CSP554—Big Data Technologies

Assignment #12 – Cassandra

Exercise 1) (4 points)
Read the article "A Big Data Modeling Methodology for Apache Cassandra" available on the blackboard in the 'Articles' section. Provide a ½ page summary including your comments and impressions.

A leading distributed database that is scalable, highly available, and transactional is Apache Cassandra. The widespread use of Cassandra in big data applications can be attributed to a number of factors, including its scalable and fault-tolerant peer-to-peer architecture, flexible and adaptable data model that developed from the BigTable data model, declarative and user-friendly Cassandra Query Language (CQL), and extremely efficient write and read access paths that allow critical big data applications to be always on, scale to millions of transactions per second, and handle node and even entity handling.

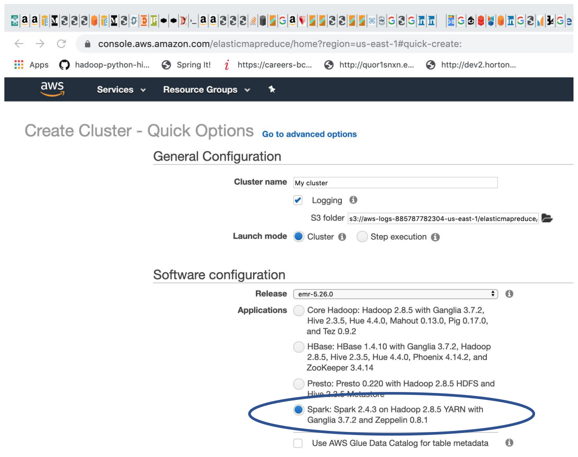
The main goals of RDBMS are to comprehend data, organize it into relations, reduce data redundancy, and prevent data duplication. In Cassandra, application queries are used as a starting point to achieve superior write and read performance.

Based on queries specified in an application workflow, a conceptual data model is translated into a logical data model. By using data modeling principles, mapping rules, and mapping patterns, we create the query-driven conceptual-to-logical data model mapping.

The four sections of Data Modeling Principles are DMP1 (Know Your Data), DMP2 (Know Your Queries), DMP3 (Data Nesting), and DMP4 (Data Duplication). After that, we define Mapping Rules, which is comprised of 5 sections. Entities and relationships, equality search attributes, inequality search attributes, ordering attributes, and MR1 through MR5 are listed (Key Attributes) We create mapping patterns that form the foundation for automating Cassandra database schema construction based on the aforementioned mapping criteria.

Exercise 2) (3 points)

<u>Step A – Start an EMR cluster</u>
Start up an EMR/Hadoop cluster as previously, but instead of choosing the "Core Hadoop" configuration chose the "Spark" configuration (see below), otherwise proceed as before.


console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Create Cluster - Quick Options Go to advanced options

General Configuration

Cluster name: My cluster

[✓] Logging

S3 folder: s3://aws-logs-885787782304-us-east-1/elasticmapreduce/

Launch mode: [✓ Cluster] Step execution

Software configuration

Release: emr-5.26.0

Applications:
- Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.5, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.5, Hue 4.4.0, Phoenix 4.14.2, and ZooKeeper 3.4.14
- Presto: Presto 0.220 with Hadoop 2.8.5 HDFS and Hive 2.3.5 Metastore
- [✓ Spark: Spark 2.4.3 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1]

[ ] Use AWS Glue Data Catalog for table metadata

<u>Step B – Install the Cassandra database software and start it</u>

Open up a terminal connection to your EMR master node. Over the course of this exercise, you will need to open up three separate terminal connections to your EMR master node. This is the first, which we will call Cass-Term.

Enter the following two command:

```
wget https://archive.apache.org/dist/cassandra/3.11.2/apache-cassandra-3.11.2-bin.tar.gz
tar -xzvf apache-cassandra-3.11.2-bin.tar.gz
```

Note, this will create a new directory (apache-cassandra-3.11.2) holding the Cassandra software release.

Then enter this command to start Cassandra (lots of diagnostic messages will appear):

```
apache-cassandra-3.11.2/bin/cassandra &
```

<u>Step C – Run the Cassandra interactive command line interface</u>

Open a second terminal connection to the EMR master node. Going forward we will call this terminal connection: Cqlsh-Term.

Enter the following into this terminal to start the command line interface csqlsh:

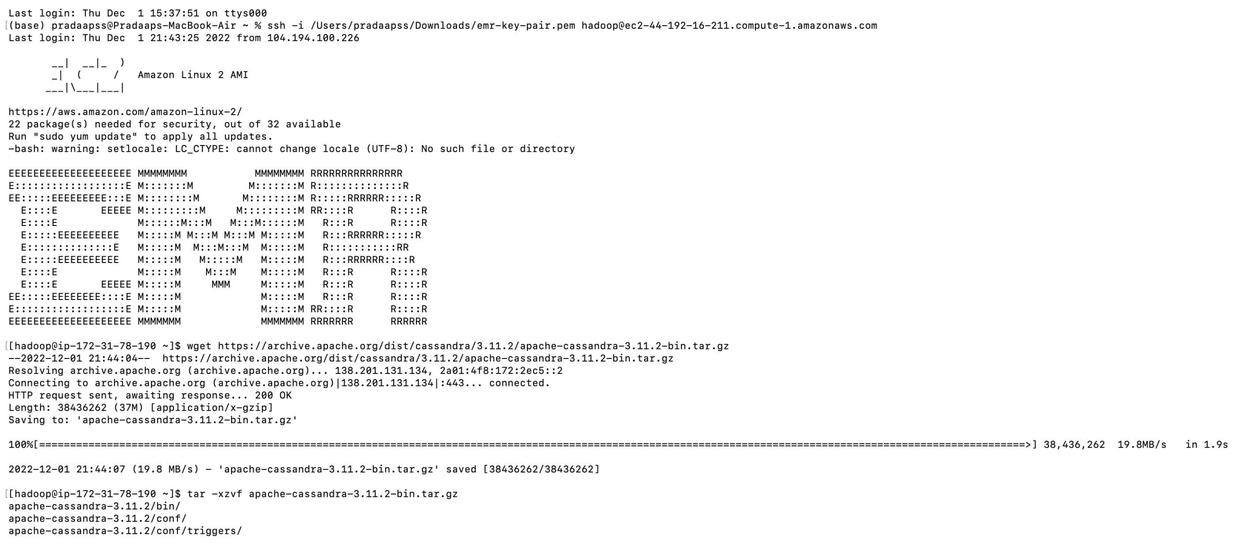
```
apache-cassandra-3.11.2/bin/cqlsh
```

<u>Step D – Prepare to edit your Cassandra code</u>

Open a third terminal connection to the EMR master node. Going forward we will call this terminal connection: Edit-Term.

You will use this terminal window to run the ‘vi’ editor to create your Cassandra code files. See the “Free Books and Chapters” section of our blackboard site for information on how to use the ‘vi’ editor.

As an alternative you could edit your Cassandra code files on your PC/MAC using a text editor like “notepad” or “textedit” "and then ‘scp’ them to the EMR mater node.



```
Last login: Thu Dec  1 15:37:51 on ttys000
[(base) pradaapss@Pradaaps-MacBook-Air ~ % ssh -i /Users/pradaapss/Downloads/emr-key-pair.pem hadoop@ec2-44-192-16-211.compute-1.amazonaws.com
Last login: Thu Dec  1 21:43:25 2022 from 104.194.100.226

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
22 package(s) needed for security, out of 32 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[[hadoop@ip-172-31-78-190 ~]$ wget https://archive.apache.org/dist/cassandra/3.11.2/apache-cassandra-3.11.2-bin.tar.gz
--2022-12-01 21:44:04--  https://archive.apache.org/dist/cassandra/3.11.2/apache-cassandra-3.11.2-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38436262 (37M) [application/x-gzip]
Saving to: 'apache-cassandra-3.11.2-bin.tar.gz'

100%[==================================================================================================================================================>] 38,436,262  19.8MB/s   in 1.9s

2022-12-01 21:44:07 (19.8 MB/s) - 'apache-cassandra-3.11.2-bin.tar.gz' saved [38436262/38436262]

[[hadoop@ip-172-31-78-190 ~]$ tar -xzvf apache-cassandra-3.11.2-bin.tar.gz
apache-cassandra-3.11.2/bin/
apache-cassandra-3.11.2/conf/
apache-cassandra-3.11.2/conf/triggers/
```

a) Create a file in your working (home) directory called init.cql using your Edit-term (or using your PC/MAC and then scp it to the EMR master node) and enter the following command. Use your IIT id as the name of your keyspace… For example, if your id is A1234567, then replace <IIT id> below with that value:

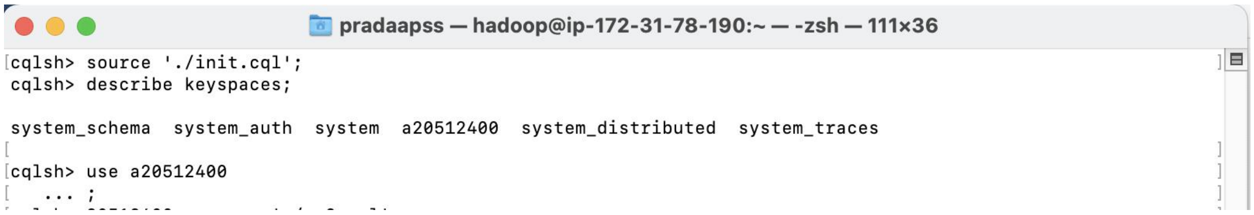
CREATE KEYSPACE <IIT id> WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };

For example, you might write:

CREATE KEYSPACE A1234567 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };

b) Then execute this file in the CQL shell using the Cqlsh-Term as follows…

source ‘./init.cql’;

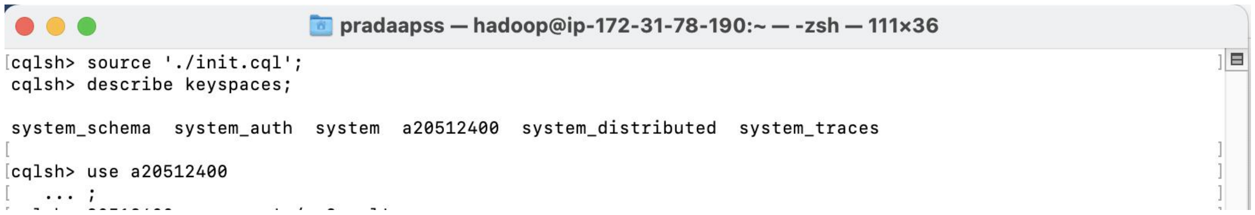


pradaapss — hadoop@ip-172-31-78-190:~ — -zsh — 111×36

```
[cqlsh> source './init.cql';
cqlsh> describe keyspaces;

system_schema  system_auth  system  a20512400  system_distributed  system_traces
[
[cqlsh> use a20512400
[   ... ;
```

c) To check if your script file has created a keyspace execute the following in the CQL shell:

describe keyspaces;



```
[cqlsh> source './init.cql';
 cqlsh> describe keyspaces;

 system_schema  system_auth  system  a20512400  system_distributed  system_traces
[
[cqlsh> use a20512400
[   ... ;
```

d) At this point you have created a keyspace unique to you. So, make that keyspace the default by entering the following into the CQL shell:

USE <IIT id>;

