)
SELECT
    card_id,
    ROUND(AVG(amount) + 3 *MAX(std_dev), 0) AS ucl,
    FIRST_VALUE(postcode) OVER(PARTITION BY card_id ORDER BY (SELECT 1)) AS postcode,
    MAX(transaction_dt) AS transaction_dt,
    credit_score
FROM
    (
        SELECT
            txn.card_id,
            txn.amount,
            FIRST_VALUE(txn.postcode) OVER(PARTITION BY card_id ORDER BY txn.txn_rank) AS postcode,
            txn.transaction_dt,
            mem.score as credit_score,
            ROUND(STDDEV(txn.amount) OVER(PARTITION BY card_id ORDER BY (SELECT 1)), 0) AS std_dev
        member_id,
        amount,
        postcode,
        transaction_dt,
        RANK() OVER(PARTITION BY card_id ORDER BY transaction_dt DESC) AS txn_rank
    FROM
        cred_financials_data.card_transactions
)
SELECT
    card_id,
    ROUND(AVG(amount) + 3 *MAX(std_dev), 0) AS ucl,
    FIRST_VALUE(postcode) OVER(PARTITION BY card_id ORDER BY (SELECT 1)) AS postcode,
    MAX(transaction_dt) AS transaction_dt,
    credit_score
FROM
    (
        SELECT
            txn.card_id,
            txn.amount,
            FIRST_VALUE(txn.postcode) OVER(PARTITION BY card_id ORDER BY txn.txn_rank) AS postcode,
            txn.transaction_dt,
            mem.score as credit_score,
            ROUND(STDDEV(txn.amount) OVER(PARTITION BY card_id ORDER BY (SELECT 1)), 0) AS std_dev
        FROM
            transaction_details txn
            INNER JOIN cred_financials_data.member_score mem
                ON txn.member_id = mem.member_id
        WHERE
            txn.txn_rank <= 10
    ) a
GROUP BY
    card_id,
    postcode,
    credit_score
''').createOrReplaceTempView('card_lookup_tmp')


# Create a staging table in Hive and load the data from temporary view
spark.sql('CREATE TABLE cred_financials_data.card_lookup_stg AS SELECT * FROM card_lookup_tmp')

# Drop temporary view to release the memory
spark.catalog.dropTempView('card_lookup_tmp')

spark.stop()
[hadoop@ip-172-31-80-45 script]$
```

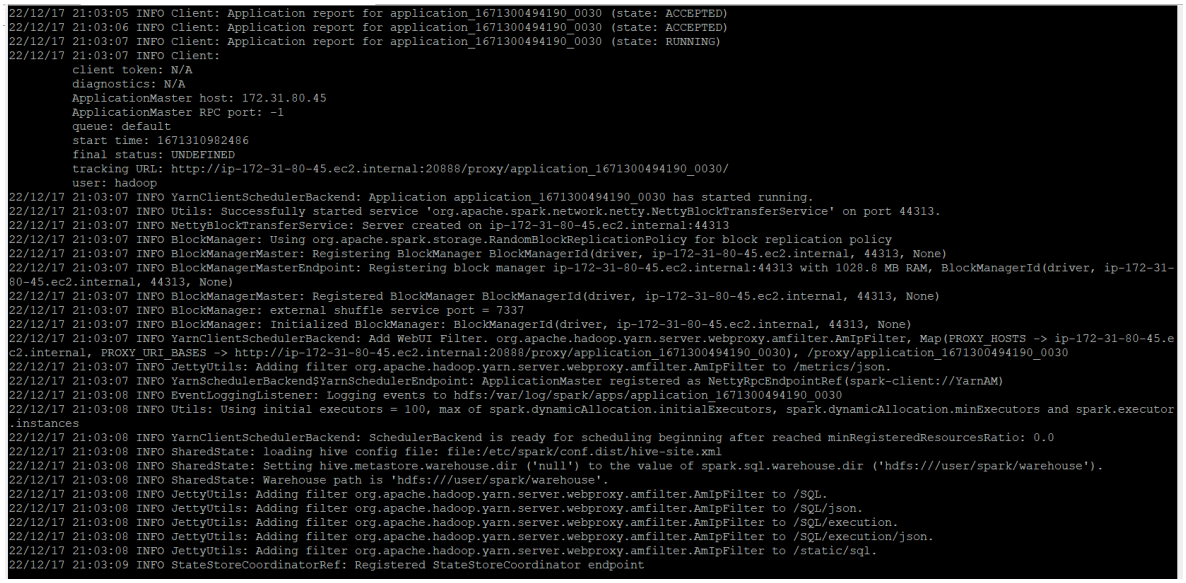
```
[hadoop@ip-172-31-80-45 script]$ cat card_lookup_insert.hql
USE cred_financials_data;

