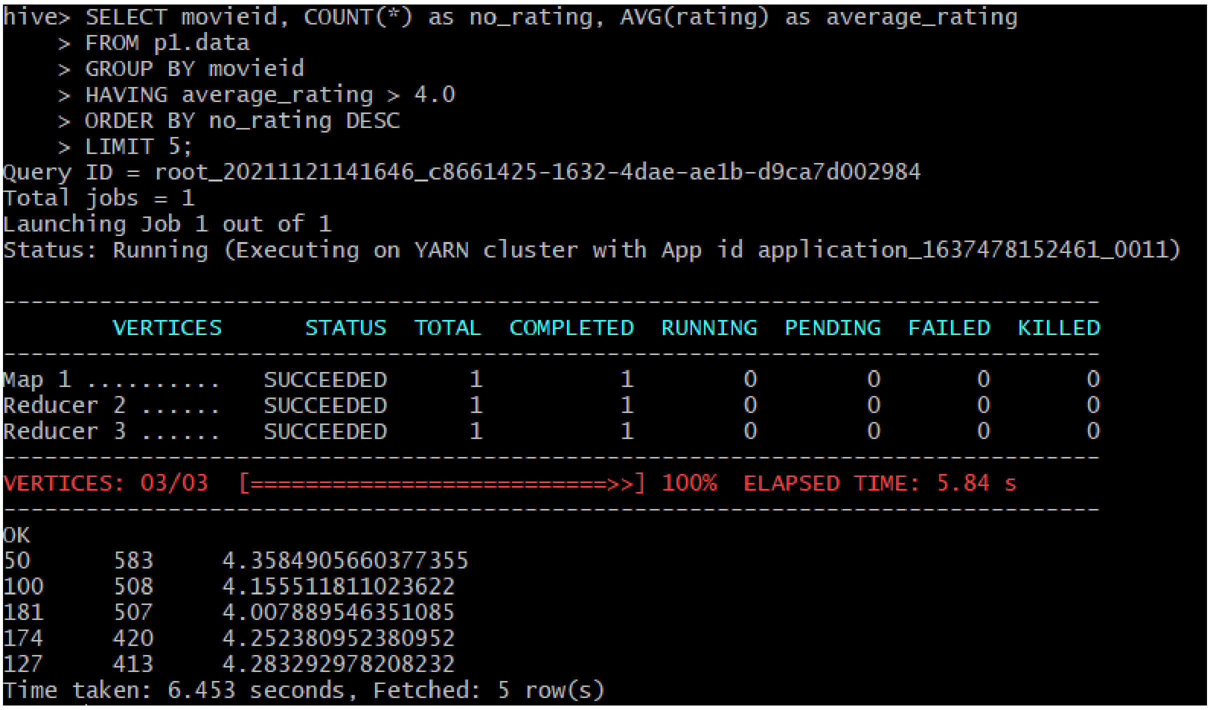
Part I - Hive

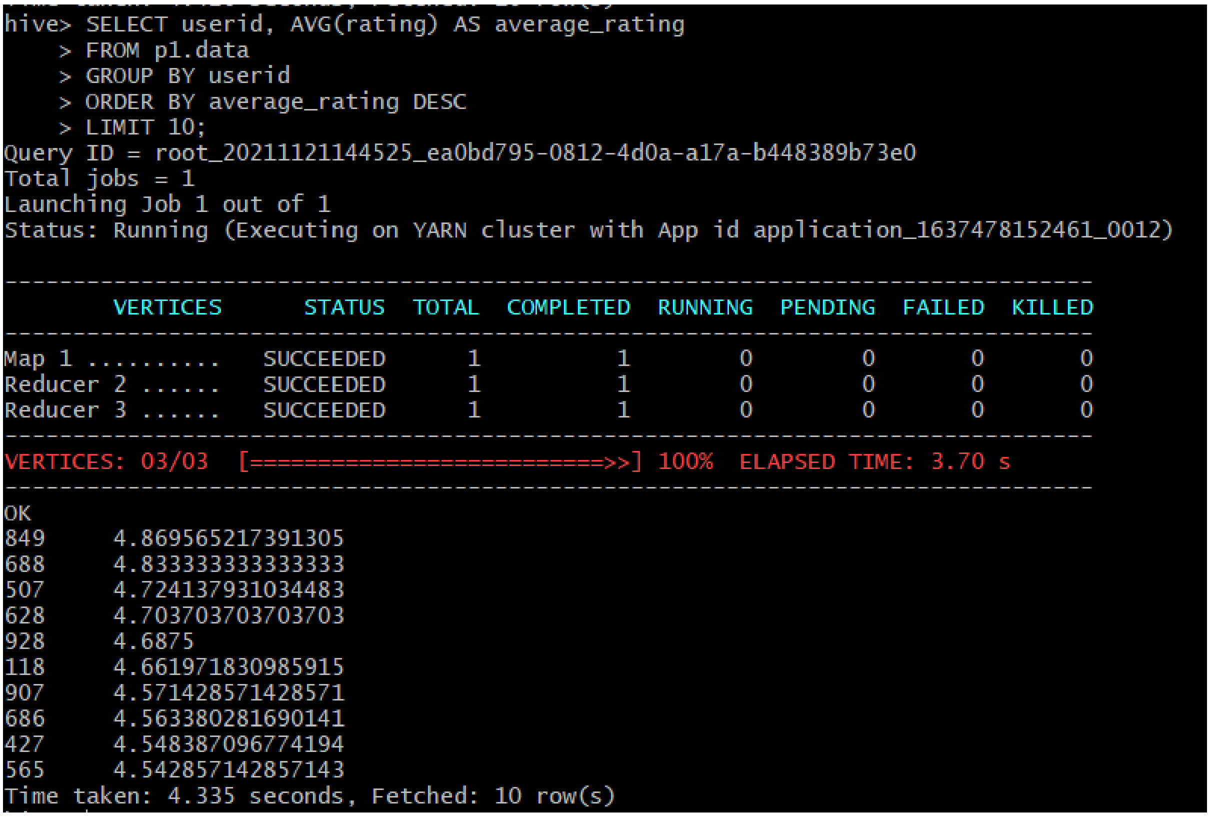
```
hive> create database p1;
hive> use p1;
hive> create table p1.data (userId INT, movieId INT, rating FLOAT, time BIGINT)
      > ROW FORMAT DELIMITED
      > FIELDS TERMINATED BY '₩t';
hive> load data inpath '/user/root/A3/u.data'
      > overwrite into table p1.data;
hive> create table userdata (userid INT, age INT, gender STRING, occup STRING, zip STRING)
      > ROW FORMAT DELIMITED FIELDS TERMINATED BY '|';
hive> load data inpath '/user/root/A3/u.user'
      > overwrite into table p1.userdata;
```

Question 1: Return 5 movies with the highest number of ratings that also had an average rating of above 4 stars (10pts)



```
hive> SELECT movieid, COUNT(*) as no_rating, AVG(rating) as average_rating
    > FROM p1.data
    > GROUP BY movieid
    > HAVING average_rating > 4.0
    > ORDER BY no_rating DESC
    > LIMIT 5;
Query ID = root_20211121141646_c8661425-1632-4dae-ae1b-d9ca7d002984
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0011)

----------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.84 s
----------------------------------------------------------------------------------
OK
50      583     4.3584905660377355
100     508     4.155511811023622
181     507     4.007889546351085
174     420     4.252380952380952
127     413     4.283292978208232
Time taken: 6.453 seconds, Fetched: 5 row(s)
```

Question 2: Find which user has given the highest average rating? (Hint: You need to check all ratings given by the user for all movies) (10pts)

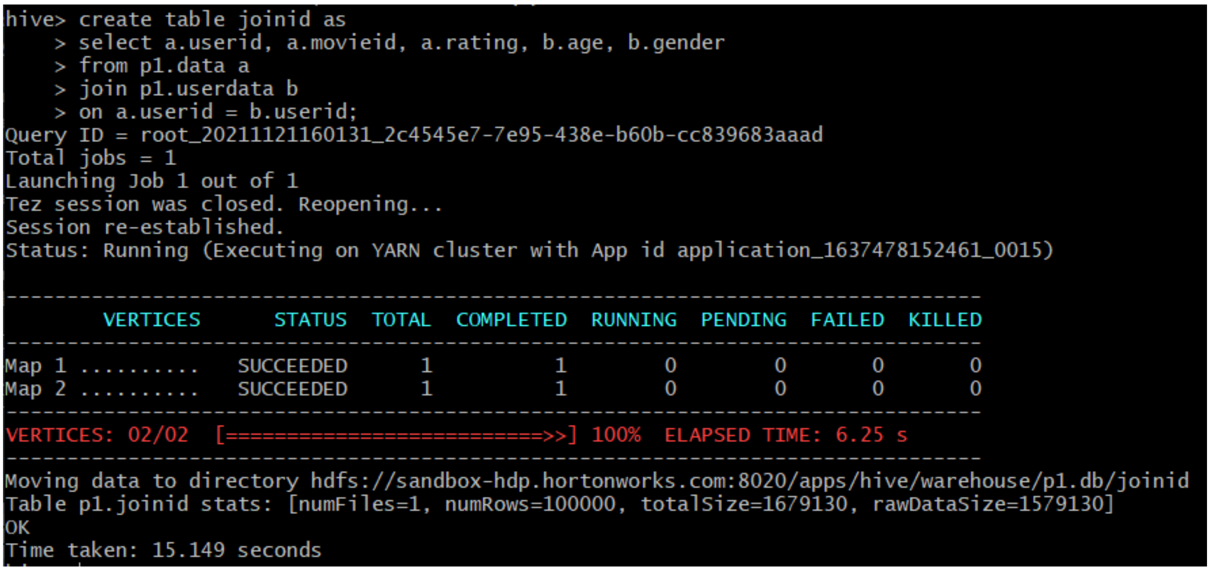


```
hive> SELECT userid, AVG(rating) AS average_rating
    > FROM p1.data
    > GROUP BY userid
    > ORDER BY average_rating DESC
    > LIMIT 10;
Query ID = root_20211121144525_ea0bd795-0812-4d0a-a17a-b448389b73e0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0012)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 3.70 s
----------------------------------------------------------------------------------------------
OK
849     4.869565217391305
688     4.833333333333333
507     4.724137931034483
628     4.703703703703703
928     4.6875
118     4.661971830985915
907     4.571428571428571
686     4.563380281690141
427     4.548387096774194
565     4.542857142857143
Time taken: 4.335 seconds, Fetched: 10 row(s)
```

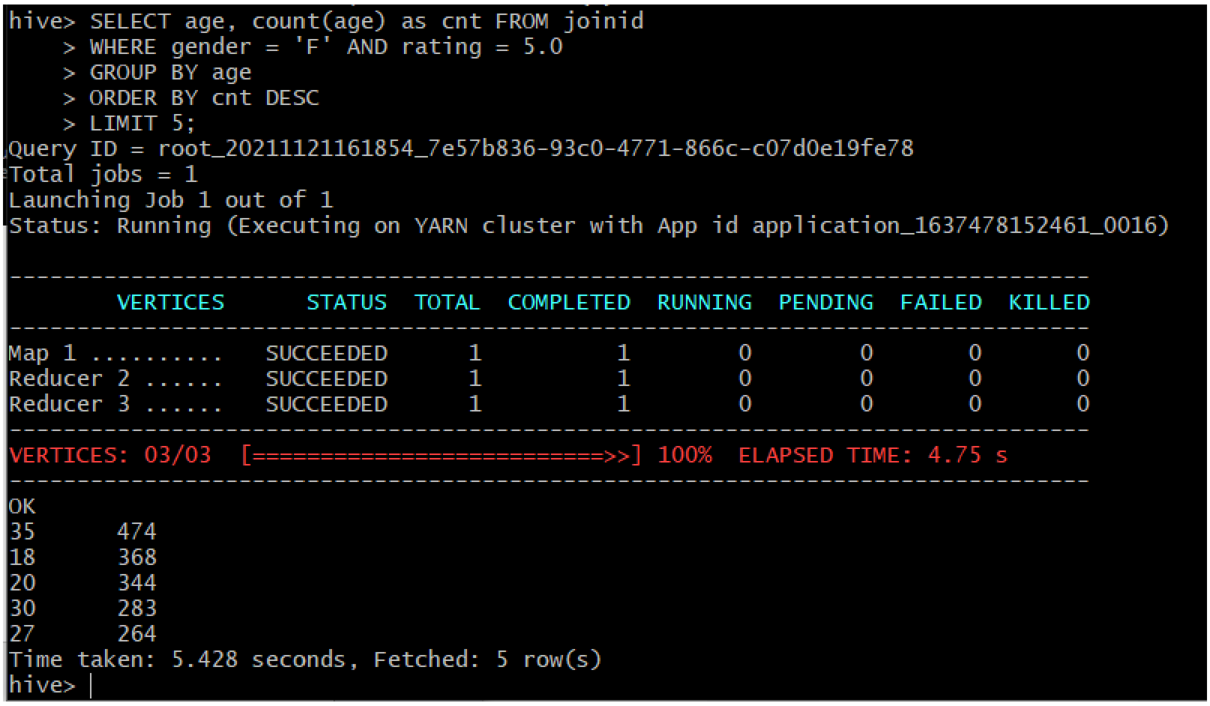
Question 3: Show the top 5 age groups of female user who gave the 5-rating to the movies? (10pts)

Joinid table created on data.userid = userdata.userid.

Then filtered with condition (Female who gave 5 rating to movies)

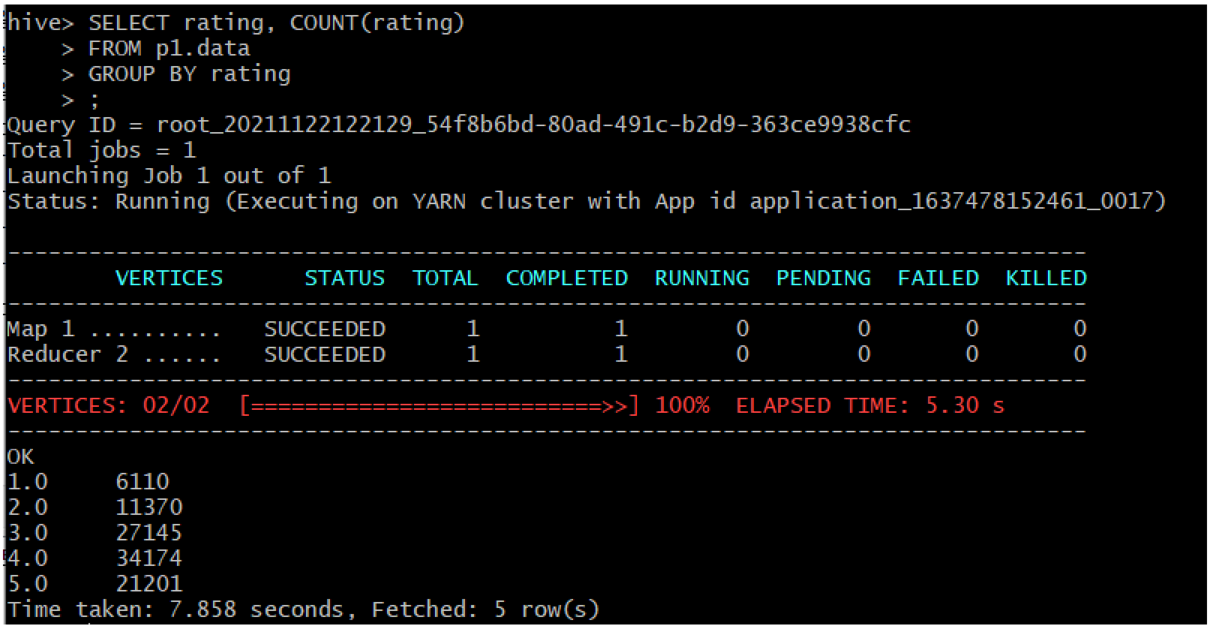


```
hive> create table joinid as
    > select a.userid, a.movieid, a.rating, b.age, b.gender
    > from p1.data a
    > join p1.userdata b
    > on a.userid = b.userid;
Query ID = root_20211121160131_2c4545e7-7e95-438e-b60b-cc839683aaad
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1637478152461_0015)

----------------------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 2 ..........   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.25 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/p1.db/joinid
Table p1.joinid stats: [numFiles=1, numRows=100000, totalSize=1679130, rawDataSize=1579130]
OK
Time taken: 15.149 seconds
```



```
hive> SELECT age, count(age) as cnt FROM joinid
    > WHERE gender = 'F' AND rating = 5.0
    > GROUP BY age
    > ORDER BY cnt DESC
    > LIMIT 5;
Query ID = root_20211121161854_7e57b836-93c0-4771-866c-c07d0e19fe78
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0016)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 4.75 s
----------------------------------------------------------------------------------------------
OK
35      474
18      368
20      344
30      283
27      264
Time taken: 5.428 seconds, Fetched: 5 row(s)
hive> |
```

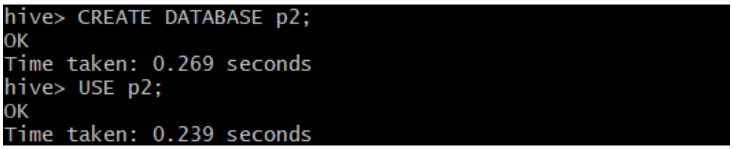
Question 4: How many times each star rating was given to a movie? (10pts)



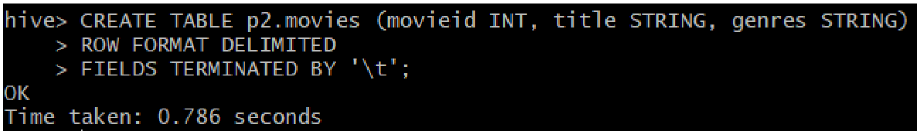
```
hive> SELECT rating, COUNT(rating)
    > FROM p1.data
    > GROUP BY rating
    > ;
Query ID = root_20211122122129_54f8b6bd-80ad-491c-b2d9-363ce9938cfc
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0017)

----------------------------------------------------------------------------------------------
        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: 5.30 s
----------------------------------------------------------------------------------------------
OK
1.0     6110
2.0     11370
3.0     27145
4.0     34174
5.0     21201
Time taken: 7.858 seconds, Fetched: 5 row(s)
```

Part II - Advance Hive

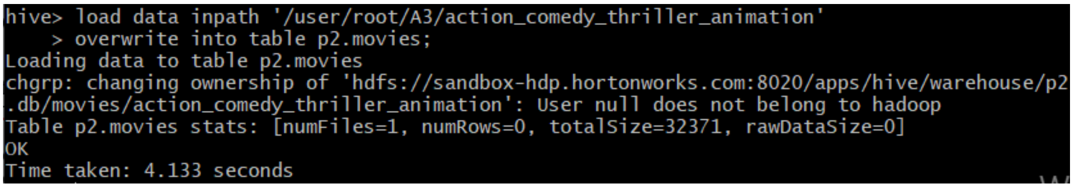
Store complete information of all movies into a hive table (2 pts).



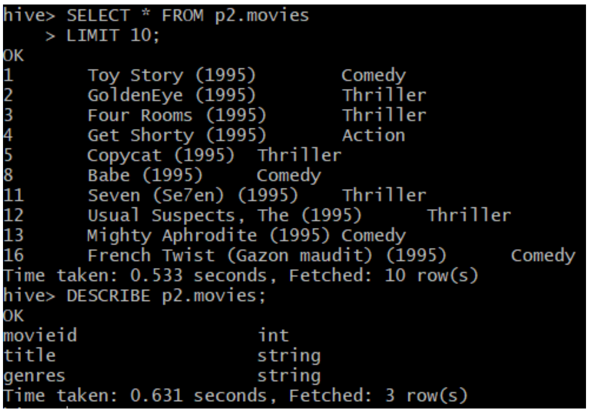
```
hive> CREATE DATABASE p2;
OK
Time taken: 0.269 seconds
hive> USE p2;
OK
Time taken: 0.239 seconds
```



```
hive> CREATE TABLE p2.movies (movieid INT, title STRING, genres STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t';
OK
Time taken: 0.786 seconds
```

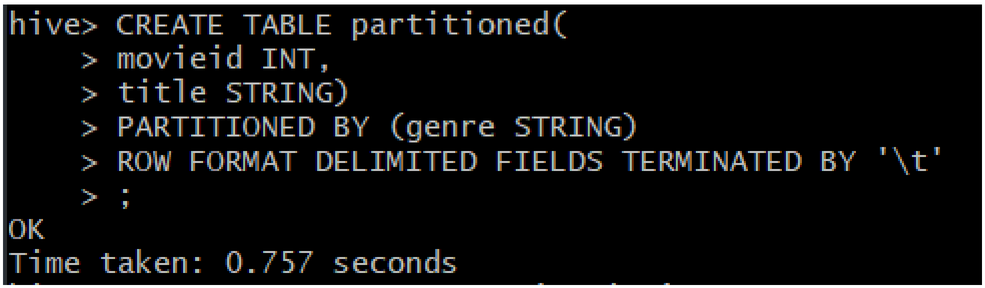


```
hive> load data inpath '/user/root/A3/action_comedy_thriller_animation'
    > overwrite into table p2.movies;
Loading data to table p2.movies
chgrp: changing ownership of 'hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/p2
.db/movies/action_comedy_thriller_animation': User null does not belong to hadoop
Table p2.movies stats: [numFiles=1, numRows=0, totalSize=32371, rawDataSize=0]
OK
Time taken: 4.133 seconds
```

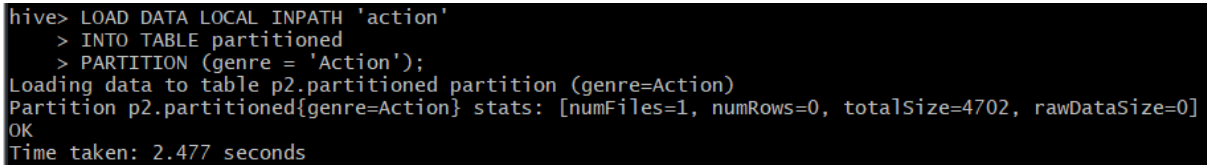


```
hive> SELECT * FROM p2.movies
    > LIMIT 10;
OK
1       Toy Story (1995)        Comedy
2       GoldenEye (1995)        Thriller
3       Four Rooms (1995)       Thriller
4       Get Shorty (1995)       Action
5       Copycat (1995)  Thriller
8       Babe (1995)     Comedy
11      Seven (Se7en) (1995)    Thriller
12      Usual Suspects, The (1995)      Thriller
13      Mighty Aphrodite (1995) Comedy
16      French Twist (Gazon maudit) (1995)      Comedy
Time taken: 0.533 seconds, Fetched: 10 row(s)
hive> DESCRIBE p2.movies;
OK
movieid                 int
title                   string
genres                  string
Time taken: 0.631 seconds, Fetched: 3 row(s)
```

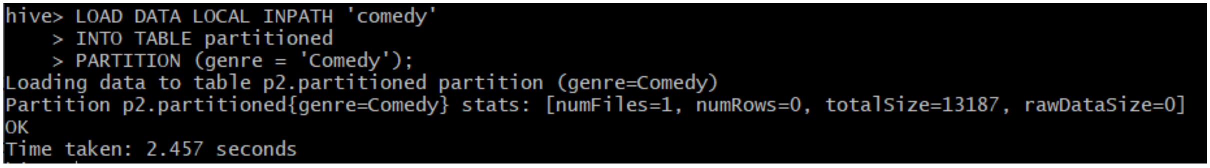
Store data into a hive table that is partitioned on genre (2 pts).



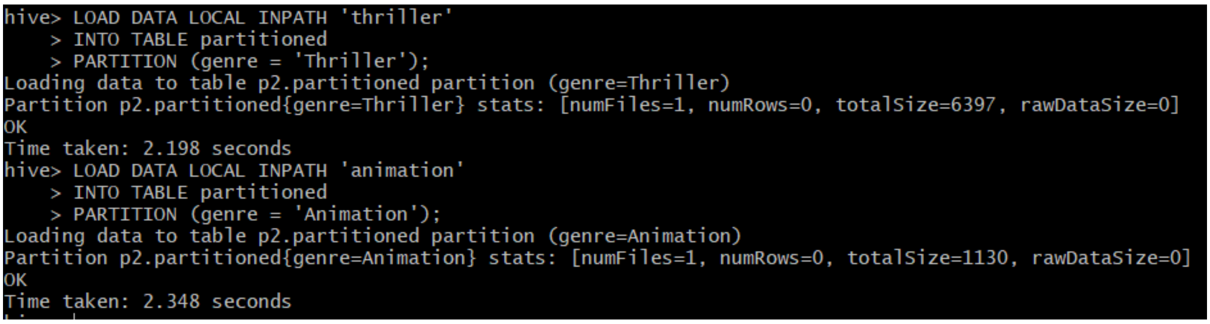
```
hive> CREATE TABLE partitioned(
    > movieid INT,
    > title STRING)
    > PARTITIONED BY (genre STRING)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    > ;
OK
Time taken: 0.757 seconds
```



```
hive> LOAD DATA LOCAL INPATH 'action'
    > INTO TABLE partitioned
    > PARTITION (genre = 'Action');
Loading data to table p2.partitioned partition (genre=Action)
Partition p2.partitioned{genre=Action} stats: [numFiles=1, numRows=0, totalSize=4702, rawDataSize=0]
OK
Time taken: 2.477 seconds
```

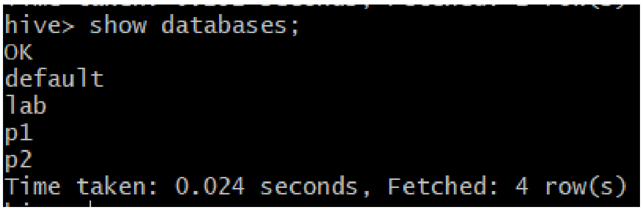


```
hive> LOAD DATA LOCAL INPATH 'comedy'
    > INTO TABLE partitioned
    > PARTITION (genre = 'Comedy');
Loading data to table p2.partitioned partition (genre=Comedy)
Partition p2.partitioned{genre=Comedy} stats: [numFiles=1, numRows=0, totalSize=13187, rawDataSize=0]
OK
Time taken: 2.457 seconds
```

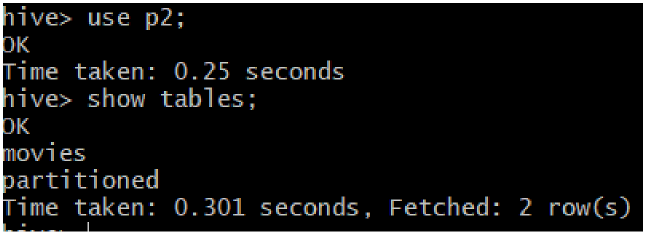


```
hive> LOAD DATA LOCAL INPATH 'thriller'
    > INTO TABLE partitioned
    > PARTITION (genre = 'Thriller');
Loading data to table p2.partitioned partition (genre=Thriller)
Partition p2.partitioned{genre=Thriller} stats: [numFiles=1, numRows=0, totalSize=6397, rawDataSize=0]
OK
Time taken: 2.198 seconds
hive> LOAD DATA LOCAL INPATH 'animation'
    > INTO TABLE partitioned
    > PARTITION (genre = 'Animation');
Loading data to table p2.partitioned partition (genre=Animation)
Partition p2.partitioned{genre=Animation} stats: [numFiles=1, numRows=0, totalSize=1130, rawDataSize=0]
OK
Time taken: 2.348 seconds
```

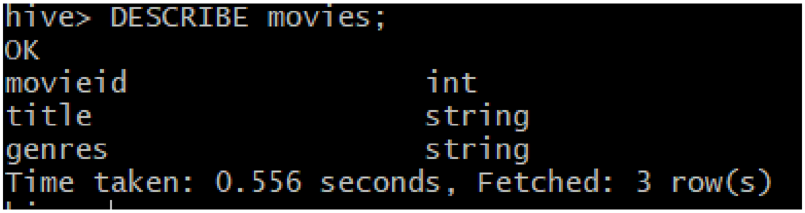
Show database and table structures (1 pts)



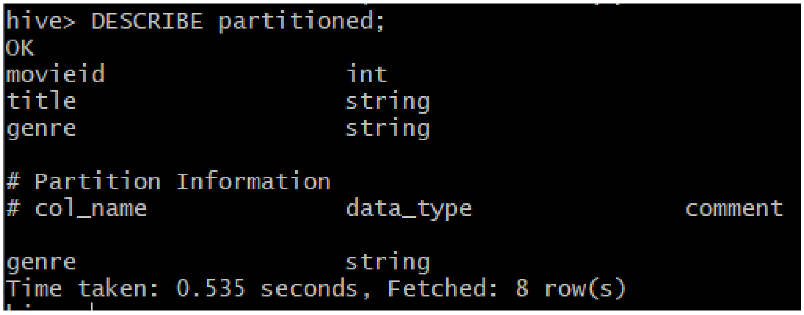
```
hive> show databases;
OK
default
lab
p1
p2
Time taken: 0.024 seconds, Fetched: 4 row(s)
```



```
hive> use p2;
OK
Time taken: 0.25 seconds
hive> show tables;
OK
movies
partitioned
Time taken: 0.301 seconds, Fetched: 2 row(s)
```



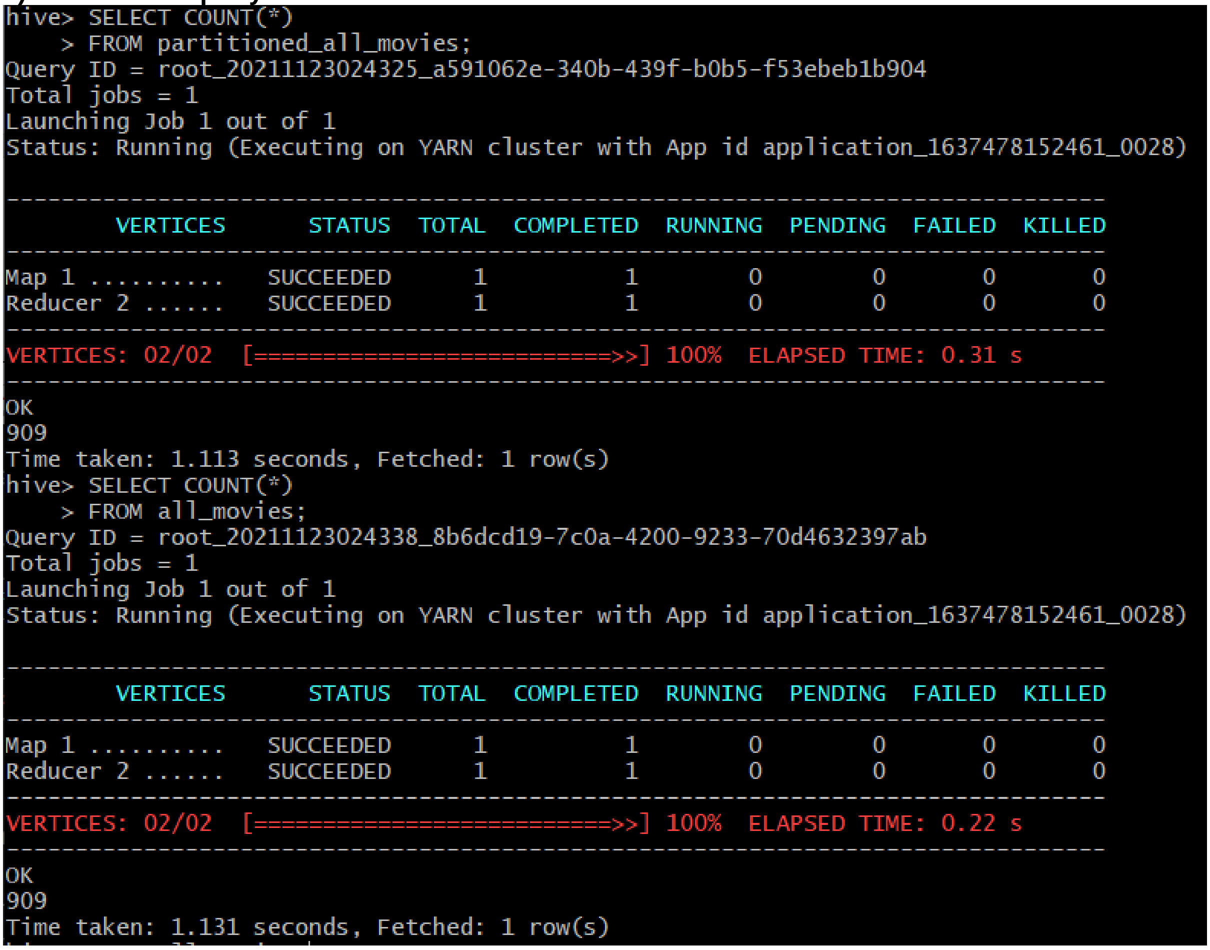
```
hive> DESCRIBE movies;
OK
movieid                 int
title                   string
genres                  string
Time taken: 0.556 seconds, Fetched: 3 row(s)
```



```
hive> DESCRIBE partitioned;
OK
movieid                 int
title                   string
genre                   string

# Partition Information
# col_name              data_type               comment

genre                   string
Time taken: 0.535 seconds, Fetched: 8 row(s)
```

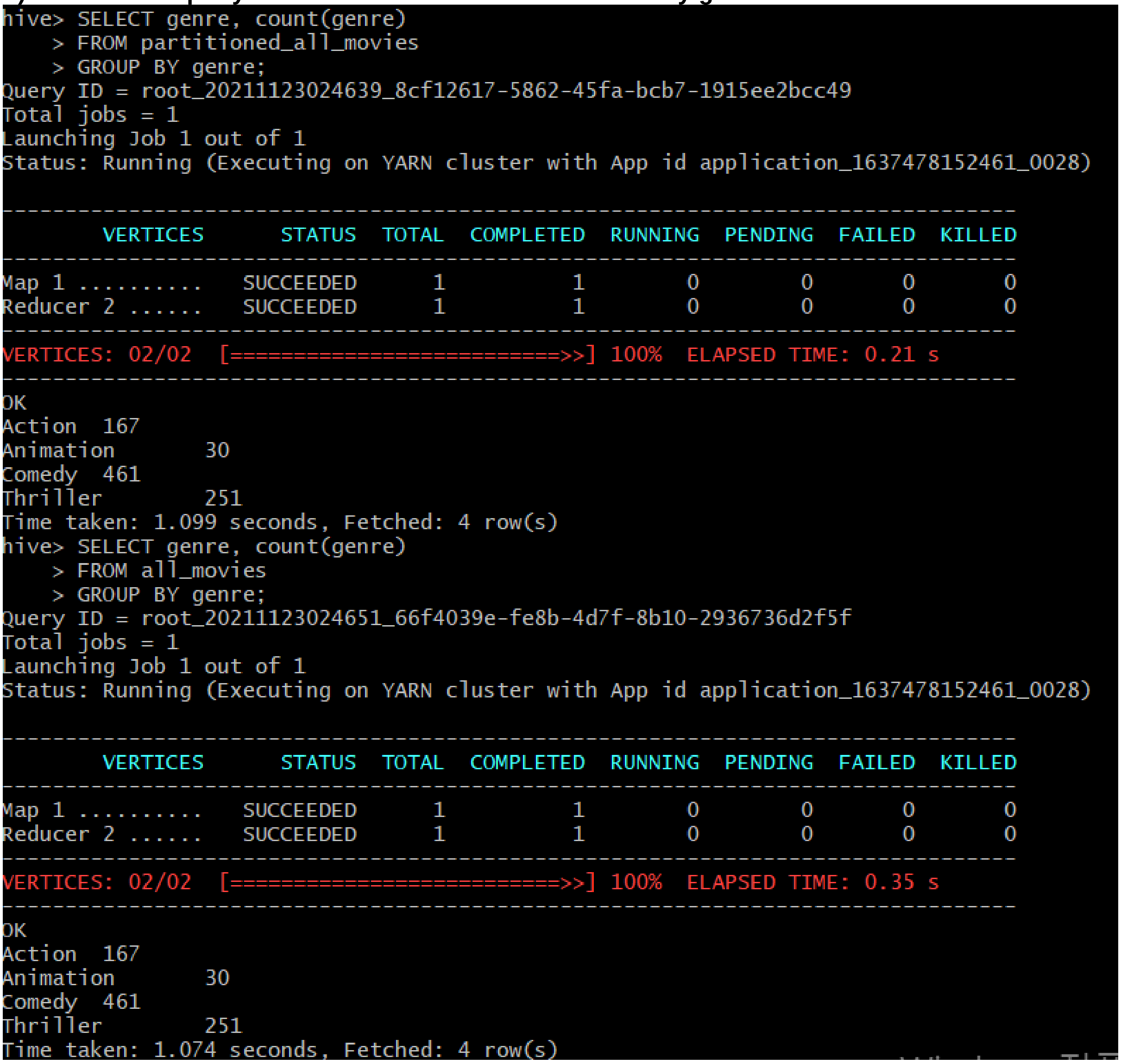
Question 5: Write the following queries, report results and execution time on both partitioned and complete data: (20 pts) -- Create a table "all_movies" and load all the movies information in it. To answer the below questions, substitute *table* with actual table name "all_movies".

a) Write a hive query to find the total number of records from the *table*



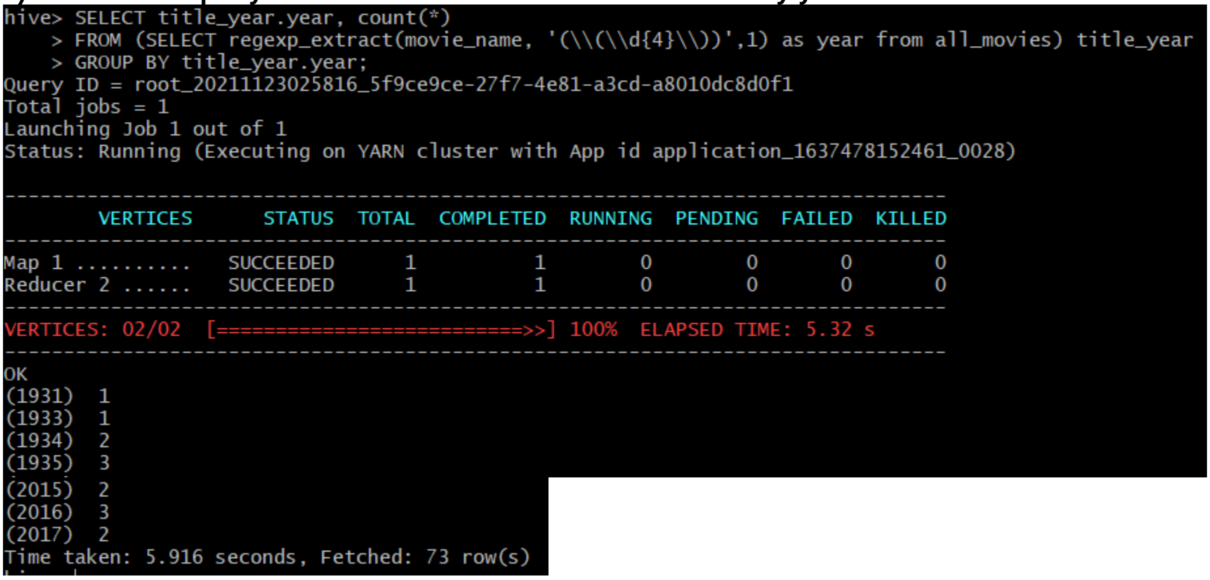
```
hive> SELECT COUNT(*)
    > FROM partitioned_all_movies;
Query ID = root_20211123024325_a591062e-340b-439f-b0b5-f53ebeb1b904
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------------------
        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: 0.31 s
----------------------------------------------------------------------------------------------
OK
909
Time taken: 1.113 seconds, Fetched: 1 row(s)
hive> SELECT COUNT(*)
    > FROM all_movies;
Query ID = root_20211123024338_8b6dcd19-7c0a-4200-9233-70d4632397ab
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------------------
        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: 0.22 s
----------------------------------------------------------------------------------------------
OK
909
Time taken: 1.131 seconds, Fetched: 1 row(s)
```

b) Write a hive query to find the total number of records by genre from *table*

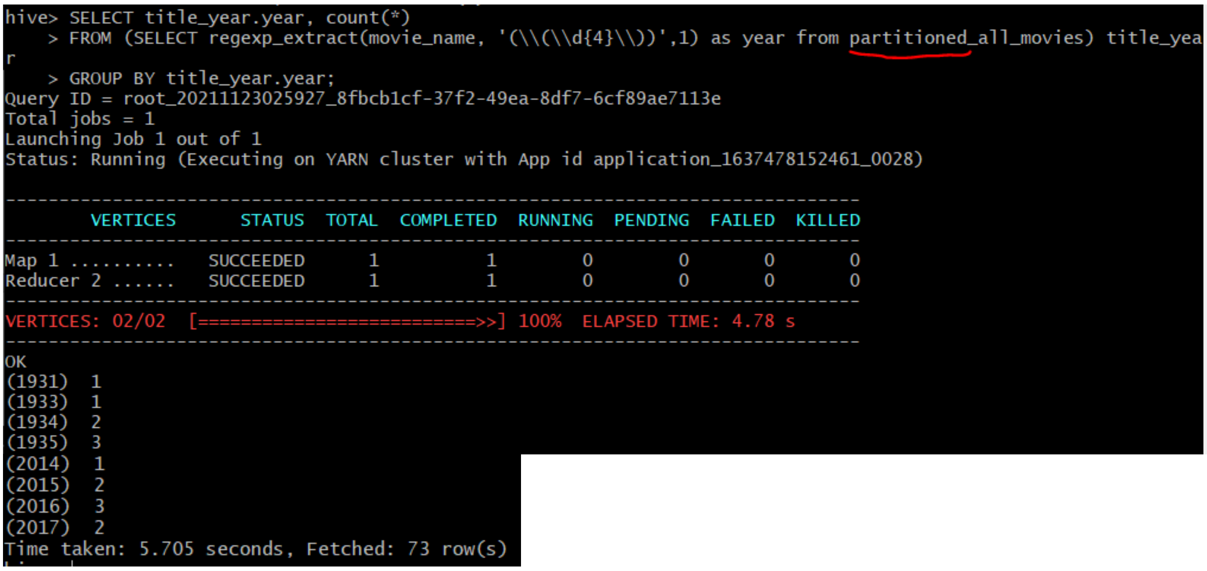


```
hive> SELECT genre, count(genre)
    > FROM partitioned_all_movies
    > GROUP BY genre;
Query ID = root_20211123024639_8cf12617-5862-45fa-bcb7-1915ee2bcc49
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------
        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: 0.21 s
----------------------------------------------------------------------------------
OK
Action  167
Animation       30
Comedy  461
Thriller        251
Time taken: 1.099 seconds, Fetched: 4 row(s)
hive> SELECT genre, count(genre)
    > FROM all_movies
    > GROUP BY genre;
Query ID = root_20211123024651_66f4039e-fe8b-4d7f-8b10-2936736d2f5f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------
        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: 0.35 s
----------------------------------------------------------------------------------
OK
Action  167
Animation       30
Comedy  461
Thriller        251
Time taken: 1.074 seconds, Fetched: 4 row(s)
```

c) Write a hive query to find the number of movies released by years from *table*

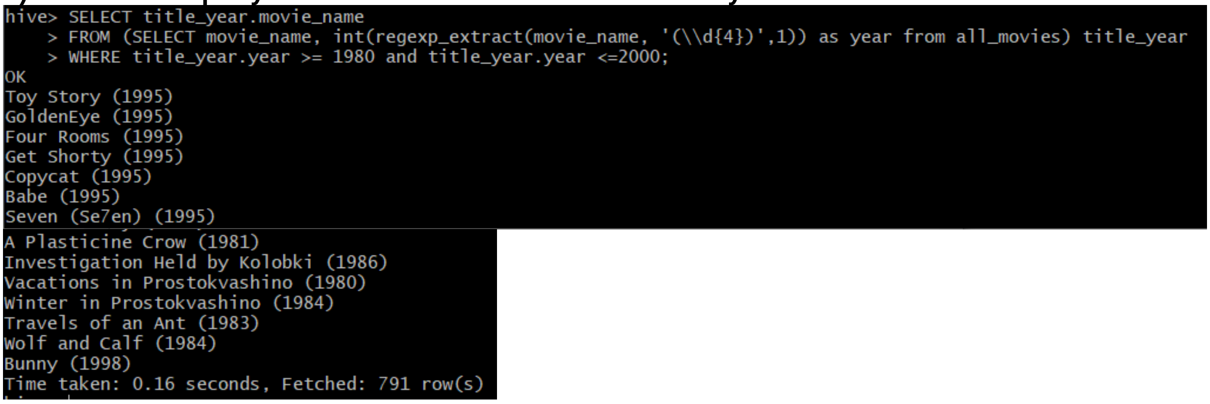


```
hive> SELECT title_year.year, count(*)
    > FROM (SELECT regexp_extract(movie_name, '(\\(\\d{4}\\))',1) as year from all_movies) title_year
    > GROUP BY title_year.year;
Query ID = root_20211123025816_5f9ce9ce-27f7-4e81-a3cd-a8010dc8d0f1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------
        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: 5.32 s
----------------------------------------------------------------------------------
OK
(1931)  1
(1933)  1
(1934)  2
(1935)  3
(2015)  2
(2016)  3
(2017)  2
Time taken: 5.916 seconds, Fetched: 73 row(s)
```

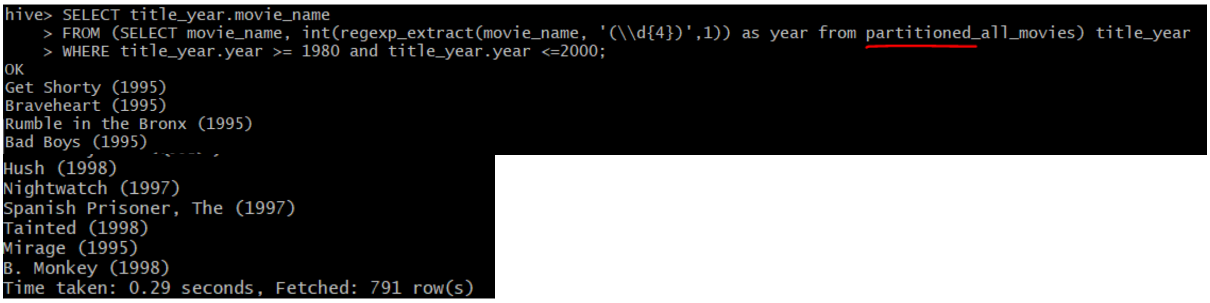


```
hive> SELECT title_year.year, count(*)
    > FROM (SELECT regexp_extract(movie_name, '(\\(\\d{4}\\))',1) as year from partitioned_all_movies) title_yea
r
    > GROUP BY title_year.year;
Query ID = root_20211123025927_8fbcb1cf-37f2-49ea-8df7-6cf89ae7113e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0028)

----------------------------------------------------------------------------------------------
        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: 4.78 s
----------------------------------------------------------------------------------------------
OK
(1931)  1
(1933)  1
(1934)  2
(1935)  3
(2014)  1
(2015)  2
(2016)  3
(2017)  2
Time taken: 5.705 seconds, Fetched: 73 row(s)
```

d) Write a hive query to find all movies released between year 1980 and 2000 from *table*