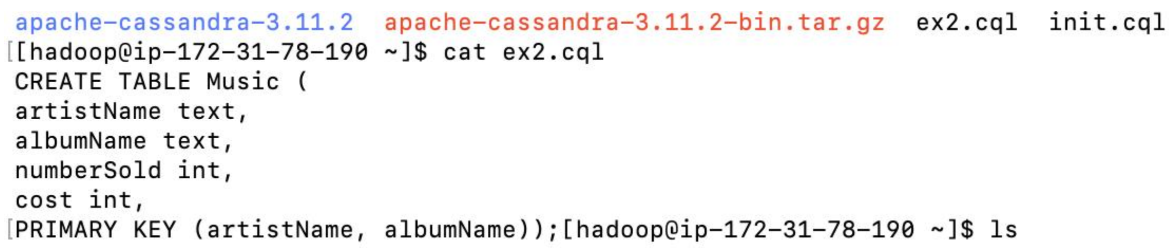
For example,

USE A1234567;

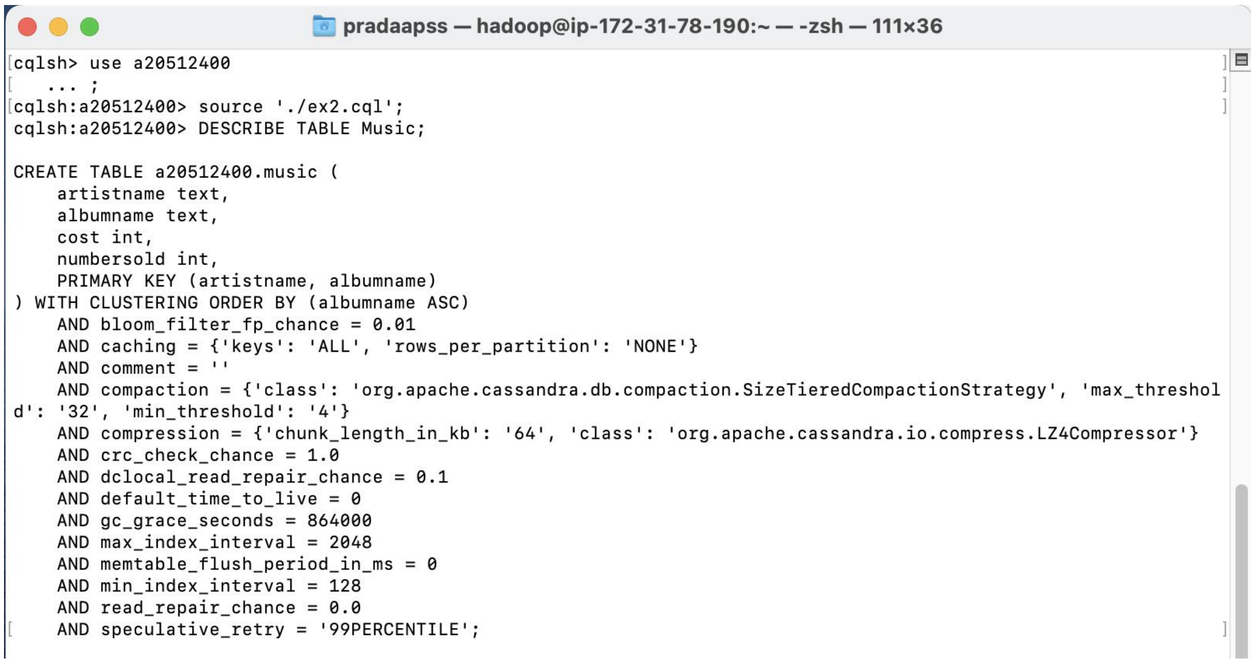
Now create a file in your working directory called ex2.cql using the Edit-Term (or PC/MAC and scp). In this file write the command to create a table named ‘Music’ with the following characteristics:

Attribute Name	Attribute Type	Primary Key / Cluster Key
artistName	text	Primary Key
albumName	text	Cluster Key
numberSold	int	Non Key Column
Cost	int	Non Key Column

Execute ex2.cql in the CQL shell. Then execute the shell command ‘DESCRIBE TABLE Music;’ and include the output as the result of this exercise.



```
 apache-cassandra-3.11.2  apache-cassandra-3.11.2-bin.tar.gz  ex2.cql  init.cql
[[hadoop@ip-172-31-78-190 ~]$ cat ex2.cql
 CREATE TABLE Music (
 artistName text,
 albumName text,
 numberSold int,
 cost int,
[PRIMARY KEY (artistName, albumName));[hadoop@ip-172-31-78-190 ~]$ ls
```



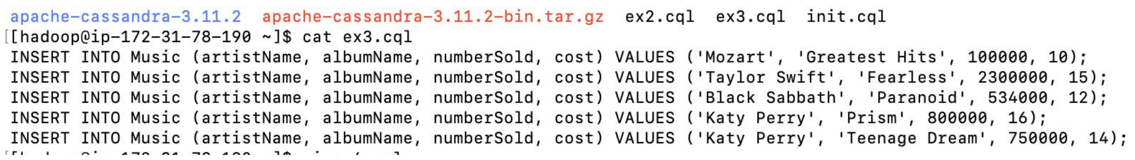
```
[cqlsh> use a20512400
[   ... ;
[cqlsh:a20512400> source './ex2.cql';
 cqlsh:a20512400> DESCRIBE TABLE Music;

 CREATE TABLE a20512400.music (
     artistname text,
     albumname text,
     cost int,
     numbersold int,
     PRIMARY KEY (artistname, albumname)
 ) WITH CLUSTERING ORDER BY (albumname ASC)
     AND bloom_filter_fp_chance = 0.01
     AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
     AND comment = ''
     AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshol
 d': '32', 'min_threshold': '4'}
     AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
     AND crc_check_chance = 1.0
     AND dclocal_read_repair_chance = 0.1
     AND default_time_to_live = 0
     AND gc_grace_seconds = 864000
     AND max_index_interval = 2048
     AND memtable_flush_period_in_ms = 0
     AND min_index_interval = 128
     AND read_repair_chance = 0.0
[    AND speculative_retry = '99PERCENTILE';
```

Exercise 3) (3 points)

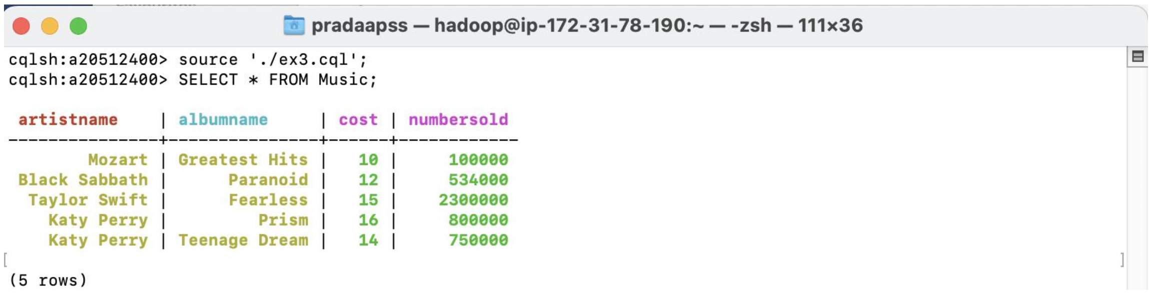
Now create a file in your working directory called ex3.cql using the Edit-Term. In this file write the commands to insert the following records into table 'Music'…

artistName	albumName	numberSold	cost
Mozart	Greatest Hits	100000	10
Taylor Swift	Fearless	2300000	15
Black Sabbath	Paranoid	534000	12
Katy Perry	Prism	800000	16
Katy Perry	Teenage Dream	750000	14

a) Execute ex3.cql. Provide the content of this file as the result of this exercise.
b) Execute the command 'SELECT * FROM Music;' and provide the output of this command as another result of the exercise.

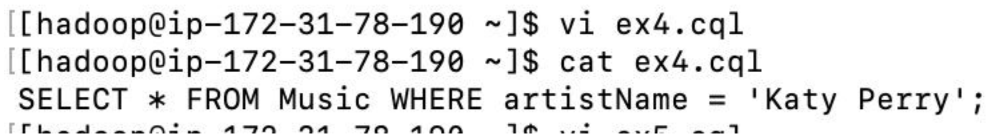


```
 apache-cassandra-3.11.2  apache-cassandra-3.11.2-bin.tar.gz  ex2.cql  ex3.cql  init.cql
[[hadoop@ip-172-31-78-190 ~]$ cat ex3.cql
 INSERT INTO Music (artistName, albumName, numberSold, cost) VALUES ('Mozart', 'Greatest Hits', 100000, 10);
 INSERT INTO Music (artistName, albumName, numberSold, cost) VALUES ('Taylor Swift', 'Fearless', 2300000, 15);
 INSERT INTO Music (artistName, albumName, numberSold, cost) VALUES ('Black Sabbath', 'Paranoid', 534000, 12);
 INSERT INTO Music (artistName, albumName, numberSold, cost) VALUES ('Katy Perry', 'Prism', 800000, 16);
 INSERT INTO Music (artistName, albumName, numberSold, cost) VALUES ('Katy Perry', 'Teenage Dream', 750000, 14);
```

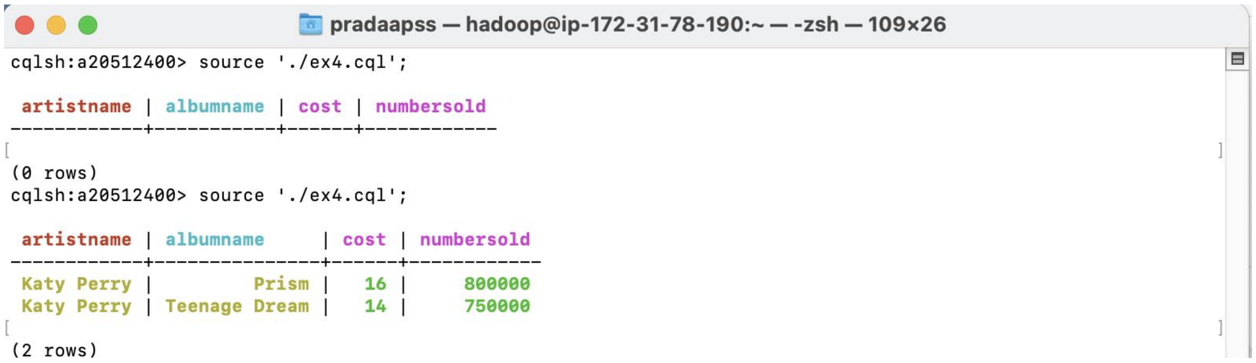


```
cqlsh:a20512400> source './ex3.cql';
cqlsh:a20512400> SELECT * FROM Music;

 artistname    | albumname     | cost | numbersold
---------------+---------------+------+------------
        Mozart | Greatest Hits |   10 |     100000
  Black Sabbath |      Paranoid |   12 |     534000
  Taylor Swift |      Fearless |   15 |    2300000
    Katy Perry |         Prism |   16 |     800000
    Katy Perry | Teenage Dream |   14 |     750000

(5 rows)
```

Exercise 4) (2 points)
Now create a file in your working directory called ex4.cql using the Edit-Term. In this file write the commands to query and output only Katy Perry songs. Execute ex4.cql. Provide the content of this file and output of executing this file as the result of this exercise.