-- Insert the card lookup data from staging (which got prepared using Spark SQL) to bucketing table
INSERT OVERWRITE TABLE card_lookup SELECT * FROM card_lookup_stg;

-- Drop staging table
DROP TABLE card_lookup_stg;

exit;
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

Execution command:

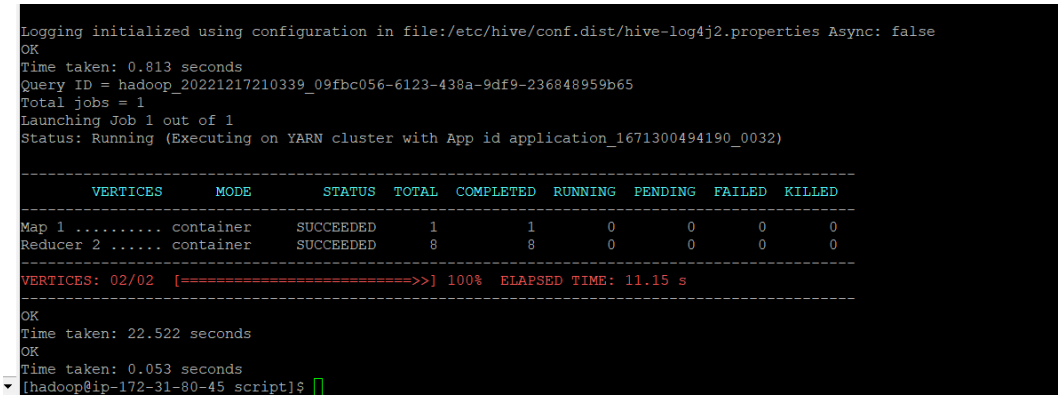
/home/hadoop/cred_financials_data/script/lookup_metrics_calculate_nosql.sh