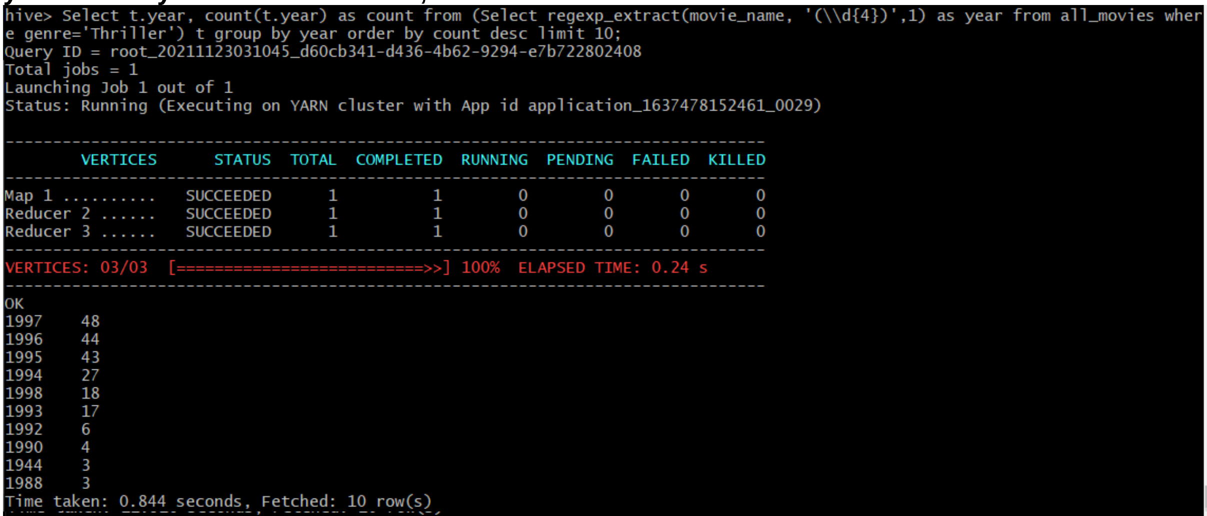


```
hive> SELECT title_year.movie_name
    > FROM (SELECT movie_name, int(regexp_extract(movie_name, '(\\d{4})',1)) as year from all_movies) title_year
    > WHERE title_year.year >= 1980 and title_year.year <=2000;
OK
Toy Story (1995)
GoldenEye (1995)
Four Rooms (1995)
Get Shorty (1995)
Copycat (1995)
Babe (1995)
Seven (Se7en) (1995)
A Plasticine Crow (1981)
Investigation Held by Kolobki (1986)
Vacations in Prostokvashino (1980)
Winter in Prostokvashino (1984)
Travels of an Ant (1983)
Wolf and Calf (1984)
Bunny (1998)
Time taken: 0.16 seconds, Fetched: 791 row(s)
```

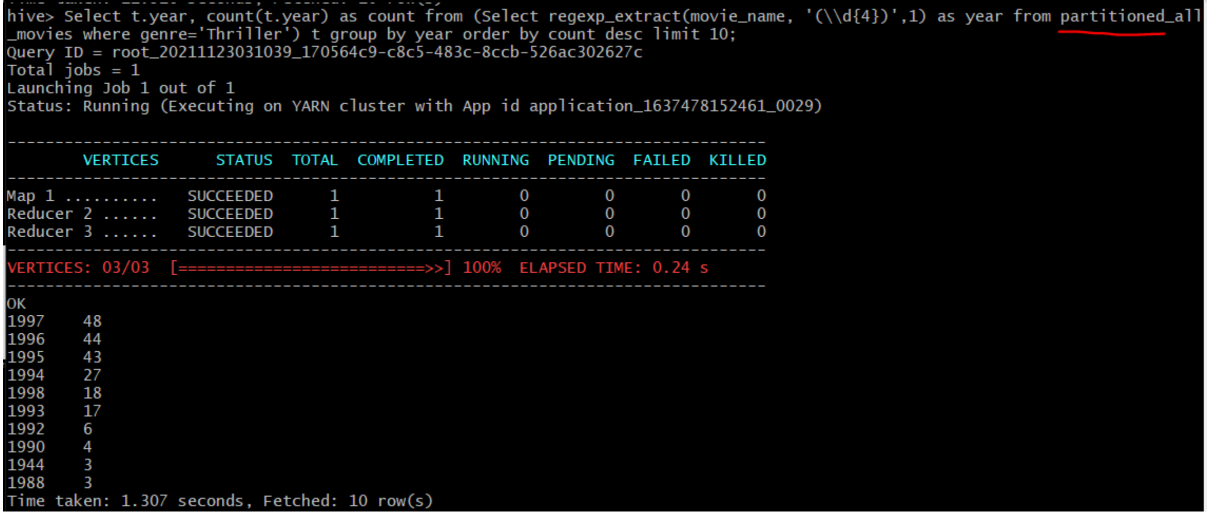


```
hive> SELECT title_year.movie_name
    > FROM (SELECT movie_name, int(regexp_extract(movie_name, '(\\d{4})',1)) as year from partitioned_all_movies) title_year
    > WHERE title_year.year >= 1980 and title_year.year <=2000;
OK
Get Shorty (1995)
Braveheart (1995)
Rumble in the Bronx (1995)
Bad Boys (1995)
Hush (1998)
Nightwatch (1997)
Spanish Prisoner, The (1997)
Tainted (1998)
Mirage (1995)
B. Monkey (1998)
Time taken: 0.29 seconds, Fetched: 791 row(s)
```

e) Provide the execution time of the query; Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from *table* where genre='Thriller') t group by year order by count desc limit 10;

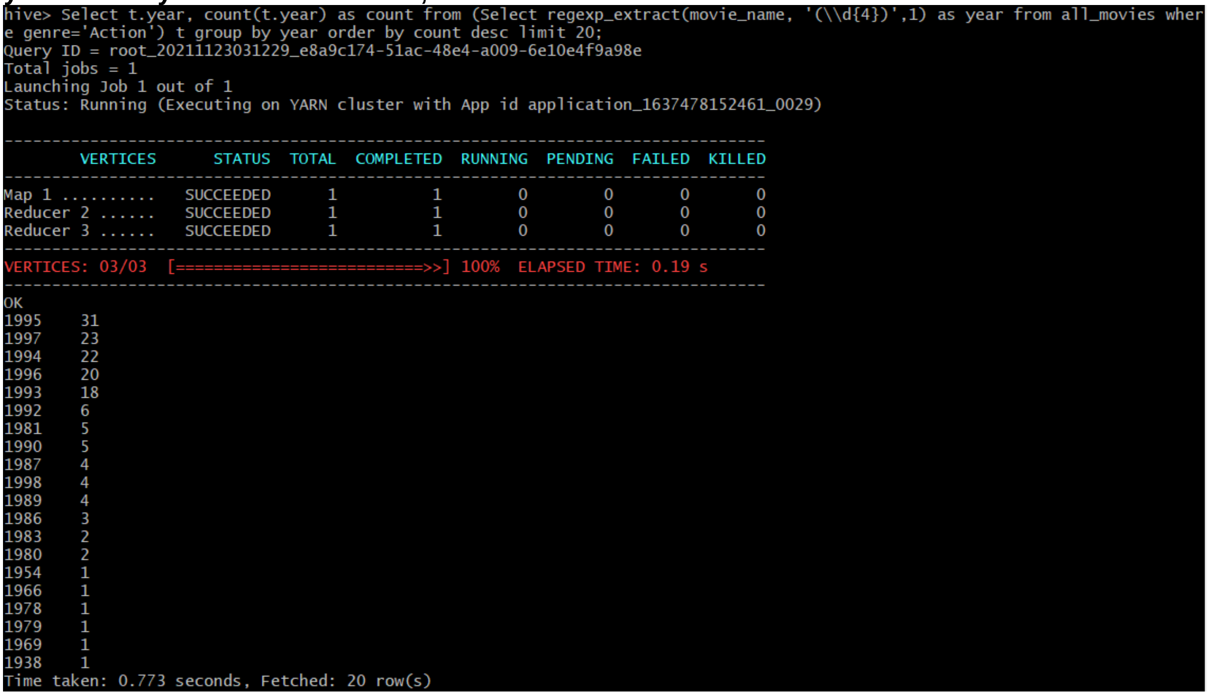


```
hive> Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from all_movies wher
e genre='Thriller') t group by year order by count desc limit 10;
Query ID = root_20211123031045_d60cb341-d436-4b62-9294-e7b722802408
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 0.24 s
----------------------------------------------------------------------------------------------
OK
1997    48
1996    44
1995    43
1994    27
1998    18
1993    17
1992    6
1990    4
1944    3
1988    3
Time taken: 0.844 seconds, Fetched: 10 row(s)
```

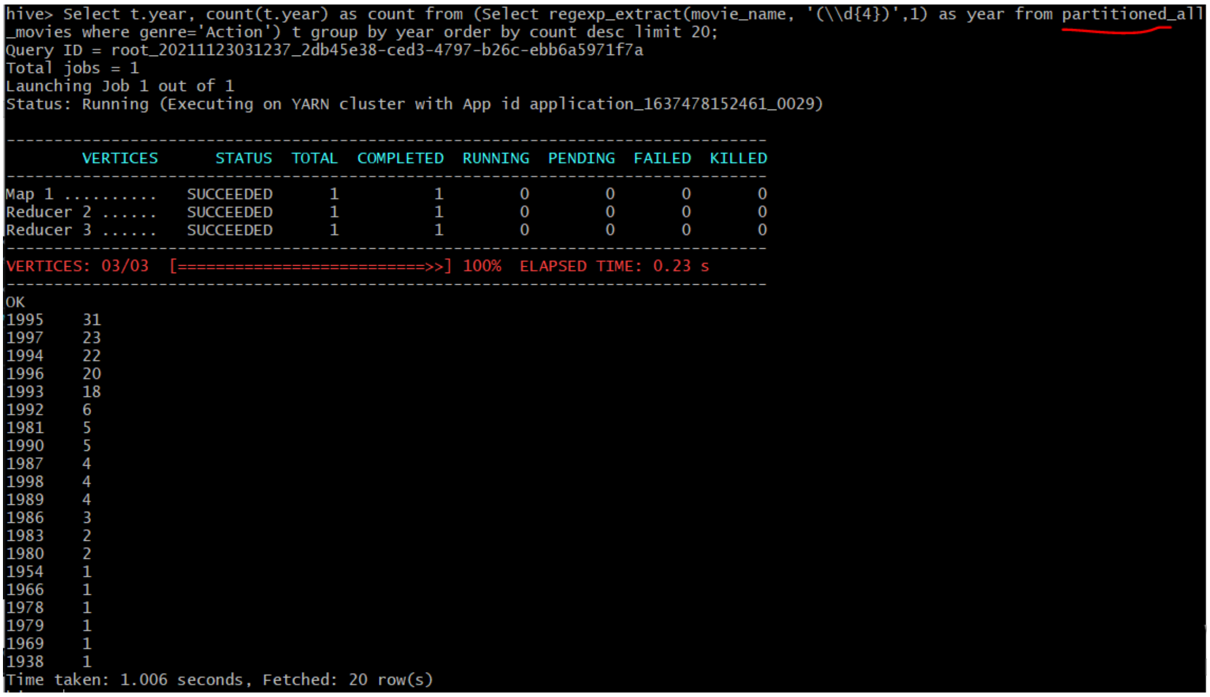


```
hive> Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from partitioned_all
_movies where genre='Thriller') t group by year order by count desc limit 10;
Query ID = root_20211123031039_170564c9-c8c5-483c-8ccb-526ac302627c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 0.24 s
----------------------------------------------------------------------------------------------
OK
1997    48
1996    44
1995    43
1994    27
1998    18
1993    17
1992    6
1990    4
1944    3
1988    3
Time taken: 1.307 seconds, Fetched: 10 row(s)
```

f) Provide the execution time of the query; Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from *table* where genre='Action') t group by year order by count desc limit 20;

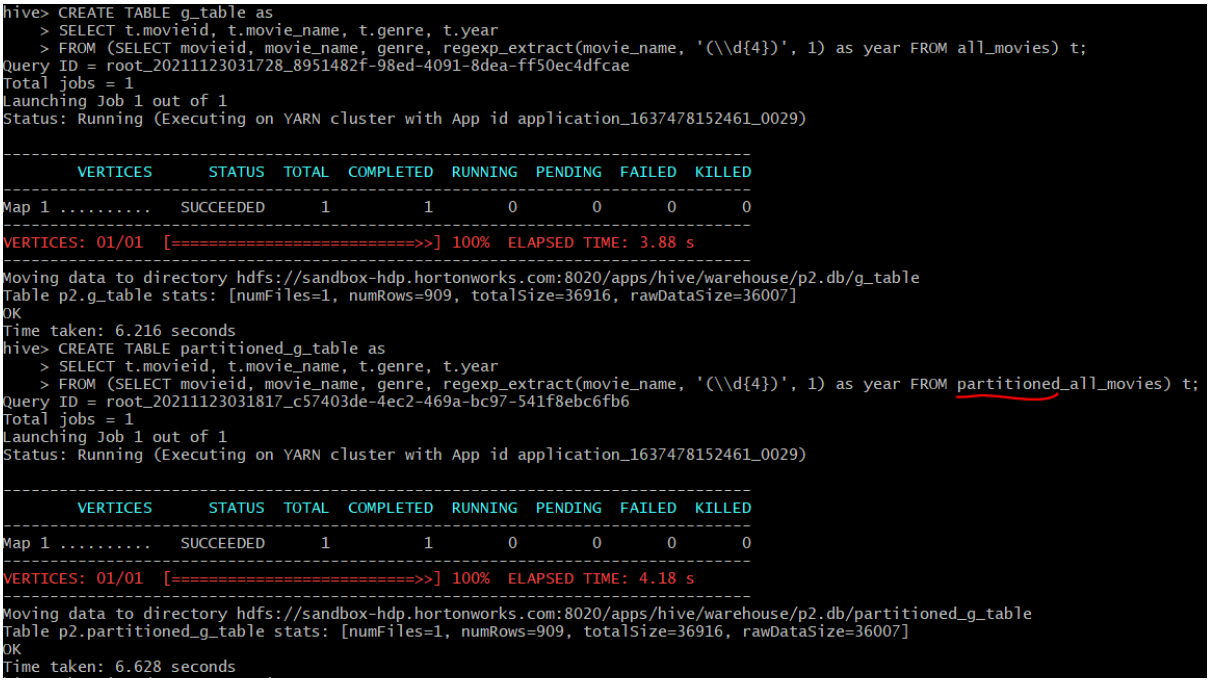


```
hive> Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from all_movies wher
e genre='Action') t group by year order by count desc limit 20;
Query ID = root_20211123031229_e8a9c174-51ac-48e4-a009-6e10e4f9a98e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 0.19 s
----------------------------------------------------------------------------------------------
OK
1995    31
1997    23
1994    22
1996    20
1993    18
1992    6
1981    5
1990    5
1987    4
1998    4
1989    4
1986    3
1983    2
1980    2
1954    1
1966    1
1978    1
1979    1
1969    1
1938    1
Time taken: 0.773 seconds, Fetched: 20 row(s)
```

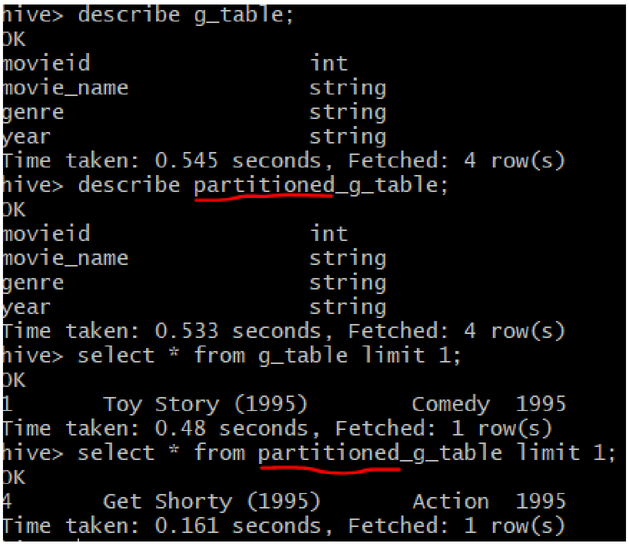


```
hive> Select t.year, count(t.year) as count from (Select regexp_extract(movie_name, '(\\d{4})',1) as year from partitioned_all
_movies where genre='Action') t group by year order by count desc limit 20;
Query ID = root_20211123031237_2db45e38-ced3-4797-b26c-ebb6a5971f7a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 0.23 s
----------------------------------------------------------------------------------------------
OK
1995    31
1997    23
1994    22
1996    20
1993    18
1992    6
1981    5
1990    5
1987    4
1998    4
1989    4
1986    3
1983    2
1980    2
1954    1
1966    1
1978    1
1979    1
1969    1
1938    1
Time taken: 1.006 seconds, Fetched: 20 row(s)
```

g) Extract movie released year from the movie title and store it by creating an additional attribute in a new table (Hint: Use regular expression and table with all information)

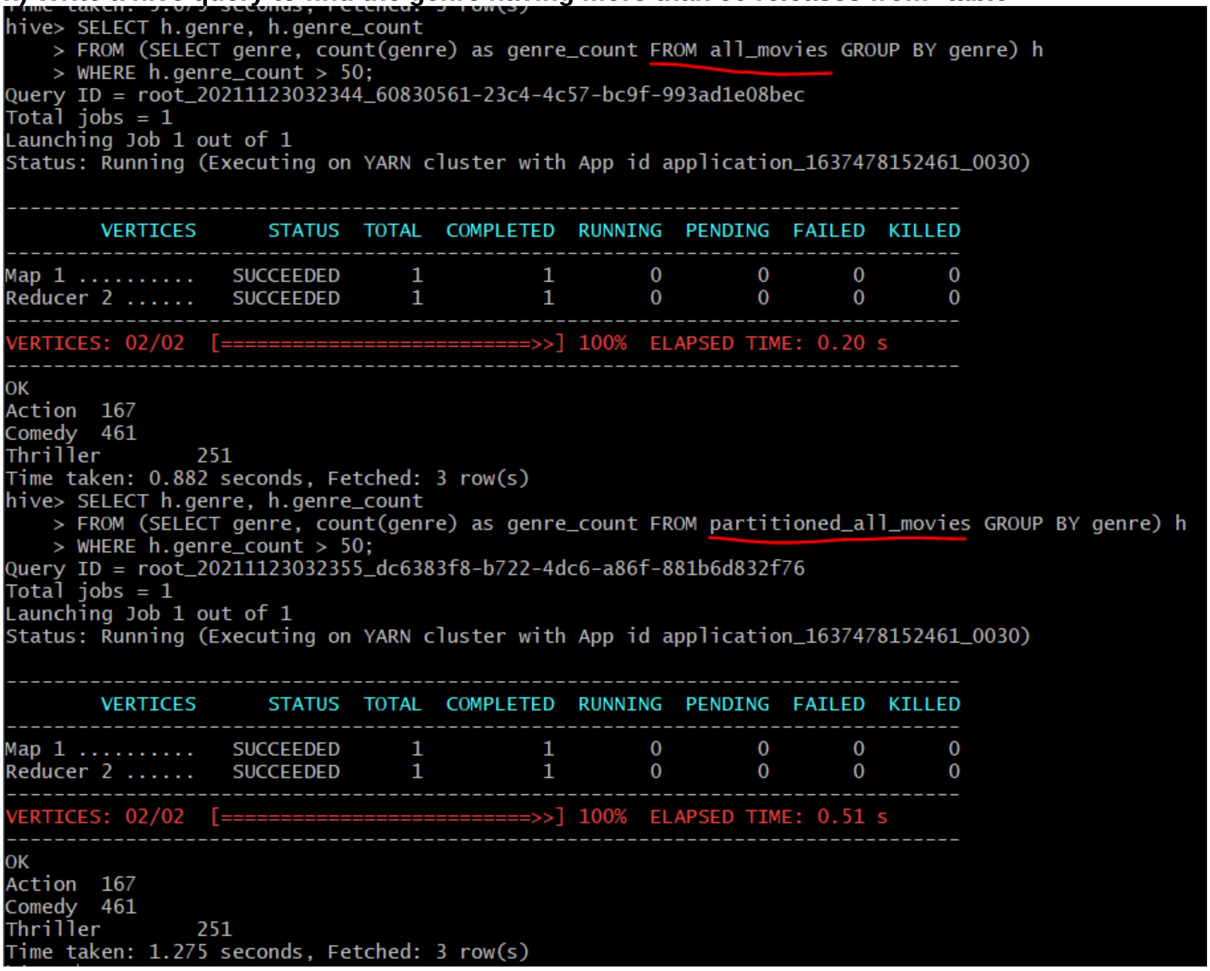


```
hive> CREATE TABLE g_table as
    > SELECT t.movieid, t.movie_name, t.genre, t.year
    > FROM (SELECT movieid, movie_name, genre, regexp_extract(movie_name, '(\\d{4})', 1) as year FROM all_movies) t;
Query ID = root_20211123031728_8951482f-98ed-4091-8dea-ff50ec4dfcae
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 3.88 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/p2.db/g_table
Table p2.g_table stats: [numFiles=1, numRows=909, totalSize=36916, rawDataSize=36007]
OK
Time taken: 6.216 seconds
hive> CREATE TABLE partitioned_g_table as
    > SELECT t.movieid, t.movie_name, t.genre, t.year
    > FROM (SELECT movieid, movie_name, genre, regexp_extract(movie_name, '(\\d{4})', 1) as year FROM partitioned_all_movies) t;
Query ID = root_20211123031817_c57403de-4ec2-469a-bc97-541f8ebc6fb6
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0029)

----------------------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 4.18 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/p2.db/partitioned_g_table
Table p2.partitioned_g_table stats: [numFiles=1, numRows=909, totalSize=36916, rawDataSize=36007]
OK
Time taken: 6.628 seconds
```



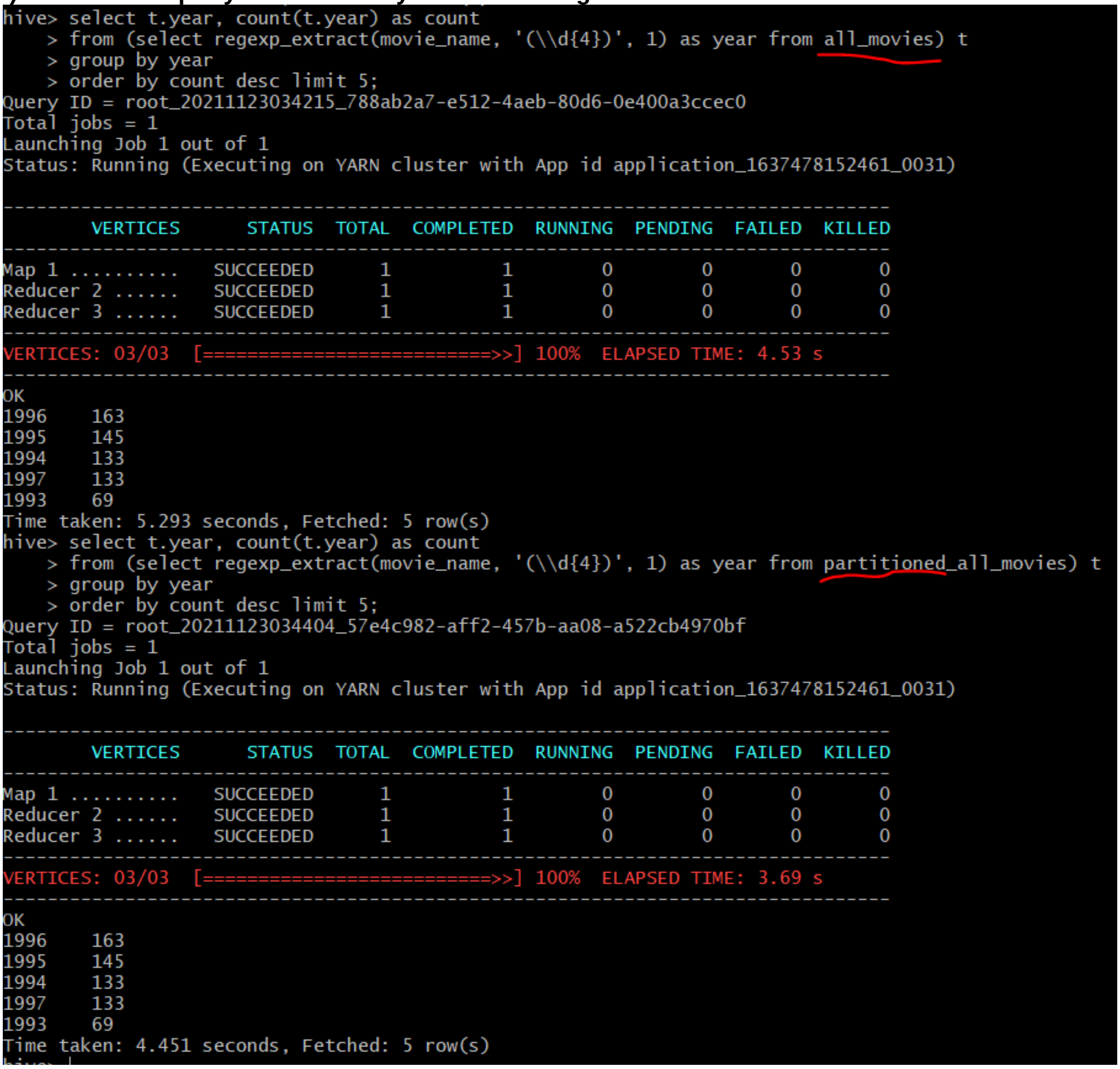
```
hive> describe g_table;
OK
movieid                 int
movie_name              string
genre                   string
year                    string
Time taken: 0.545 seconds, Fetched: 4 row(s)
hive> describe partitioned_g_table;
OK
movieid                 int
movie_name              string
genre                   string
year                    string
Time taken: 0.533 seconds, Fetched: 4 row(s)
hive> select * from g_table limit 1;
OK
1       Toy Story (1995)        Comedy  1995
Time taken: 0.48 seconds, Fetched: 1 row(s)
hive> select * from partitioned_g_table limit 1;
OK
4       Get Shorty (1995)       Action  1995
Time taken: 0.161 seconds, Fetched: 1 row(s)
```

h) Write a hive query to find the genre having more than 50 releases from *table*



```
hive> SELECT h.genre, h.genre_count
    > FROM (SELECT genre, count(genre) as genre_count FROM all_movies GROUP BY genre) h
    > WHERE h.genre_count > 50;
Query ID = root_20211123032344_60830561-23c4-4c57-bc9f-993ad1e08bec
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0030)

----------------------------------------------------------------------------------
        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: 0.20 s
----------------------------------------------------------------------------------
OK
Action  167
Comedy  461
Thriller        251
Time taken: 0.882 seconds, Fetched: 3 row(s)
hive> SELECT h.genre, h.genre_count
    > FROM (SELECT genre, count(genre) as genre_count FROM partitioned_all_movies GROUP BY genre) h
    > WHERE h.genre_count > 50;
Query ID = root_20211123032355_dc6383f8-b722-4dc6-a86f-881b6d832f76
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0030)

----------------------------------------------------------------------------------
        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: 0.51 s
----------------------------------------------------------------------------------
OK
Action  167
Comedy  461
Thriller        251
Time taken: 1.275 seconds, Fetched: 3 row(s)
```

i) Write a hive query to return the year with the highest number of releases.

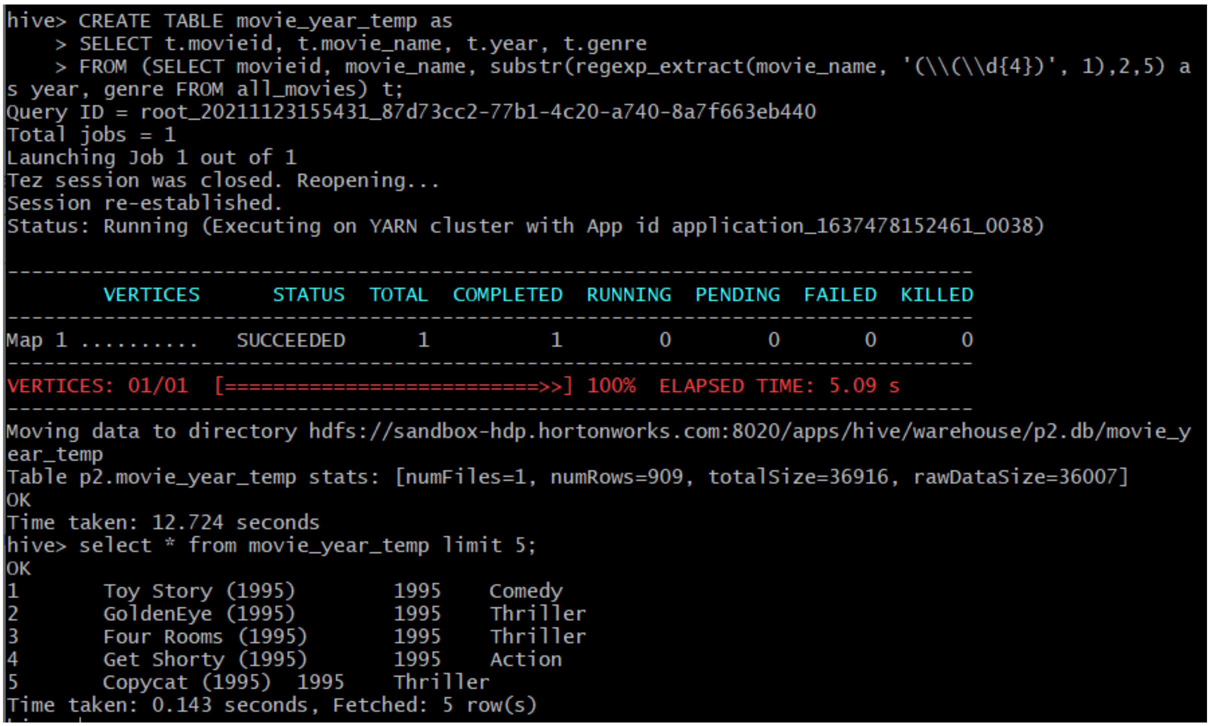


```
hive> select t.year, count(t.year) as count
    > from (select regexp_extract(movie_name, '(\\d{4})', 1) as year from all_movies) t
    > group by year
    > order by count desc limit 5;
Query ID = root_20211123034215_788ab2a7-e512-4aeb-80d6-0e400a3ccec0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0031)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 4.53 s
----------------------------------------------------------------------------------------------
OK
1996    163
1995    145
1994    133
1997    133
1993    69
Time taken: 5.293 seconds, Fetched: 5 row(s)
hive> select t.year, count(t.year) as count
    > from (select regexp_extract(movie_name, '(\\d{4})', 1) as year from partitioned_all_movies) t
    > group by year
    > order by count desc limit 5;
Query ID = root_20211123034404_57e4c982-aff2-457b-aa08-a522cb4970bf
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0031)

----------------------------------------------------------------------------------------------
        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
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 3.69 s
----------------------------------------------------------------------------------------------
OK
1996    163
1995    145
1994    133
1997    133
1993    69
Time taken: 4.451 seconds, Fetched: 5 row(s)
```

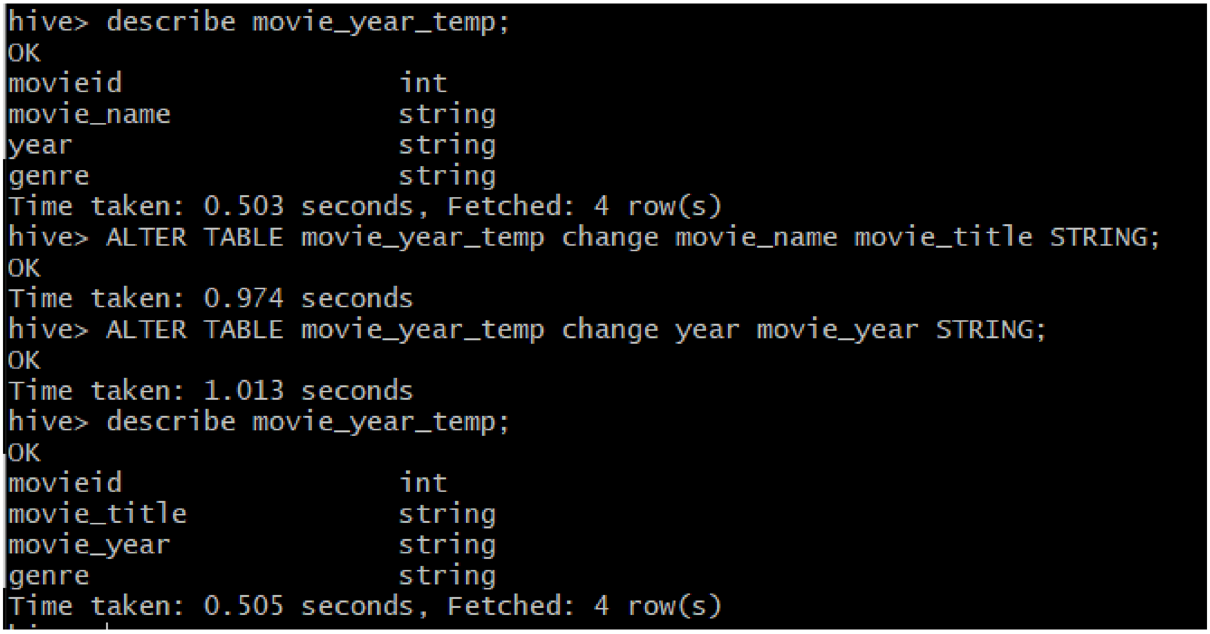
j) On which tables (partitioned or with all movies) do they actually run faster and why?

In theory, partitioned version should have better performance, however most of my results show that partitioned version takes slightly more time than ‘with all movies’ but differences were very small. The reason why the partitioned version takes slightly more time than ‘with all movies’ is because the dataset size is not too large. If the dataset is very large, partitioned version would have taken less time than ‘with all movies’ as partitioned version skip all but relevant columns.

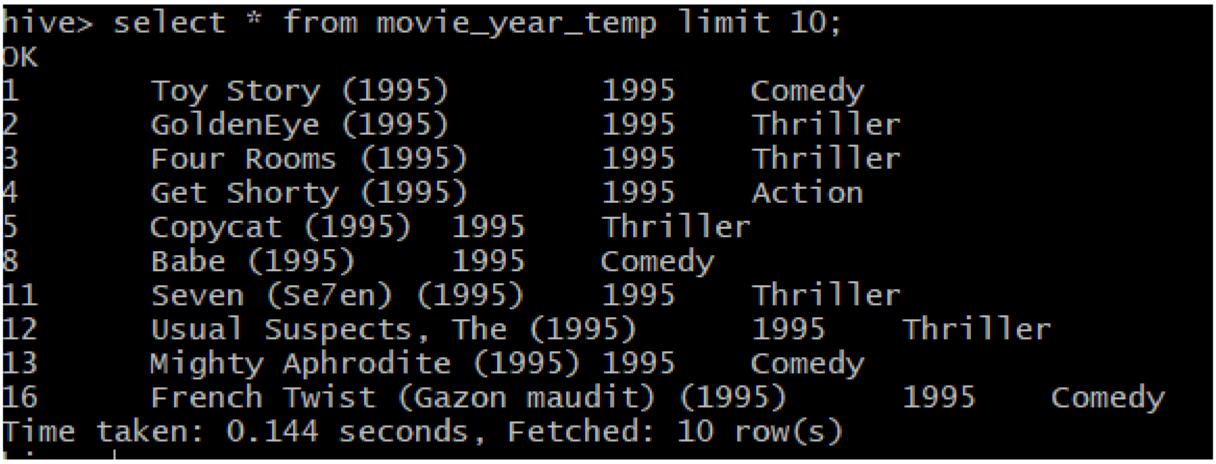
Question 6: With some help from the "select" statement in Question 5 (e) -> create a table called movie_year_temp with following columns (movieid, movie_title, movie_year, genre) (5 pts)



```
hive> CREATE TABLE movie_year_temp as
    > SELECT t.movieid, t.movie_name, t.year, t.genre
    > FROM (SELECT movieid, movie_name, substr(regexp_extract(movie_name, '(\\(\\d{4})', 1),2,5) a
s year, genre FROM all_movies) t;
Query ID = root_20211123155431_87d73cc2-77b1-4c20-a740-8a7f663eb440
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_1637478152461_0038)

----------------------------------------------------------------------------------
        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 directory hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/p2.db/movie_y
ear_temp
Table p2.movie_year_temp stats: [numFiles=1, numRows=909, totalSize=36916, rawDataSize=36007]
OK
Time taken: 12.724 seconds
hive> select * from movie_year_temp limit 5;
OK
1       Toy Story (1995)        1995    Comedy
2       GoldenEye (1995)        1995    Thriller
3       Four Rooms (1995)       1995    Thriller
4       Get Shorty (1995)       1995    Action
5       Copycat (1995)  1995    Thriller
Time taken: 0.143 seconds, Fetched: 5 row(s)
```



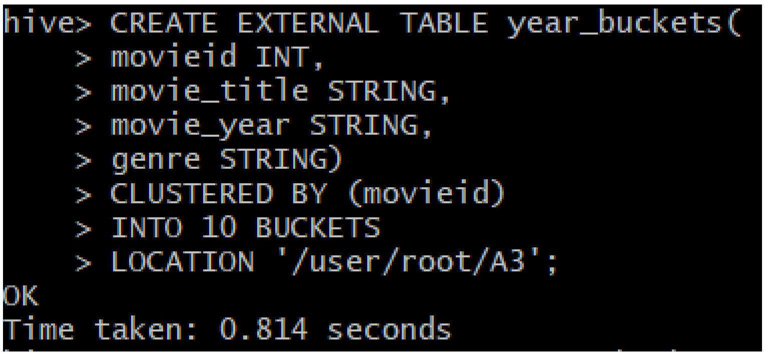
```
hive> describe movie_year_temp;
OK
movieid                 int
movie_name              string
year                    string
genre                   string
Time taken: 0.503 seconds, Fetched: 4 row(s)
hive> ALTER TABLE movie_year_temp change movie_name movie_title STRING;
OK
Time taken: 0.974 seconds
hive> ALTER TABLE movie_year_temp change year movie_year STRING;
OK
Time taken: 1.013 seconds
hive> describe movie_year_temp;
OK
movieid                 int
movie_title             string
movie_year              string
genre                   string
Time taken: 0.505 seconds, Fetched: 4 row(s)
```



```
hive> select * from movie_year_temp limit 10;
OK
1       Toy Story (1995)        1995    Comedy
2       GoldenEye (1995)        1995    Thriller
3       Four Rooms (1995)       1995    Thriller
4       Get Shorty (1995)       1995    Action
5       Copycat (1995)  1995    Thriller
8       Babe (1995)     1995    Comedy
11      Seven (Se7en) (1995)    1995    Thriller
12      Usual Suspects, The (1995)      1995    Thriller
13      Mighty Aphrodite (1995) 1995    Comedy
16      French Twist (Gazon maudit) (1995)      1995    Comedy
Time taken: 0.144 seconds, Fetched: 10 row(s)
```

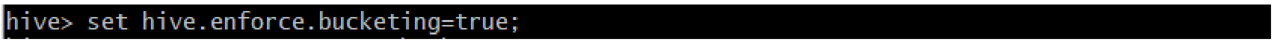
Bucketing data in hive

Question 7: Create a table called year_buckets with the same column definitions as movie_year_temp, but with 10 buckets, clustered on movieid (10 pts)

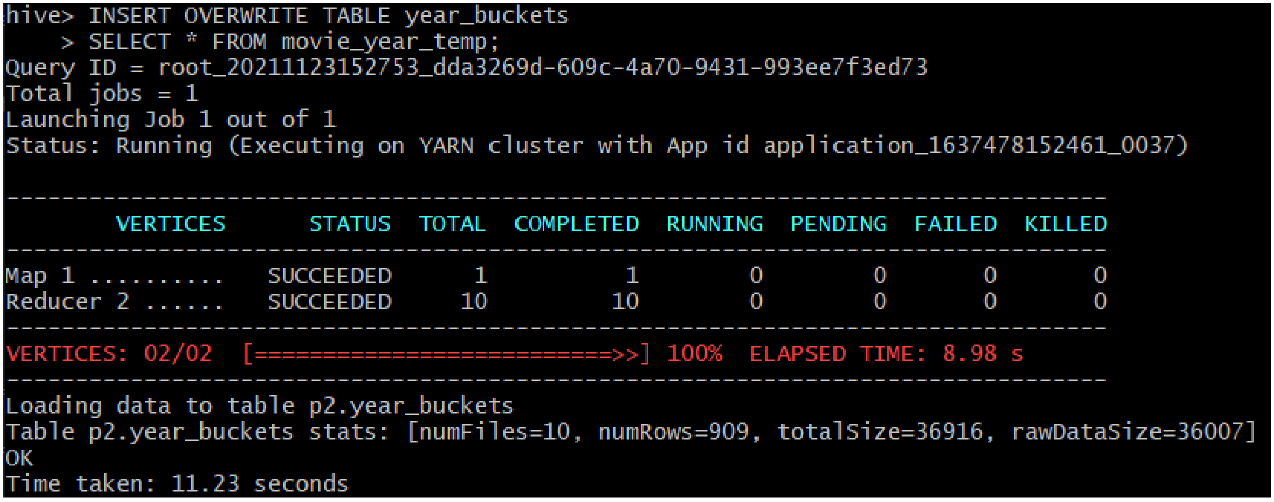


```
hive> CREATE EXTERNAL TABLE year_buckets(
    > movieid INT,
    > movie_title STRING,
    > movie_year STRING,
    > genre STRING)
    > CLUSTERED BY (movieid)
    > INTO 10 BUCKETS
    > LOCATION '/user/root/A3';
OK
Time taken: 0.814 seconds
```

Question 8: Use insert overwrite table to load the rows in movie_year_temp into year_buckets. (5 pts) (set "hive.enforce.bucketing" to true)



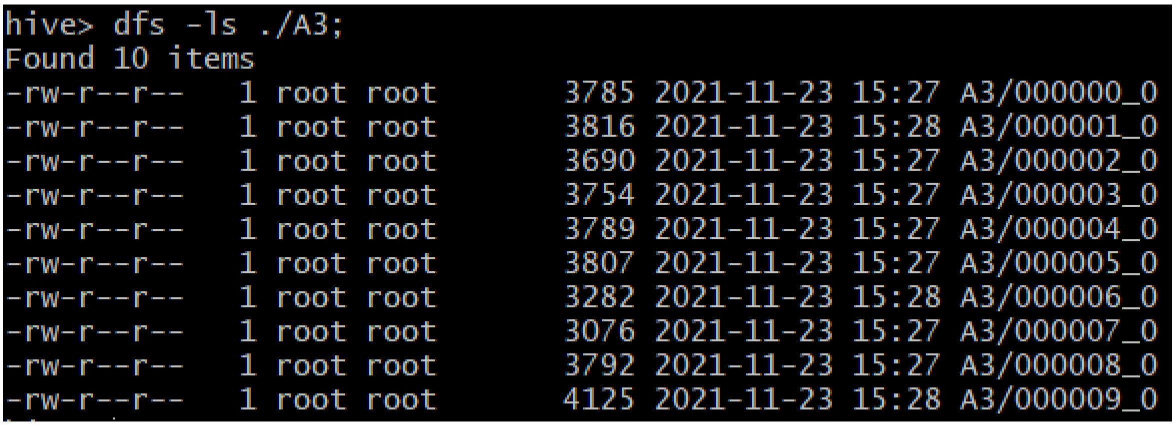
```
hive> set hive.enforce.bucketing=true;
```



```
hive> INSERT OVERWRITE TABLE year_buckets
    > SELECT * FROM movie_year_temp;
Query ID = root_20211123152753_dda3269d-609c-4a70-9431-993ee7f3ed73
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0037)

----------------------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED     10         10        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 8.98 s
----------------------------------------------------------------------------------------------
Loading data to table p2.year_buckets
Table p2.year_buckets stats: [numFiles=10, numRows=909, totalSize=36916, rawDataSize=36007]
OK
Time taken: 11.23 seconds
```

Question 9: Navigate to the location of year_buckets on HDFS. How does the partitioned table look on HDFS? Provide screenshot (5 pts)

Partitioned table look on HDFS: 10 files as there are 10 buckets in the year_bucket table

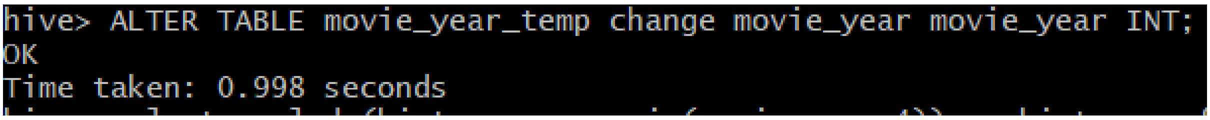


```
hive> dfs -ls ./A3;
Found 10 items
-rw-r--r--   1 root root       3785 2021-11-23 15:27 A3/000000_0
-rw-r--r--   1 root root       3816 2021-11-23 15:28 A3/000001_0
-rw-r--r--   1 root root       3690 2021-11-23 15:27 A3/000002_0
-rw-r--r--   1 root root       3754 2021-11-23 15:27 A3/000003_0
-rw-r--r--   1 root root       3789 2021-11-23 15:27 A3/000004_0
-rw-r--r--   1 root root       3807 2021-11-23 15:27 A3/000005_0
-rw-r--r--   1 root root       3282 2021-11-23 15:28 A3/000006_0
-rw-r--r--   1 root root       3076 2021-11-23 15:27 A3/000007_0
-rw-r--r--   1 root root       3792 2021-11-23 15:27 A3/000008_0
-rw-r--r--   1 root root       4125 2021-11-23 15:28 A3/000009_0
```

Apply Histogram function

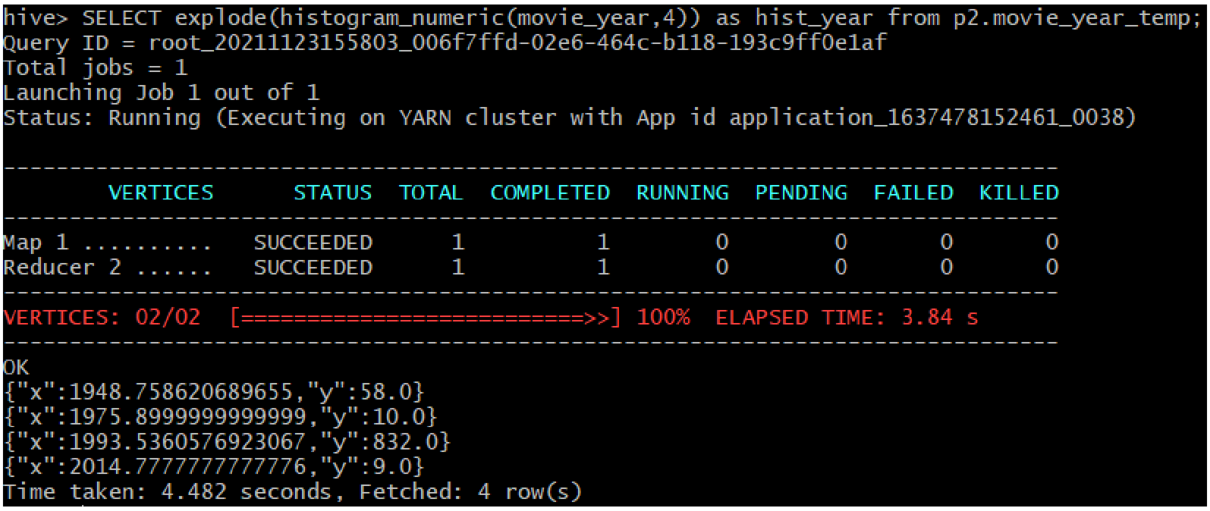
Question 10: Using the table movie_year_temp apply the histogram function (with 4 buckets) on movie_year to get the distribution of year values in the table (10 pts)

1. Changed movie_year STRING column of movie_year_temp to INT type.



```
hive> ALTER TABLE movie_year_temp change movie_year movie_year INT;
OK
Time taken: 0.998 seconds
```

2. Explode()



```
hive> SELECT explode(histogram_numeric(movie_year,4)) as hist_year from p2.movie_year_temp;
Query ID = root_20211123155803_006f7ffd-02e6-464c-b118-193c9ff0e1af
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1637478152461_0038)

----------------------------------------------------------------------------------------------
        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: 3.84 s
----------------------------------------------------------------------------------------------
OK
{"x":1948.758620689655,"y":58.0}
{"x":1975.8999999999999,"y":10.0}
{"x":1993.5360576923067,"y":832.0}
{"x":2014.7777777777776,"y":9.0}
Time taken: 4.482 seconds, Fetched: 4 row(s)
```