Demonstrate word count on an input file using MapReduce Program.

Procedure:

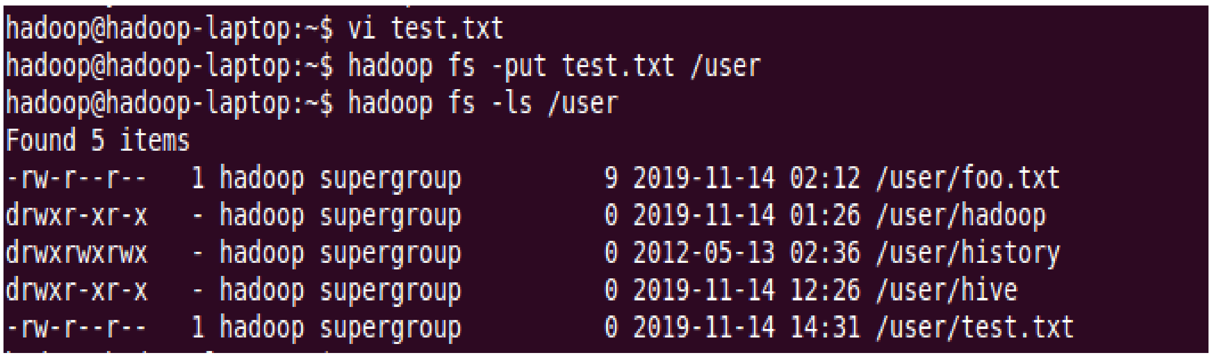
1.First create a file 'test.txt' and add any data into it.

Copy the file into HDFS by using the following command

Cmd: hadoop fs -put test.txt /user

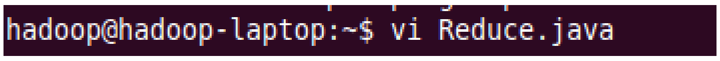
Verify if the file is copied by using the following command

Cmd: hadoop fs -ls /user



```
hadoop@hadoop-laptop:~$ vi test.txt
hadoop@hadoop-laptop:~$ hadoop fs -put test.txt /user
hadoop@hadoop-laptop:~$ hadoop fs -ls /user
Found 5 items
-rw-r--r--   1 hadoop supergroup          9 2019-11-14 02:12 /user/foo.txt
drwxr-xr-x   - hadoop supergroup          0 2019-11-14 01:26 /user/hadoop
drwxrwxrwx   - hadoop supergroup          0 2012-05-13 02:36 /user/history
drwxr-xr-x   - hadoop supergroup          0 2019-11-14 12:26 /user/hive
-rw-r--r--   1 hadoop supergroup          0 2019-11-14 14:31 /user/test.txt
```

2. Now create 3 .java files for Mapper, Reducer and main program wordcount .java



```
hadoop@hadoop-laptop:~$ vi Reduce.java
```

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

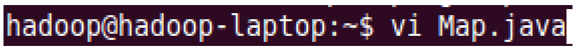
import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

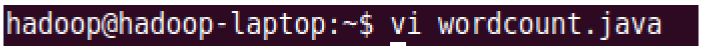
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

```
public class Reduce extends Reducer<Text,IntWritable,Text,IntWritable>

{

        private IntWritable result=new IntWritable();

        public void reduce(Text key,Iterable<IntWritable>values,Context context)throws
IOException,InterruptedException

        {

        int sum=0;

        for(IntWritable val : values)

        {

                sum+=val.get();

        }

        result.set(sum);

        context.write(key,result);

        }

        }
```



```
hadoop@hadoop-laptop:~$ vi Map.java
```

```
import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Map extends Mapper<Object,Text,Text,IntWritable>{

private final IntWritable one = new IntWritable(1);

private Text word=new Text();

public void map(Object key,Text value,Context context)throws
IOException,InterruptedException{

StringTokenizer itr=new StringTokenizer(value.toString());

while(itr.hasMoreTokens())

{

        word.set(itr.nextToken());

        context.write(word,one);

        }

        }

        }
```



```
hadoop@hadoop-laptop:~$ vi wordcount.java
```

```
import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class wordcount

{

public static void main(String[] args)throws Exception

{

        Configuration conf=new Configuration();

        Job job=Job.getInstance(conf,"wordcount");

        job.setJarByClass(wordcount.class);

        job.setMapperClass(Map.class);

        job.setCombinerClass(Reduce.class);

        job.setReducerClass(Reduce.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job,new Path(args[0]));

        FileOutputFormat.setOutputPath(job,new Path(args[1]));

        System.exit(job.waitForCompletion(true)? 0 : 1);

}

}
```

3. Now compile the above .java files by using the command:

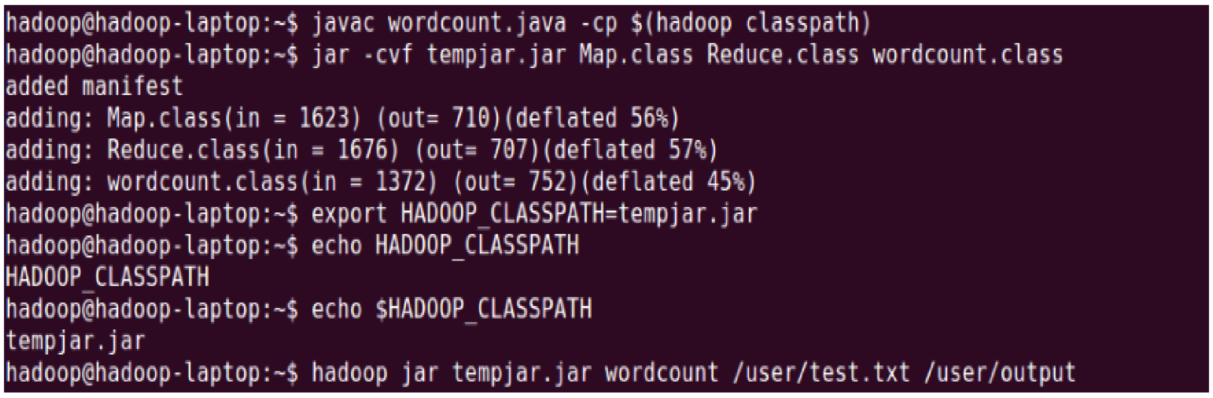
```
Cmd :javac wordcount.java -cp $(hadoop classpath)
```

4. Create a jar file from the java files using the command:

```
Cmd: jar cvf temp.jar Map.class Reduce.class wordcount.class
```

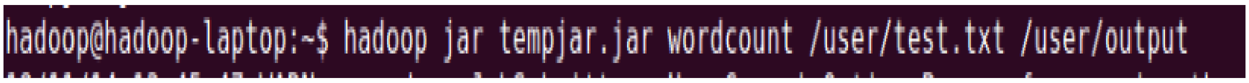
5. Use the below commands to export and execute

```
export HADOOP_CLASSPATH=tempjar.jar
echo $HADOOP_CLASSPATH
```


```
hadoop@hadoop-laptop:~$ javac wordcount.java -cp $(hadoop classpath)
hadoop@hadoop-laptop:~$ jar -cvf tempjar.jar Map.class Reduce.class wordcount.class
added manifest
adding: Map.class(in = 1623) (out= 710)(deflated 56%)
adding: Reduce.class(in = 1676) (out= 707)(deflated 57%)
adding: wordcount.class(in = 1372) (out= 752)(deflated 45%)
hadoop@hadoop-laptop:~$ export HADOOP_CLASSPATH=tempjar.jar
hadoop@hadoop-laptop:~$ echo HADOOP_CLASSPATH
HADOOP_CLASSPATH
hadoop@hadoop-laptop:~$ echo $HADOOP_CLASSPATH
tempjar.jar
hadoop@hadoop-laptop:~$ hadoop jar tempjar.jar wordcount /user/test.txt /user/output
```

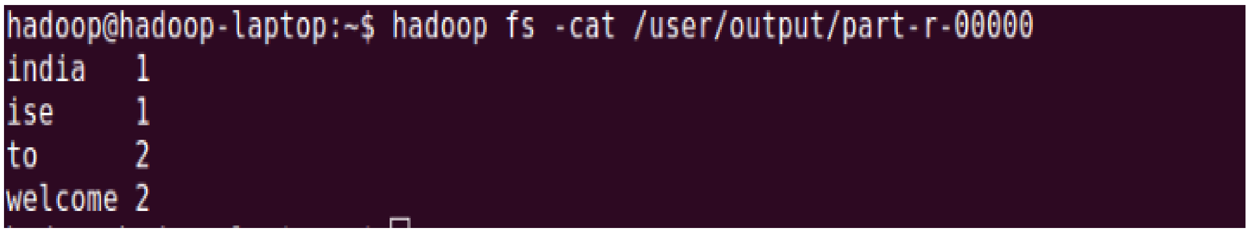
6. Use this command to start MapReduce execution:

    hadoop jar tempjar.jar wordcount /user/test.txt /user/output


```
hadoop@hadoop-laptop:~$ hadoop jar tempjar.jar wordcount /user/test.txt /user/output
```

7. Command to view the output:

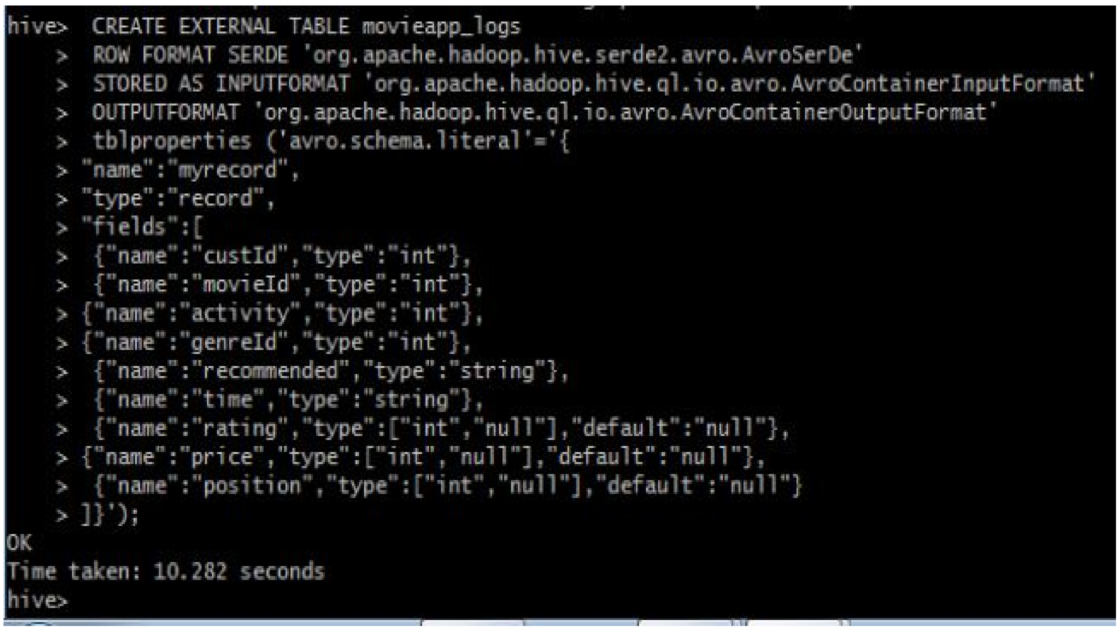
    Hadoop fs -cat /user/output/part-r-00000


```
hadoop@hadoop-laptop:~$ hadoop fs -cat /user/output/part-r-00000
india   1
ise     1
to      2
welcome 2
```

6. Create an external table(example: movieapp_log). An external table is created for the conversion text file to binary file which can then be sent to HSFS through the operating system. Avro system is used for conversion of text file to binary file.

The command used is:
CREATE EXTERNAL TABLE movieapp_logs
 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
tblproperties ('avro.schema.literal'='{"name": "my_record", "type": "record", "fields": [{"name":"custId", "type":"int"}, {"name":"movieId", "type":"int"}, {"name":"activity", "type":"int"}, {"name":"genereId", "type":"int"}, {"name":"recommended", "type":"string"}, {"name":"time", "type":"string"},

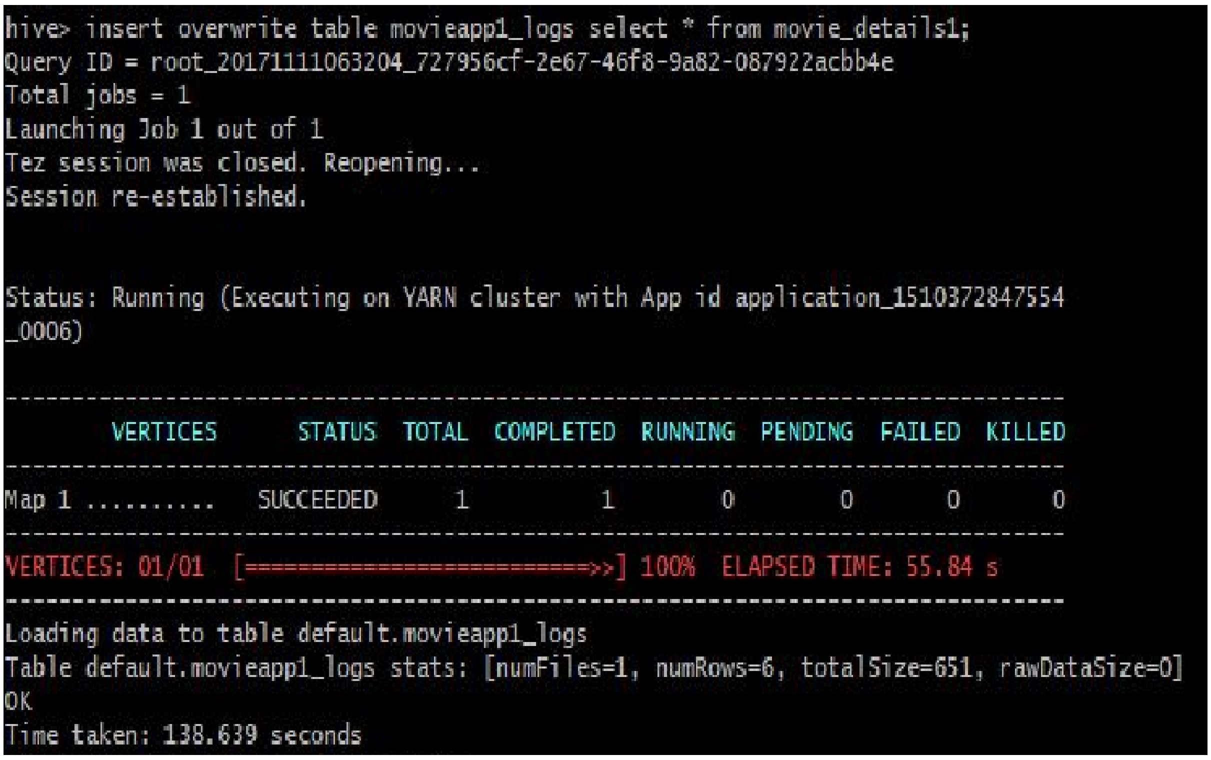
{"name":"rating","type":["int","null"],"default":"null"},
{"name":"price","type":["int","null"],"default":"null"},
{"name":"position","type":["int","null"],"default":"null"}]}');



```
hive>  CREATE EXTERNAL TABLE movieapp_logs
    >  ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
    >  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
    >  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
    >  tblproperties ('avro.schema.literal'='{
    > "name":"myrecord",
    > "type":"record",
    > "fields":[
    >  {"name":"custId","type":"int"},
    >  {"name":"movieId","type":"int"},
    > {"name":"activity","type":"int"},
    > {"name":"genreId","type":"int"},
    >  {"name":"recommended","type":"string"},
    >  {"name":"time","type":"string"},
    >  {"name":"rating","type":["int","null"],"default":"null"},
    > {"name":"price","type":["int","null"],"default":"null"},
    >  {"name":"position","type":["int","null"],"default":"null"}
    > ]}');
OK
Time taken: 10.282 seconds
hive>
```

7. Now insert the details of internally created table(cmovie_details) using insert overwrite command to the external table (loading the external table with the data)

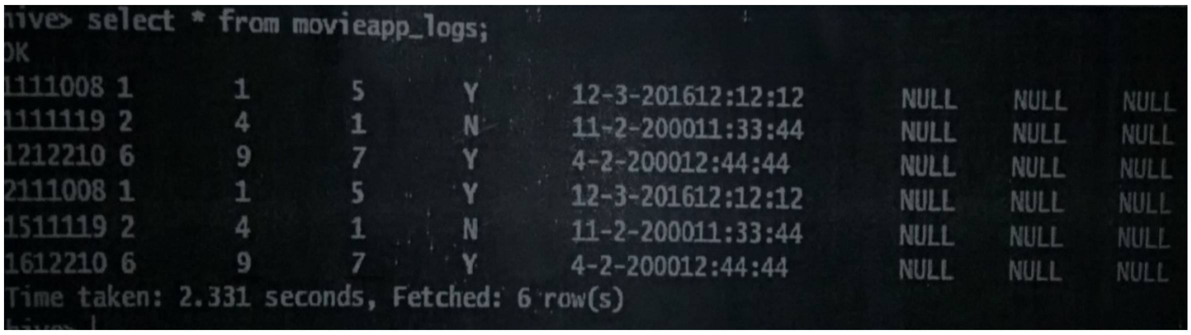
Command: insert overwrite table movieapp_logs select * from cmovie_details;



```
hive> insert overwrite table movieapp1_logs select * from movie_details1;
Query ID = root_20171111063204_727956cf-2e67-46f8-9a82-087922acbb4e
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_1510372847554
_0006)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 55.84 s
--------------------------------------------------------------------------------
Loading data to table default.movieapp1_logs
Table default.movieapp1_logs stats: [numFiles=1, numRows=6, totalSize=651, rawDataSize=0]
OK
Time taken: 138.639 seconds
```

To view the contents of external table.

Command: select * from movieapp_logs;

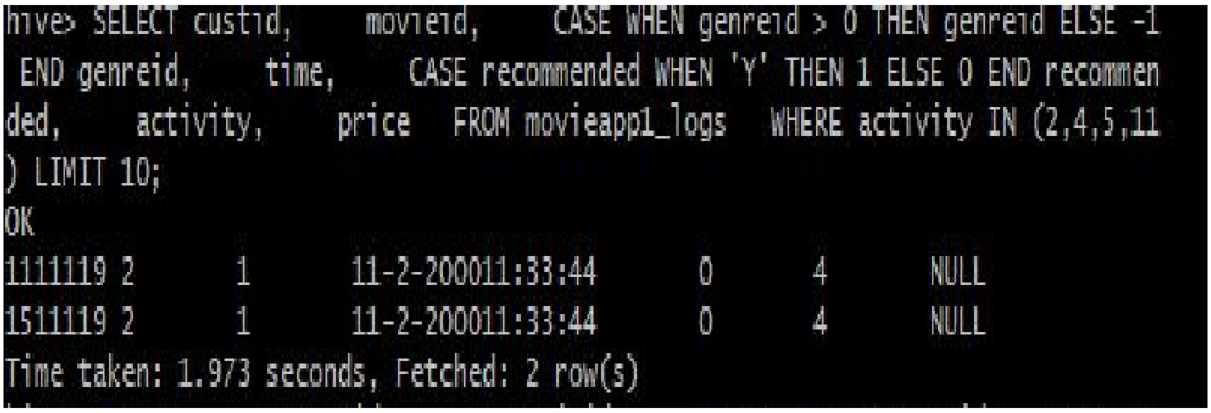


```
ive> select * from movieapp_logs;
K
111008 1      1     5     Y     12-3-201612:12:12     NULL    NULL    NULL
111119 2      4     1     N     11-2-200011:33:44     NULL    NULL    NULL
212210 6      9     7     Y     4-2-200012:44:44      NULL    NULL    NULL
111008 1      1     5     Y     12-3-201612:12:12     NULL    NULL    NULL
511119 2      4     1     N     11-2-200011:33:44     NULL    NULL    NULL
612210 6      9     7     Y     4-2-200012:44:44      NULL    NULL    NULL
ime taken: 2.331 seconds, Fetched: 6 row(s)
```

Till now we have created the tables and loaded the data into the tables. Now we are going to start the query.

Query 1: Write a query to select only those clicks which correspond to starting, browsing, completing, or purchasing movies. Use a CASE statement to transform the RECOMMENDED column into integers where 'Y' is 1 and 'N' is 0. Also, ensure GENREID is not null. Only include the first 10 rows.

Command: SELECT custid, movieid, CASE WHEN genreid> 0 THEN genreid ELSE -1 END genreid, time, CASE recommended WHEN 'Y' THEN 1 ELSE 0 END recommended, activity, price FROM movieapp_log WHERE activity IN (2,4,5,11) LIMIT 10;

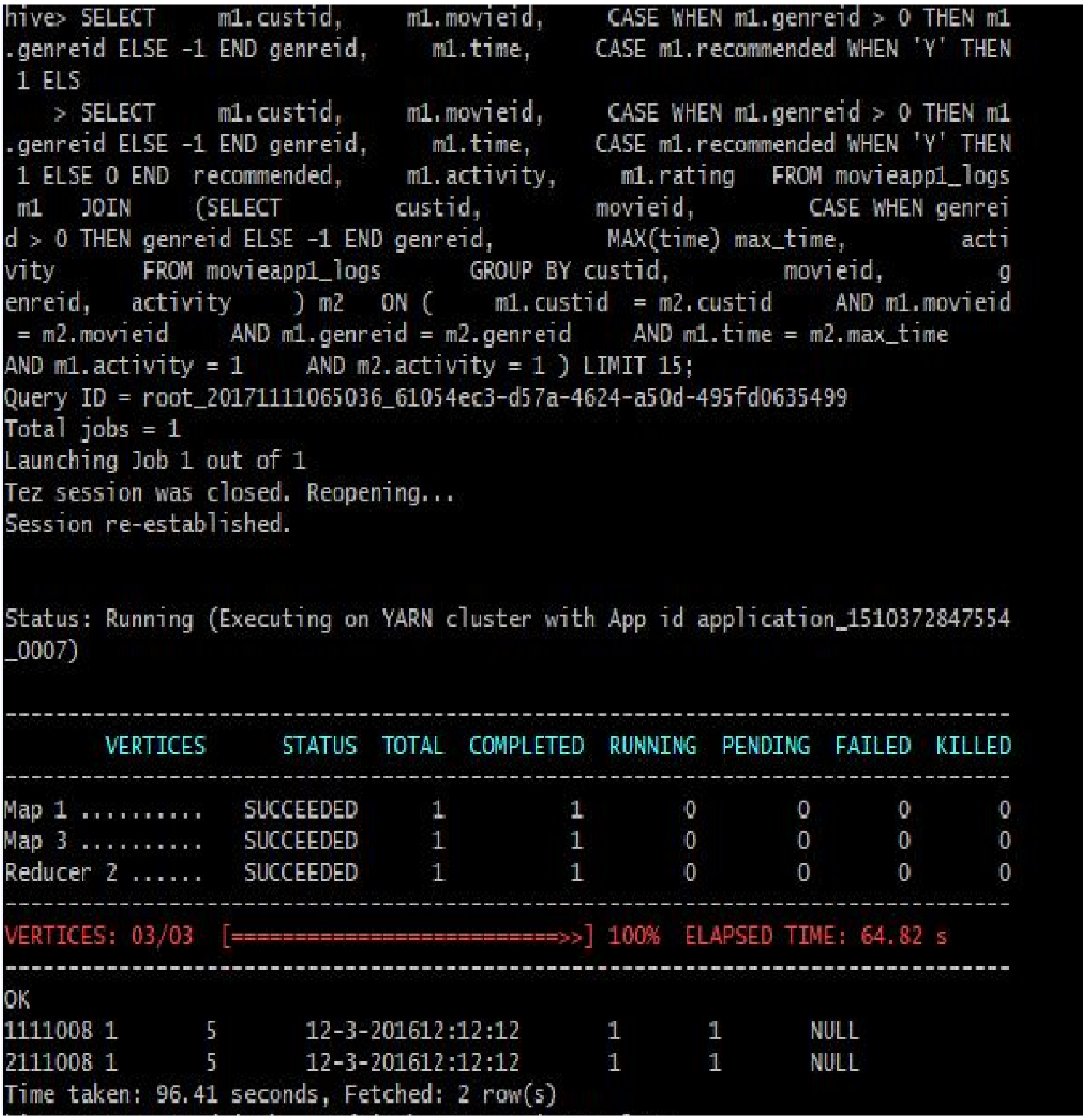


```
hive> SELECT custid,    movieid,    CASE WHEN genreid > 0 THEN genreid ELSE -1
 END genreid,    time,    CASE recommended WHEN 'Y' THEN 1 ELSE 0 END recommen
ded,    activity,    price   FROM movieapp1_logs   WHERE activity IN (2,4,5,11
) LIMIT 10;
OK
1111119 2      1      11-2-200011:33:44      0      4      NULL
1511119 2      1      11-2-200011:33:44      0      4      NULL
Time taken: 1.973 seconds, Fetched: 2 row(s)
```

Query 2: Write a query to select the customer ID, movie ID, recommended state and most recent rating for each movie.

Command: SELECT m1.custid, m1.movieid, CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid, m1.time, CASE m1.recommended WHEN 'Y' THEN

1 ELSE 0 END recommended, m1.activity, m1.rating FROM movieapp_logs m1 JOIN (SELECT custid, movieid, CASE WHEN genreid> 0 THEN genreid ELSE -1 END genreid, MAX(time) max_time, activity FROM movieapp_logs GROUP BY custid, movieid, genreid, activity) m2 ON (m1.custid = m2.custid AND m1.movieid = m2.movieid AND m1.genreid = m2.genreid AND m1.time = m2.max_time AND m1.activity = 1 AND m2.activity = 1) LIMIT 15;



```
hive> SELECT     m1.custid,     m1.movieid,     CASE WHEN m1.genreid > 0 THEN m1
.genreid ELSE -1 END genreid,     m1.time,     CASE m1.recommended WHEN 'Y' THEN
 1 ELS
    > SELECT     m1.custid,     m1.movieid,     CASE WHEN m1.genreid > 0 THEN m1
.genreid ELSE -1 END genreid,     m1.time,     CASE m1.recommended WHEN 'Y' THEN
 1 ELSE 0 END  recommended,     m1.activity,     m1.rating   FROM movieapp1_logs
 m1   JOIN     (SELECT          custid,          movieid,          CASE WHEN genrei
d > 0 THEN genreid ELSE -1 END genreid,          MAX(time) max_time,          acti
vity       FROM movieapp1_logs       GROUP BY custid,          movieid,          g
enreid,   activity     ) m2   ON (     m1.custid  = m2.custid     AND m1.movieid
 = m2.movieid     AND m1.genreid = m2.genreid     AND m1.time = m2.max_time
AND m1.activity = 1     AND m2.activity = 1 ) LIMIT 15;
Query ID = root_20171111065036_61054ec3-d57a-4624-a50d-495fd0635499
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_1510372847554
_0007)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 3 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 64.82 s
--------------------------------------------------------------------------------
OK
1111008 1       5       12-3-201612:12:12       1       1       NULL
2111008 1       5       12-3-201612:12:12       1       1       NULL
Time taken: 96.41 seconds, Fetched: 2 row(s)
```

Program 6

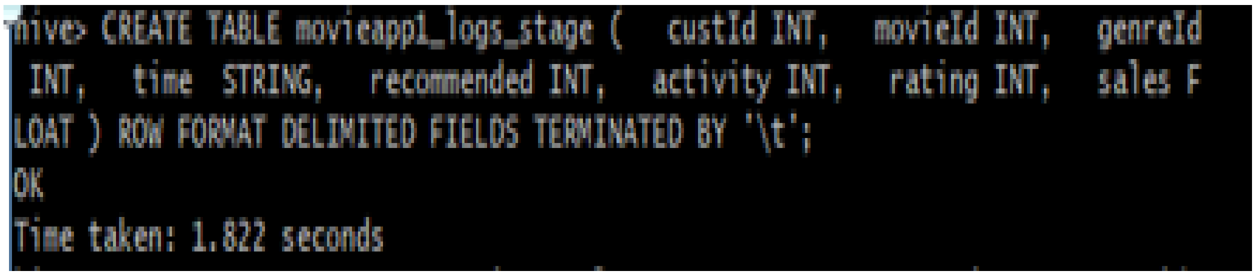
The moveapp_log_json table contains an activity column. Activity states are as follows:

- RATE_MOVIE
- COMPLETED_MOVIE
- PAUSE_MOVIE
- START_MOVIE
- BROWSE_MOVIE
- LIST_MOVIE
- SEARCH_MOVIE
- LOGIN
- LOGOUT
- INCOMPLETE_MOVIE

a. Load the results of the previous two queries into a staging table. First create the staging table:
b. Next, load the results of the queries into the staging table.

11. Load the results of the previous two queries into a staging table.First, create the staging table:

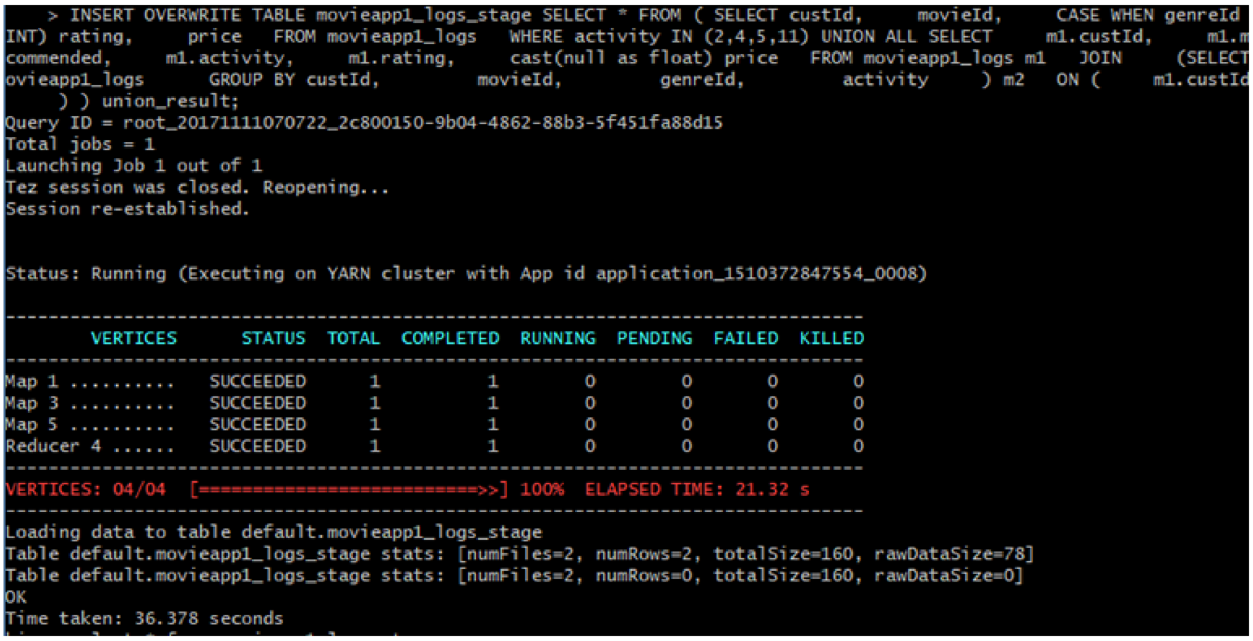
Command: CREATE TABLE movieapp_logs_stage (custId INT, movieId INT, genreId INT, time STRING, recommended INT, activity INT, rating INT, sales FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';



```
hive> CREATE TABLE movieapp1_logs_stage (   custId INT,   movieId INT,   genreId
 INT,   time  STRING,   recommended INT,   activity INT,   rating INT,   sales F
LOAT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 1.822 seconds
```

Next, load the results of the queries into the staging table:

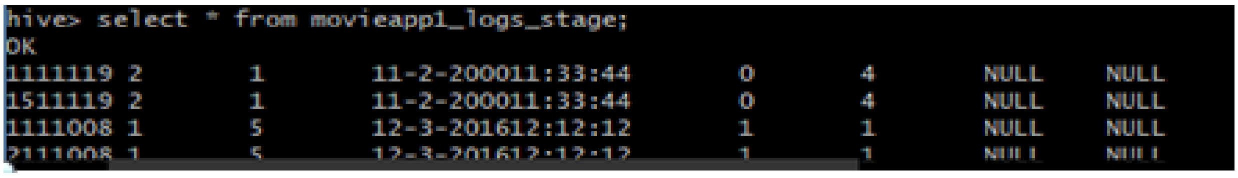
Command: INSERT OVERWRITE TABLE movieapp1_logs_stage SELECT * FROM (SELECT custId, movieId, CASE WHEN genreId > 0 THEN genreId ELSE -1 END genreId, time, CAST((CASE recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended, activity, cast(null AS INT) rating, price FROM movieapp1_logs WHERE activity IN (2,4,5,11) UNION ALL SELECT m1.custId, m1.movieId, CASE WHEN m1.genreId > 0 THEN m1.genreId ELSE -1 END genreId, m1.time, CAST((CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended, m1.activity, m1.rating, cast(null as float) price FROM movieapp1_logs m1 JOIN (SELECT custId, movieId, CASE WHEN genreId > 0 THEN genreId ELSE -1 END genreid, MAX(time) max_time, activity FROM movieapp1_logs GROUP BY custId, movieId, genreId, activity) m2 ON (m1.custId = m2.custId AND m1.movieId = m2.movieId AND m1.genreId = m2.genreId AND m1.time = m2.max_time AND m1.activity = 1 AND m2.activity = 1)) union_result;



```
    > INSERT OVERWRITE TABLE movieapp1_logs_stage SELECT * FROM ( SELECT custId,     movieId,     CASE WHEN genreId
INT) rating,     price   FROM movieapp1_logs   WHERE activity IN (2,4,5,11) UNION ALL SELECT     m1.custId,     m1.m
commended,     m1.activity,     m1.rating,     cast(null as float) price   FROM movieapp1_logs m1   JOIN     (SELECT
ovieapp1_logs     GROUP BY custId,         movieId,         genreId,         activity     ) m2   ON (     m1.custId
     ) ) union_result;
Query ID = root_20171111070722_2c800150-9b04-4862-88b3-5f451fa88d15
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_1510372847554_0008)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 3 ..........   SUCCEEDED      1          1        0        0       0       0
Map 5 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 4 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 21.32 s
--------------------------------------------------------------------------------
Loading data to table default.movieapp1_logs_stage
Table default.movieapp1_logs_stage stats: [numFiles=2, numRows=2, totalSize=160, rawDataSize=78]
Table default.movieapp1_logs_stage stats: [numFiles=2, numRows=0, totalSize=160, rawDataSize=0]
OK
Time taken: 36.378 seconds
```

12. Now view the staging table.

Command: select * from movieapp1_logs_stage;



```
hive> select * from movieapp1_logs_stage;
OK
1111119 2       1       11-2-200011:33:44       0       4       NULL    NULL
1511119 2       1       11-2-200011:33:44       0       4       NULL    NULL
1111008 1       5       12-3-201612:12:12       1       1       NULL    NULL
2111008 1       5       12-3-201612:12:12       1       1       NULL    NULL
```