PySpark logs



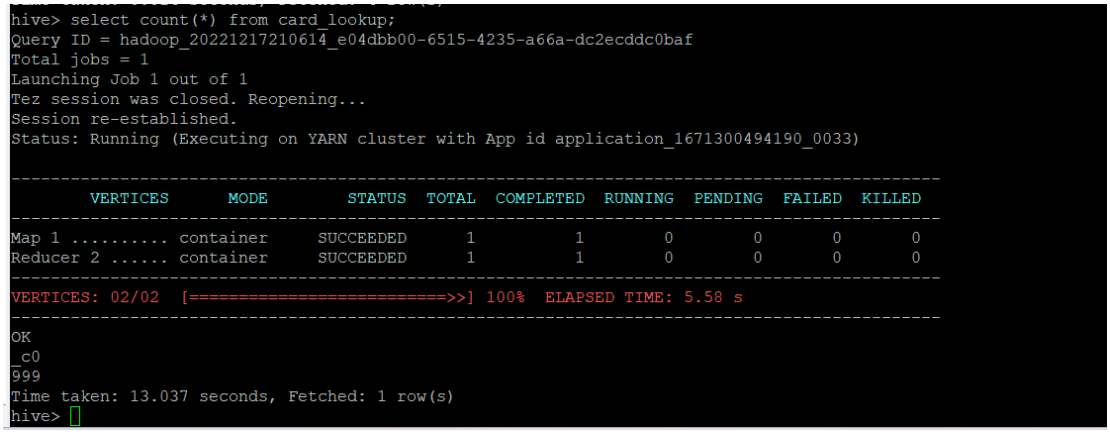
```
22/12/17 21:03:05 INFO Client: Application report for application_1671300494190_0030 (state: ACCEPTED)
22/12/17 21:03:06 INFO Client: Application report for application_1671300494190_0030 (state: ACCEPTED)
22/12/17 21:03:07 INFO Client: Application report for application_1671300494190_0030 (state: RUNNING)
22/12/17 21:03:07 INFO Client:
        client token: N/A
        diagnostics: N/A
        ApplicationMaster host: 172.31.80.45
        ApplicationMaster RPC port: -1
        queue: default
        start time: 1671310982486
        final status: UNDEFINED
        tracking URL: http://ip-172-31-80-45.ec2.internal:20888/proxy/application_1671300494190_0030/
        user: hadoop
22/12/17 21:03:07 INFO YarnClientSchedulerBackend: Application application_1671300494190_0030 has started running.
22/12/17 21:03:07 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44313.
22/12/17 21:03:07 INFO NettyBlockTransferService: Server created on ip-172-31-80-45.ec2.internal:44313
22/12/17 21:03:07 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
22/12/17 21:03:07 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, ip-172-31-80-45.ec2.internal, 44313, None)
22/12/17 21:03:07 INFO BlockManagerMasterEndpoint: Registering block manager ip-172-31-80-45.ec2.internal:44313 with 1028.8 MB RAM, BlockManagerId(driver, ip-172-31-
80-45.ec2.internal, 44313, None)
22/12/17 21:03:07 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ip-172-31-80-45.ec2.internal, 44313, None)
22/12/17 21:03:07 INFO BlockManager: external shuffle service port = 7337
22/12/17 21:03:07 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ip-172-31-80-45.ec2.internal, 44313, None)
22/12/17 21:03:07 INFO YarnClientSchedulerBackend: Add WebUI Filter. org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter, Map(PROXY_HOSTS -> ip-172-31-80-45.e
c2.internal, PROXY_URI_BASES -> http://ip-172-31-80-45.ec2.internal:20888/proxy/application_1671300494190_0030), /proxy/application_1671300494190_0030
22/12/17 21:03:07 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /metrics/json.
22/12/17 21:03:07 INFO YarnSchedulerBackend$YarnSchedulerEndpoint: ApplicationMaster registered as NettyRpcEndpointRef(spark-client://YarnAM)
22/12/17 21:03:08 INFO EventLoggingListener: Logging events to hdfs:/var/log/spark/apps/application_1671300494190_0030
22/12/17 21:03:08 INFO Utils: Using initial executors = 100, max of spark.dynamicAllocation.initialExecutors, spark.dynamicAllocation.minExecutors and spark.executor
.instances
22/12/17 21:03:08 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
22/12/17 21:03:08 INFO SharedState: loading hive config file: file:/etc/spark/conf.dist/hive-site.xml
22/12/17 21:03:08 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('hdfs:///user/spark/warehouse').
22/12/17 21:03:08 INFO SharedState: Warehouse path is 'hdfs:///user/spark/warehouse'.
22/12/17 21:03:08 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL.
22/12/17 21:03:08 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/json.
22/12/17 21:03:08 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution.
22/12/17 21:03:08 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /SQL/execution/json.
22/12/17 21:03:08 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /static/sql.
22/12/17 21:03:09 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
```

Hive script logs



```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 0.813 seconds
Query ID = hadoop_20221217210339_09fbc056-6123-438a-9df9-236848959b65
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1671300494190_0032)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      8          8        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.15 s
--------------------------------------------------------------------------------
OK
Time taken: 22.522 seconds
OK
Time taken: 0.053 seconds
[hadoop@ip-172-31-80-45 script]$
```

Hive table



```
hive> select count(*) from card_lookup;
Query ID = hadoop_20221217210614_e04dbb00-6515-4235-a66a-dc2ecddc0baf
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_1671300494190_0033)

--------------------------------------------------------------------------------
        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: 5.58 s
--------------------------------------------------------------------------------
OK
_c0
999
Time taken: 13.037 seconds, Fetched: 1 row(s)
hive>
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

HBase table

```
hbase(main):025:0> list
TABLE
card_lookup
card_transactions
2 row(s) in 0.0100 seconds

=> ["card_lookup", "card_transactions"]
hbase(main):026:0> count 'card_lookup'
999 row(s) in 0.0480 seconds

=> 999
hbase(main):027:0>
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved