

Management of Big Data and Tools – DS8003 – Fall 2016

Assignment 03

NAJLIS, BERNARDO - Student Number #500744793

Dataset:

1. We are going to use the Dataset : **hive2_dataset.zip** (The entire dataset is available here Dataset and Scripts -> Movielens & IMDB -> movielens.zip.)
2. Unzip the **hive2_dataset.zip**.
3. The zip contains 4 files: action, thriller, comedy, action_comedy_thriller
4. Copy files **action, thriller, comedy, action_comedy_thriller** to the virtual machine (**Filezilla**)
5. Copy files **action, thriller, comedy, action_comedy_thriller** from virtual machine into HDFS (**hadoop fs -put**)

Submission:

1. Submit both the hive commands and the results (copy it into a file and submit)

Example: if you submit a file called assignment1.txt. It should contain the following information for every question.

Question XX: (XX is the question number)

HiveQL: Select count () from u.data;*

Result: 100000

2. Submit using Assessment -> Dropbox -> Assignment 3: Advanced Hive

Assignment (10 points)

-- Partitioning data in hive

1. create table called movies_whole with 3 columns (movieid, movie_name, genre)
2. load action_comedy_thriller file into table
3. create a table called movies_part with 2 columns (movieid, movie_name) that is partitioned on genre **(1)**
4. load each file (action, comedy, and thriller) into a partitions ("Action", "Comedy", and "Thriller") **(1)**

5. describe the structure of the table and list the partitions (hint: describe and show partitions command)

6. navigate to the location of movie_part on HDFS. How does the partitioned table look on HDFS? Write 1 line on what you think is happening when partitioned tables are created. **(0.5)**

7. Run the following queries on both **movies_part** and **movies_whole** table and find out the time it takes to execute the query. **(2.5)**

-- Substitute *table* with actual table name

- (a) Select * from *table* limit 20;
- (b) Select count(*) from *table* where genre='Action';
- (c) Select count(*) from *table*;
- (d) Select t.year, count(*) as count from (Select regexp_extract(movie_name, '([1-2][0-9][0-9][0-9])',1) as year from *table*) t group by year order by count desc limit 5;
- (e) Select t.year, count(*) as count from (Select regexp_extract(movie_name, '([1-2][0-9][0-9][0-9])',1) as year from *table* where genre='Thriller') t group by year order by count desc limit 5;

Answer the following two questions for each of the queries above

(7.1) On which table do you think queries should run faster?

(7.2) On which tables (movie_part or movie_whole) do they actually run faster.

8. With some help from the "select" statement in 7(e) -> create a table called movie_year_temp with following columns (movieid, movie_title, movie_year) **(1)**

Bucketing data in hive

9. create a table called year_buckets with the same column definitions as movie_year_temp, but with 8 buckets, clustered on movie_year **(1)**

10. use insert overwrite table to load the rows in movie_year_temp into year_buckets. **(1)**
(set "hive.enforce.bucketing" to true)

[<http://www.dummies.com/how-to/content/hive-insert-command-examples.html>]

11. Navigate to the location of year_buckets on HDFS. How does the partitioned table look on HDFS?

Apply Histogram function

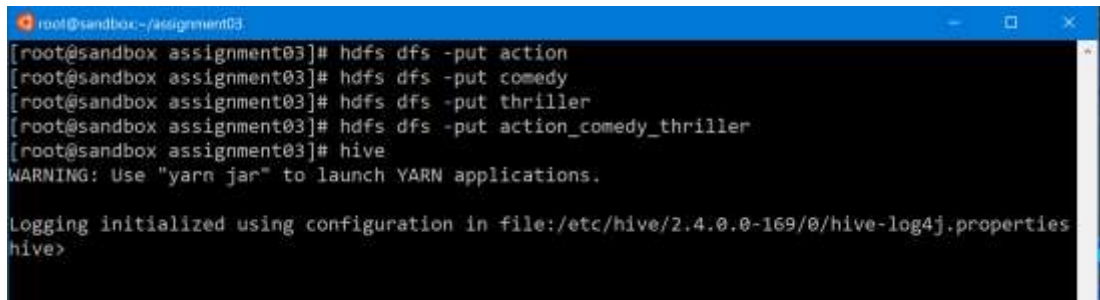
12. Using the table **movie_year_temp** apply the histogram function (with 5 buckets) on **movie_year** to get the distribution of year values in the table **(2)**