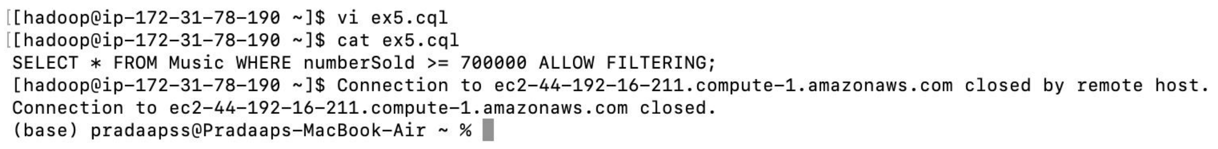


```
[[hadoop@ip-172-31-78-190 ~]$ vi ex4.cql
[[hadoop@ip-172-31-78-190 ~]$ cat ex4.cql
 SELECT * FROM Music WHERE artistName = 'Katy Perry';
```

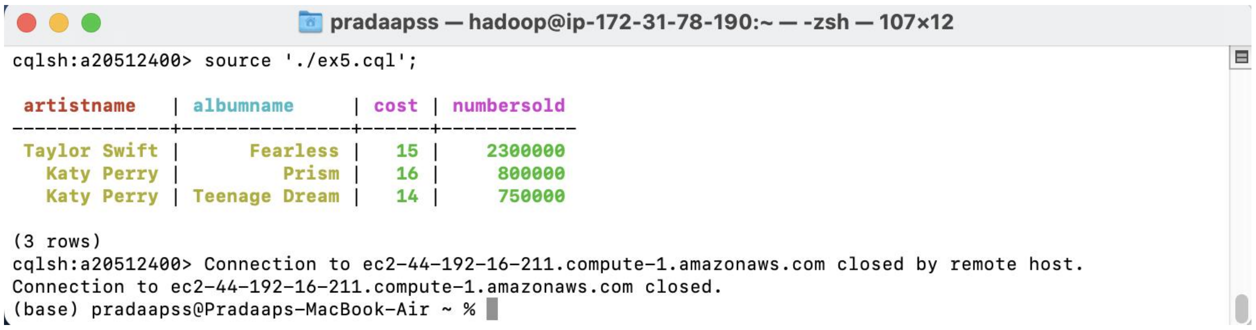


```
cqlsh:a20512400> source './ex4.cql';

 artistname | albumname | cost | numbersold
------------+-----------+------+------------

(0 rows)
cqlsh:a20512400> source './ex4.cql';

 artistname | albumname     | cost | numbersold
------------+---------------+------+------------
 Katy Perry |         Prism |   16 |     800000
 Katy Perry | Teenage Dream |   14 |     750000

(2 rows)
```

Exercise 5) (2 points)
Now create a file in your working directory called ex5.cql using the Edit-Term. In this file write the commands to query only albums that have sold 700000 copies or more. Execute ex5.cql. Provide the content of this file and the output of executing this file as the result of this exercise.



```
[[hadoop@ip-172-31-78-190 ~]$ vi ex5.cql
[[hadoop@ip-172-31-78-190 ~]$ cat ex5.cql
 SELECT * FROM Music WHERE numberSold >= 700000 ALLOW FILTERING;
 [hadoop@ip-172-31-78-190 ~]$ Connection to ec2-44-192-16-211.compute-1.amazonaws.com closed by remote host.
 Connection to ec2-44-192-16-211.compute-1.amazonaws.com closed.
 (base) pradaapss@Pradaaps-MacBook-Air ~ %
```



```
cqlsh:a20512400> source './ex5.cql';

 artistname   | albumname     | cost | numbersold
--------------+---------------+------+------------
 Taylor Swift |      Fearless |   15 |    2300000
   Katy Perry |         Prism |   16 |     800000
   Katy Perry | Teenage Dream |   14 |     750000

(3 rows)
cqlsh:a20512400> Connection to ec2-44-192-16-211.compute-1.amazonaws.com closed by remote host.
Connection to ec2-44-192-16-211.compute-1.amazonaws.com closed.
(base) pradaapss@Pradaaps-MacBook-Air ~ %
```

Remember to terminate your EMR cluster when you complete this assignment.