Resolution

0. Environment setup

- a. Files copied into HDFS and invoke hive.

```
hdfs dfs -put action
hdfs dfs -put comedy
hdfs dfs -put thriller
hdfs dfs -put action_comedy_thriller
hive
```

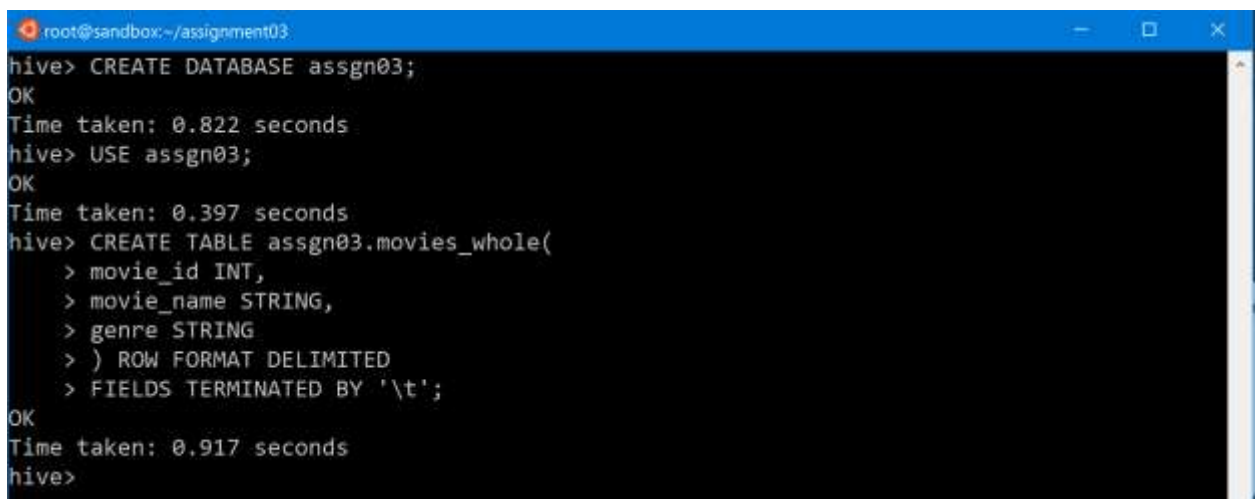


```
root@sandbox:~/assignment03
[root@sandbox assignment03]# hdfs dfs -put action
[root@sandbox assignment03]# hdfs dfs -put comedy
[root@sandbox assignment03]# hdfs dfs -put thriller
[root@sandbox assignment03]# hdfs dfs -put action_comedy_thriller
[root@sandbox assignment03]# hive
WARNING: Use "yarn jar" to launch YARN applications.

Logging initialized using configuration in file:/etc/hive/2.4.0.0-169/0/hive-log4j.properties
hive>
```

1.

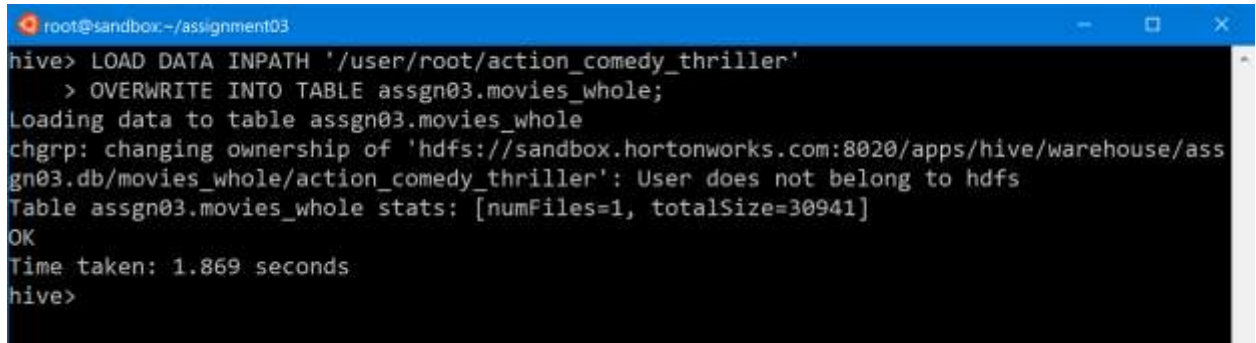
```
CREATE DATABASE assgn03;
USE assgn03;
CREATE TABLE assgn03.movies_whole(
    movie_id INT,
    movie_name STRING,
    genre STRING
) ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';
```



```
root@sandbox:~/assignment03
hive> CREATE DATABASE assgn03;
OK
Time taken: 0.822 seconds
hive> USE assgn03;
OK
Time taken: 0.397 seconds
hive> CREATE TABLE assgn03.movies_whole(
> movie_id INT,
> movie_name STRING,
> genre STRING
> ) ROW FORMAT DELIMITED
> FIELDS TERMINATED BY '\t';
OK
Time taken: 0.917 seconds
hive>
```

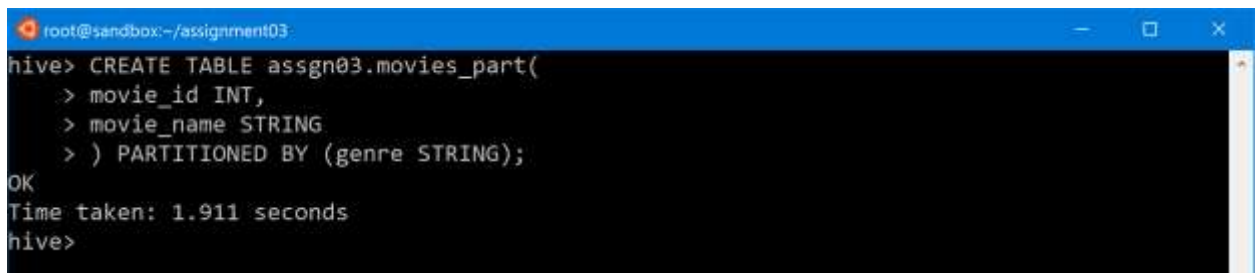
2.

```
LOAD DATA INPATH '/user/root/action_comedy_thriller'
OVERWRITE INTO TABLE assgn03.movies_whole;
```

A terminal window titled 'root@sandbox:~/assignment03' showing the execution of a Hive command. The command is 'LOAD DATA INPATH '/user/root/action_comedy_thriller' OVERWRITE INTO TABLE assgn03.movies_whole;'. The output shows 'Loading data to table assgn03.movies_whole', a warning about changing ownership of the HDFS path, the table statistics '[numFiles=1, totalSize=30941]', and a confirmation 'OK'. The time taken is '1.869 seconds'.

```
root@sandbox:~/assignment03
hive> LOAD DATA INPATH '/user/root/action_comedy_thriller'
> OVERWRITE INTO TABLE assgn03.movies_whole;
Loading data to table assgn03.movies_whole
chgrp: changing ownership of 'hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/ass
gn03.db/movies_whole/action_comedy_thriller': User does not belong to hdfs
Table assgn03.movies_whole stats: [numFiles=1, totalSize=30941]
OK
Time taken: 1.869 seconds
hive>
```

3.
CREATE TABLE assgn03.movies_part(
 movie_id INT,
 movie_name STRING
) PARTITIONED BY (genre STRING);

A terminal window titled 'root@sandbox:~/assignment03' showing the execution of a Hive command. The command is 'CREATE TABLE assgn03.movies_part(movie_id INT, movie_name STRING) PARTITIONED BY (genre STRING);'. The output shows 'OK' and 'Time taken: 1.911 seconds'.

```
root@sandbox:~/assignment03
hive> CREATE TABLE assgn03.movies_part(
> movie_id INT,
> movie_name STRING
> ) PARTITIONED BY (genre STRING);
OK
Time taken: 1.911 seconds
hive>
```

4.
LOAD DATA INPATH '/user/root/action'
INTO TABLE assgn03.movies_part
PARTITION(genre='action');
LOAD DATA INPATH '/user/root/comedy'
INTO TABLE assgn03.movies_part
PARTITION(genre='comedy');
LOAD DATA INPATH '/user/root/thriller'
INTO TABLE assgn03.movies_part
PARTITION(genre='thriller');

```
root@sandbox:~/assignment03 Microsoft Edge
hive> LOAD DATA INPATH '/user/root/action'
> INTO TABLE assign03.movies_part
> PARTITION(genre='action');
Loading data to table assign03.movies_part partition (genre=action)
chgrp: changing ownership of 'hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/assign03.db/movies_part/genre=action/action': User does not belong to hdfs
Partition assign03.movies_part{genre=action} stats: [numFiles=1, numRows=0, totalSize=4702, rawDataSize=0]
OK
Time taken: 5.886 seconds
hive> LOAD DATA INPATH '/user/root/comedy'
> INTO TABLE assign03.movies_part
> PARTITION(genre='comedy');
Loading data to table assign03.movies_part partition (genre=comedy)
chgrp: changing ownership of 'hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/assign03.db/movies_part/genre=comedy/comedy': User does not belong to hdfs
Partition assign03.movies_part{genre=comedy} stats: [numFiles=1, numRows=0, totalSize=13187, rawDataSize=0]
OK
Time taken: 5.161 seconds
hive> LOAD DATA INPATH '/user/root/thriller'
> INTO TABLE assign03.movies_part
> PARTITION(genre='thriller');
Loading data to table assign03.movies_part partition (genre=thriller)
chgrp: changing ownership of 'hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/assign03.db/movies_part/genre=thriller/thriller': User does not belong to hdfs
Partition assign03.movies_part{genre=thriller} stats: [numFiles=1, numRows=0, totalSize=6397, rawDataSize=0]
OK
Time taken: 5.789 seconds
hive>
```

- 5.
- ```
DESCRIBE assign03.movies_part;
SHOW PARTITIONS assign03.movies_part;
```

```
root@sandbox:~/assignment03
hive> DESCRIBE assign03.movies_part;
OK
movie_id int
movie_name string
genre string

Partition Information
col_name data_type comment
genre string
Time taken: 0.635 seconds, Fetched: 8 row(s)
hive> SHOW PARTITIONS assign03.movies_part;
OK
genre=action
genre=comedy
genre=thriller
Time taken: 1.125 seconds, Fetched: 3 row(s)
hive>
```

6.

```
hdfs dfs -ls /apps/hive/warehouse/
hdfs dfs -ls /apps/hive/warehouse/assgn03.db
hdfs dfs -ls /apps/hive/warehouse/assgn03.db/movies_part/
```

```
root@sandbox:~/assignment03
[root@sandbox assignment03]# hdfs dfs -put action_comedy_thriller
[root@sandbox assignment03]# hdfs dfs -ls
Found 23 items
drwx----- - root root 0 2016-10-04 06:17 .Trash
drwxr-xr-x - root root 0 2016-10-03 05:06 .hiveJars
[root@sandbox assignment03]# hdfs dfs -ls /apps/hive/warehouse
Found 10 items
drwxrwxrwx - root hdfs 0 2016-10-04 06:25 /apps/hive/warehouse/assgn03.db
drwxrwxrwx - root hdfs 0 2016-05-12 00:27 /apps/hive/warehouse/full_text
drwxrwxrwx - root hdfs 0 2016-10-03 08:41 /apps/hive/warehouse/midterm.db
drwxrwxrwx - hive hdfs 0 2016-03-14 14:31 /apps/hive/warehouse/sample_07
drwxrwxrwx - hive hdfs 0 2016-03-14 14:31 /apps/hive/warehouse/sample_08
drwxrwxrwx - root hdfs 0 2016-06-04 11:49 /apps/hive/warehouse/titanic.db
drwxrwxrwx - root hdfs 0 2016-05-13 00:58 /apps/hive/warehouse/twitter.db
drwxrwxrwx - root hdfs 0 2016-10-03 05:07 /apps/hive/warehouse/twitter_new.db
drwxrwxrwx - root hdfs 0 2016-05-12 00:29 /apps/hive/warehouse/wordcount
drwxrwxrwx - hive hdfs 0 2016-03-14 14:52 /apps/hive/warehouse/xademo.db
[root@sandbox assignment03]# hdfs dfs -ls /apps/hive/warehouse/assgn03.db
Found 2 items
drwxrwxrwx - root hdfs 0 2016-10-04 06:31 /apps/hive/warehouse/assgn03.db/movies_part
drwxrwxrwx - root hdfs 0 2016-10-04 06:22 /apps/hive/warehouse/assgn03.db/movies_whole
[root@sandbox assignment03]# hdfs dfs -ls /apps/hive/warehouse/assgn03.db/movies_part
Found 3 items
drwxrwxrwx - root hdfs 0 2016-10-04 06:31 /apps/hive/warehouse/assgn03.db/movies_part/genre=action
drwxrwxrwx - root hdfs 0 2016-10-04 06:31 /apps/hive/warehouse/assgn03.db/movies_part/genre=comedy
drwxrwxrwx - root hdfs 0 2016-10-04 06:31 /apps/hive/warehouse/assgn03.db/movies_part/genre=thriller
[root@sandbox assignment03]#
```

Instead of keeping the default structure of one data file inside the table directory, the partitioning created three directories for the partitioned table (movies\_part) with the following structure: one directory per partition value, named [FIELD]=[PARTITION\_VALUE].

## 7. Query Times

| Query | movies_part time (s) | movies_whole time (s) | Diff (s) | Diff (%) |
|-------|----------------------|-----------------------|----------|----------|
| a     | 5.469                | 1.228                 | 4.241    | 345.3    |
| b     | 14.486               | 7.157                 | 7.329    | 102.4    |
| c     | 9.558                | 6.535                 | 3.023    | 46.2     |
| d     | 11.101               | 6.737                 | 4.364    | 64.7     |
| e     | 6.037                | 6.619                 | -0.582   | -8%      |

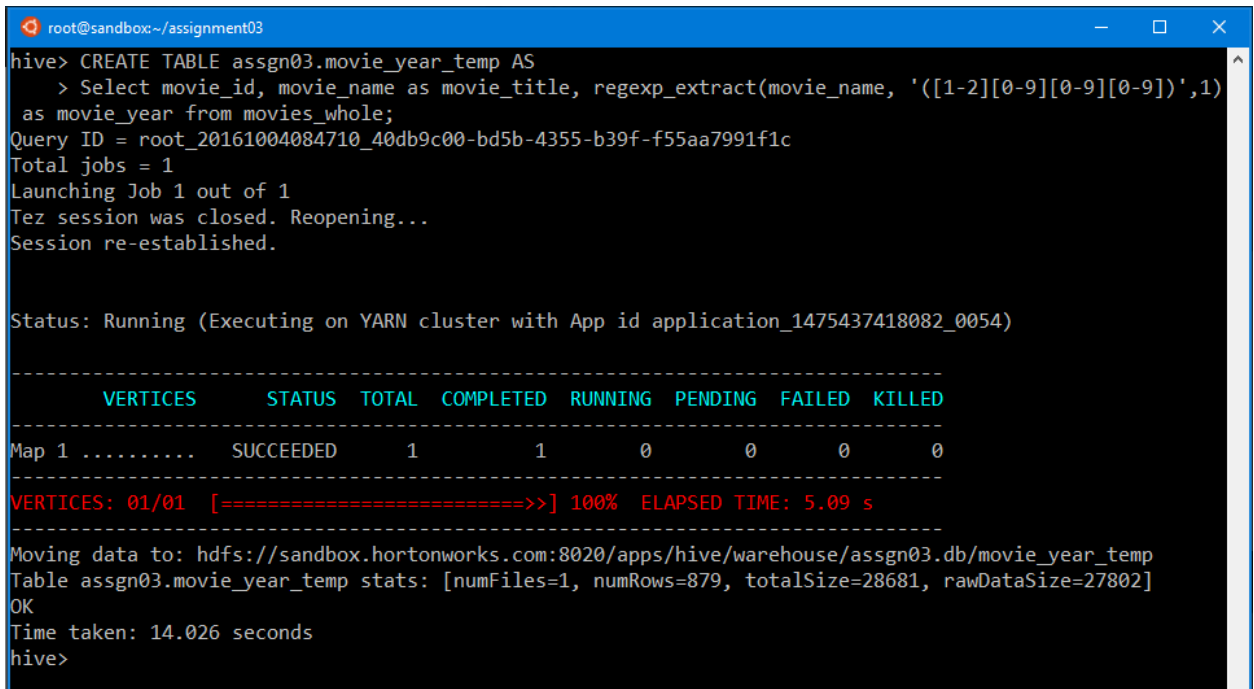
- Select \* from \*table\* limit 20;
- Select count(\*) from \*table\* where genre='Action';
- Select count(\*) from \*table\*;
- Select t.year, count(\*) as count from (Select regexp\_extract(movie\_name, '([1-2][0-9][0-9][0-9])',1) as year from \*table\*) t group by year order by count desc limit 5;
- Select t.year, count(\*) as count from (Select regexp\_extract(movie\_name, '([1-2][0-9][0-9][0-9])',1) as year from \*table\* where genre='Thriller') t group by year order by count desc limit 5;

**7.1-** Intuitively queries should run faster on the partitioned version of the table **movies\_part**.

**7.2-** Four out of all the five queries ran faster in the non-partitioned **movies\_whole** table. Run times difference vary from 345% faster in the best case to 8% slower in the worst case.

8.

```
CREATE TABLE assgn03.movie_year_temp AS
Select movie_id, movie_name as movie_title,
regexp_extract(movie_name, '([1-2][0-9][0-9][0-9])',1) as
movie_year from movies_whole;
```

A terminal window titled 'root@sandbox:~/assignment03' showing the execution of a Hive query. The query creates a table 'assgn03.movie\_year\_temp' by selecting from 'movies\_whole' and extracting the year from the movie name. The terminal output shows the query ID, total jobs, and job status. A progress bar indicates 100% completion with an elapsed time of 5.09 seconds. The final status is 'OK' and the time taken is 14.026 seconds.

```
root@sandbox:~/assignment03
hive> CREATE TABLE assgn03.movie_year_temp AS
> Select movie_id, movie_name as movie_title, regexp_extract(movie_name, '([1-2][0-9][0-9][0-9])',1)
as movie_year from movies_whole;
Query ID = root_20161004084710_40db9c00-bd5b-4355-b39f-f55aa7991f1c
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_1475437418082_0054)

VERTICES STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

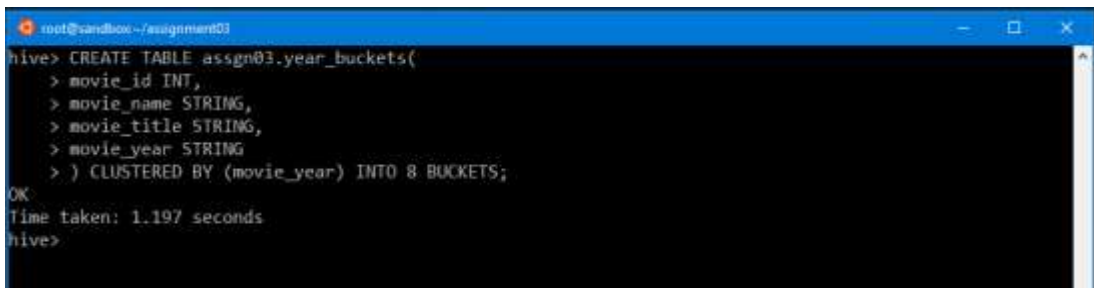
Map 1 SUCCEEDED 1 1 0 0 0 0

VERTICES: 01/01 [=====>>] 100% ELAPSED TIME: 5.09 s

Moving data to: hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/assgn03.db/movie_year_temp
Table assgn03.movie_year_temp stats: [numFiles=1, numRows=879, totalSize=28681, rawDataSize=27802]
OK
Time taken: 14.026 seconds
hive>
```

9.

```
CREATE TABLE assgn03.year_buckets(
 movie_id INT,
 movie_name STRING,
 movie_title STRING,
 movie_year STRING
) CLUSTERED BY (movie_year) INTO 8 BUCKETS;
```

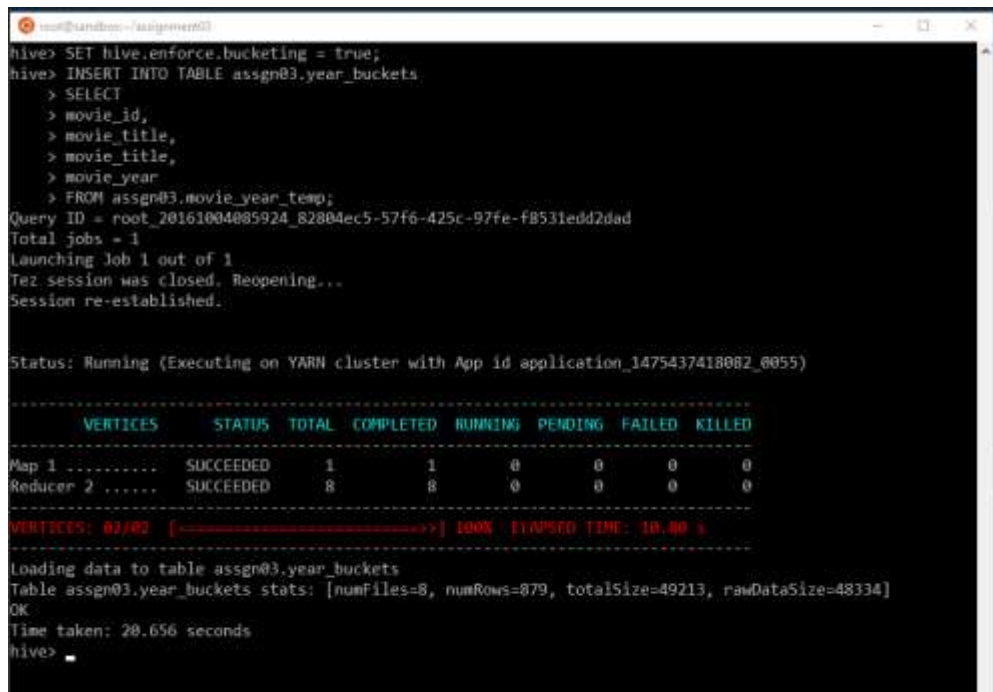
A terminal window titled 'root@sandbox:~/assignment03' showing the execution of a Hive query. The query creates a table 'assgn03.year\_buckets' with columns for movie\_id, movie\_name, movie\_title, and movie\_year, clustered by movie\_year into 8 buckets. The terminal output shows the query execution and the time taken, which is 1.197 seconds.

```
root@sandbox:~/assignment03
hive> CREATE TABLE assgn03.year_buckets(
> movie_id INT,
> movie_name STRING,
> movie_title STRING,
> movie_year STRING
>) CLUSTERED BY (movie_year) INTO 8 BUCKETS;
OK
Time taken: 1.197 seconds
hive>
```



10.

```
SET hive.enforce.bucketing = true;
INSERT INTO TABLE assign03.year_buckets
SELECT
 movie_id,
 movie_title,
 movie_title,
 movie_year
FROM assign03.movie_year_temp;
```



The screenshot shows a Hive CLI session with the following output:

```
hive> SET hive.enforce.bucketing = true;
hive> INSERT INTO TABLE assign03.year_buckets
> SELECT
> movie_id,
> movie_title,
> movie_title,
> movie_year
> FROM assign03.movie_year_temp;
Query ID = root_20161004085924_82804ec5-57f6-425c-97fe-f8531edd2dad
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_1475437418082_0055)

VERTICES STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

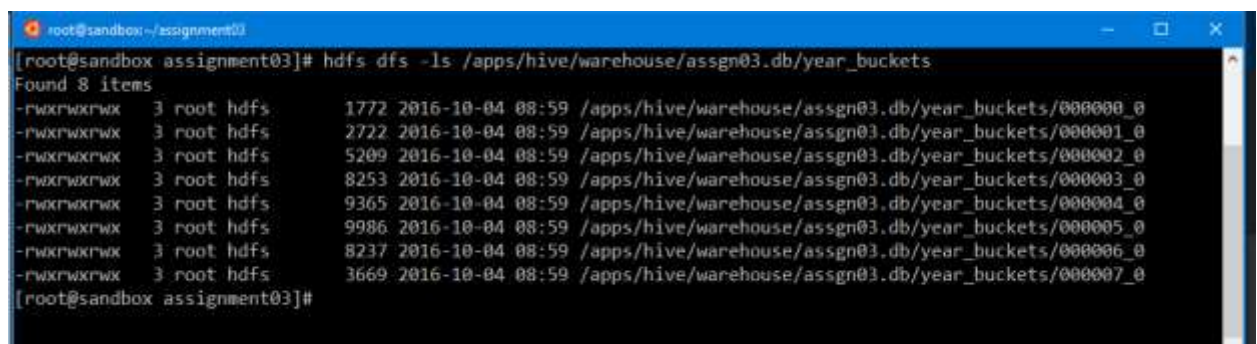
Map 1 SUCCEEDED 1 1 0 0 0 0
Reducer 2 SUCCEEDED 8 8 0 0 0 0

VERTICES: 02/02 [=====] 100% [ELAPSED TIME: 10.00 s]

Loading data to table assign03.year_buckets
Table assign03.year_buckets stats: [numFiles=8, numRows=879, totalSize=49213, rawDataSize=48334]
OK
Time taken: 20.656 seconds
hive>
```

11.

```
hdfs dfs -ls /apps/hive/warehouse/assign03.db/year_buckets
```



The screenshot shows a Hadoop CLI session with the following output:

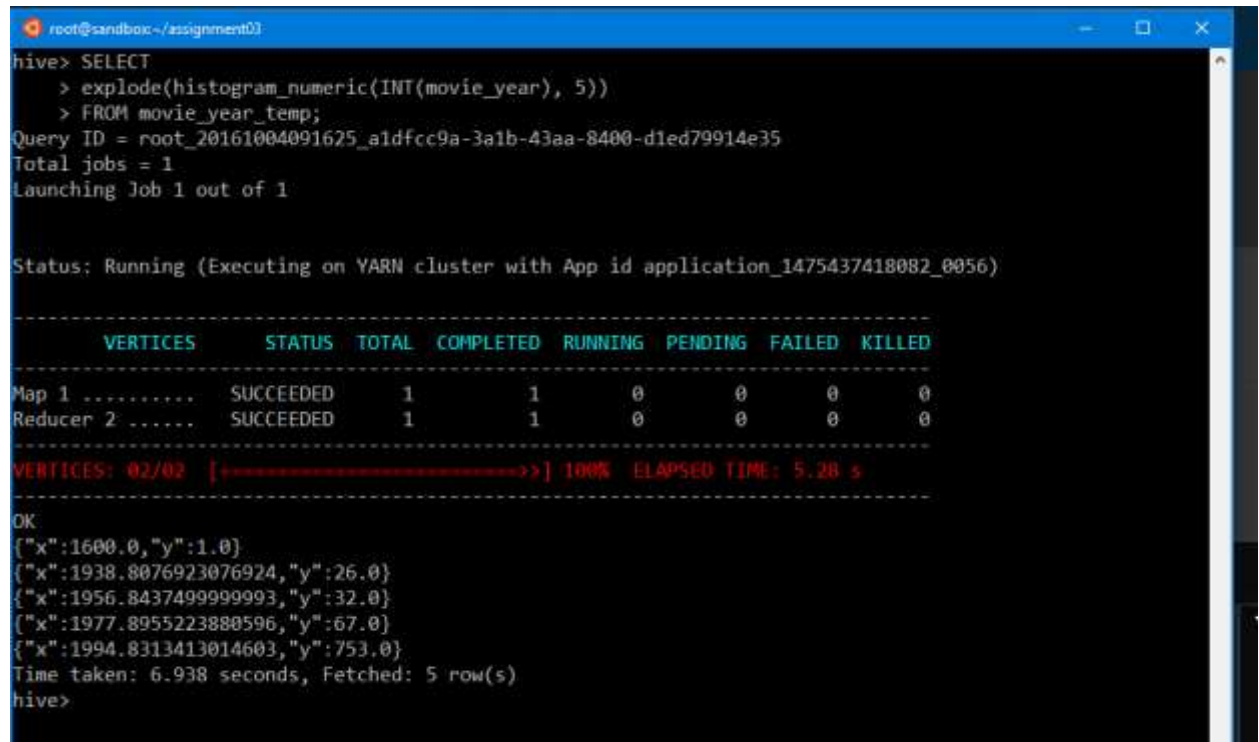
```
[root@sandbox assignment03]# hdfs dfs -ls /apps/hive/warehouse/assign03.db/year_buckets
Found 8 items
-rwxrwxrwx 3 root hdfs 1772 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000000_0
-rwxrwxrwx 3 root hdfs 2722 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000001_0
-rwxrwxrwx 3 root hdfs 5209 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000002_0
-rwxrwxrwx 3 root hdfs 8253 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000003_0
-rwxrwxrwx 3 root hdfs 9365 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000004_0
-rwxrwxrwx 3 root hdfs 9986 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000005_0
-rwxrwxrwx 3 root hdfs 8237 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000006_0
-rwxrwxrwx 3 root hdfs 3669 2016-10-04 08:59 /apps/hive/warehouse/assign03.db/year_buckets/000007_0
[root@sandbox assignment03]#
```



The partitioned table looks like a series of sequentially numbered files inside the table directory.

12.

```
SELECT
 explode(histogram_numeric(INT(movie_year), 5))
FROM movie_year_temp;
```



```
root@sandbox:~/assignment03
hive> SELECT
> explode(histogram_numeric(INT(movie_year), 5))
> FROM movie_year_temp;
Query ID = root_20161004091625_a1dfcc9a-3a1b-43aa-8400-d1ed79914e35
Total jobs = 1
Launching Job 1 out of 1

Status: Running (Executing on YARN cluster with App id application_1475437418082_0056)

VERTICES STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

Map 1 SUCCEEDED 1 1 0 0 0 0
Reducer 2 SUCCEEDED 1 1 0 0 0 0

VERTICES: 02/02 [=====] 100% ELAPSED TIME: 5.28 s

OK
{"x":1600.0,"y":1.0}
{"x":1938.8076923076924,"y":26.0}
{"x":1956.8437499999993,"y":32.0}
{"x":1977.8955223880596,"y":67.0}
{"x":1994.8313413014603,"y":753.0}
Time taken: 6.938 seconds, Fetched: 5 row(s)
hive>
```