CSP 554 Big Data Technologies

Assignment – #12

Shiva Sankar Modala(A20517528)

Exercise 1

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.

Ans:

Apache Cassandra is a leading transactional, scalable, and highly available distributed database. It is known to manage some of the world’s largest datasets on clusters with many thousands of nodes deployed across multiple data centers. Numerous factors contribute to Cassandra's widespread use in big data applications, including its fault-tolerant and scalable peer-to-peer architecture, adaptable and flexible data model that evolved from the BigTable data model, declarative and user-friendly Cassandra Query Language (CQL), and extremely efficient write and read access paths that enable critical big data applications to remain always on, scale to millions of transactions per second, handle node and even entire clusters.

A database schema in Cassandra is represented by a keyspace that serves as a top-level namespace where all other data objects, such as tables, reside. Within a keyspace, a set of CQL tables is defined to store and query data for a particular application.

Data comprehension, relational organization, data reduction, and data duplication prevention are the main objectives of RDBMS. In Cassandra, application queries are initially run to improve write and read performance.

A conceptual data model is converted into a logical data model using queries defined in an application workflow. We define the query-driven conceptual-to-logical data model mapping utilizing the concepts, rules, and patterns of data modeling.

Logical Data Modeling:

Data Modeling Principles

- DMP1 (Know Your Data): The first key to successful database design is understanding the data, which is captured with a conceptual data model.

- DMP2 (Know Your Queries): The second key to successful database design is queries, which are captured via an application workflow model.
- DMP3 (Data Nesting): The third key to successful database design is data nesting. Data nesting refers to a technique that organizes multiple entities (usually of the same type) together based on a known criterion.
- DMP4 (Data Duplication): The fourth key to successful database design is data duplication. Duplicating data in Cassandra across multiple tables, partitions, and rows is a common practice that is required to efficiently support different queries over the same data.

Mapping Rules:

- MR1 (Entities and Relationships). Entity and relationship types of maps to tables, while entities and relationships map to table rows.
- MR2 (Equality Search Attributes). Equality search at- tributes, which are used in a query predicate, map to the prefix columns of a table primary key.
- MR3 (Inequality Search Attributes). An inequality search attribute, which is used in a query predicate, maps to a table clustering key column.
- MR4 (Ordering Attributes). Ordering attributes, which are specified in a query, map to clustering key columns with ascending or descending clustering order as prescribed by the query.
- MR5 (Key Attributes). Key attribute types of maps to primary key columns.

Mapping Patterns:

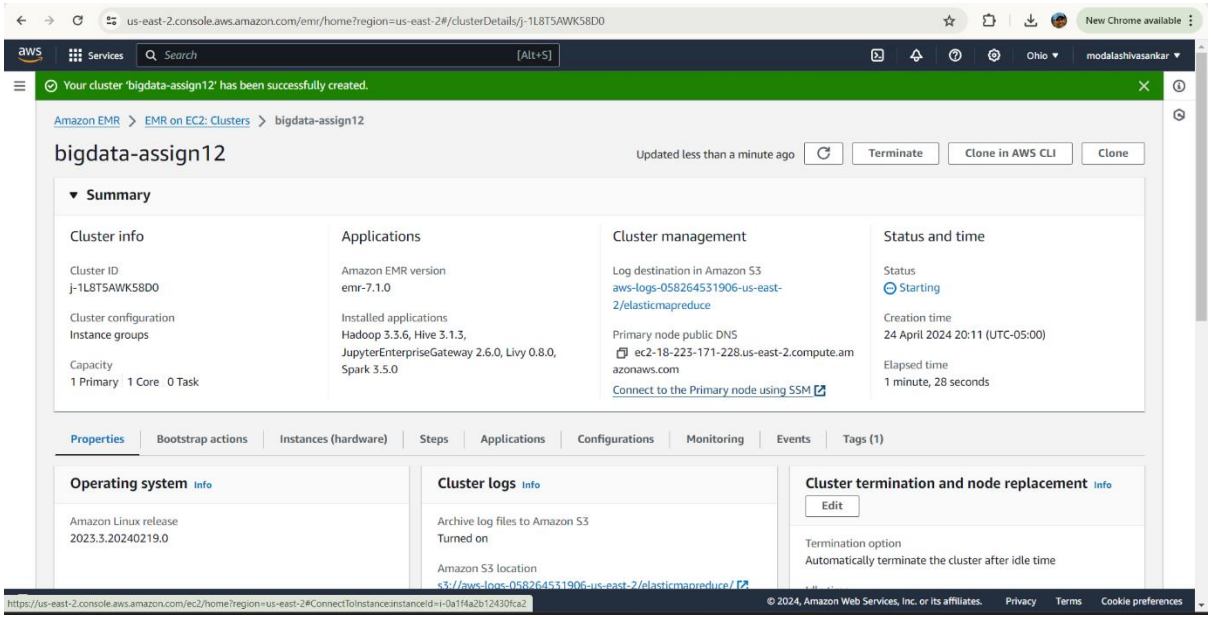
Given a query and a conceptual data model subgraph that is relevant to the query, each mapping pattern defines final table schema design without the need to apply individual mapping rules. While we define a number of different mapping patterns [9], due to space limitations, we only present one mapping pattern and one example.

For Apache Cassandra, they presented a strict query-driven data modeling paradigm. In various instances, such as query-driven schema design, data nesting, and data duplication, it was demonstrated that the methodology was significantly different from the conventional relational data modeling approach. In addition to outlining the purpose of physical data modeling, they also suggested a cutting-edge visualization method called Chebotko Diagrams that may be utilized to visualize intricate

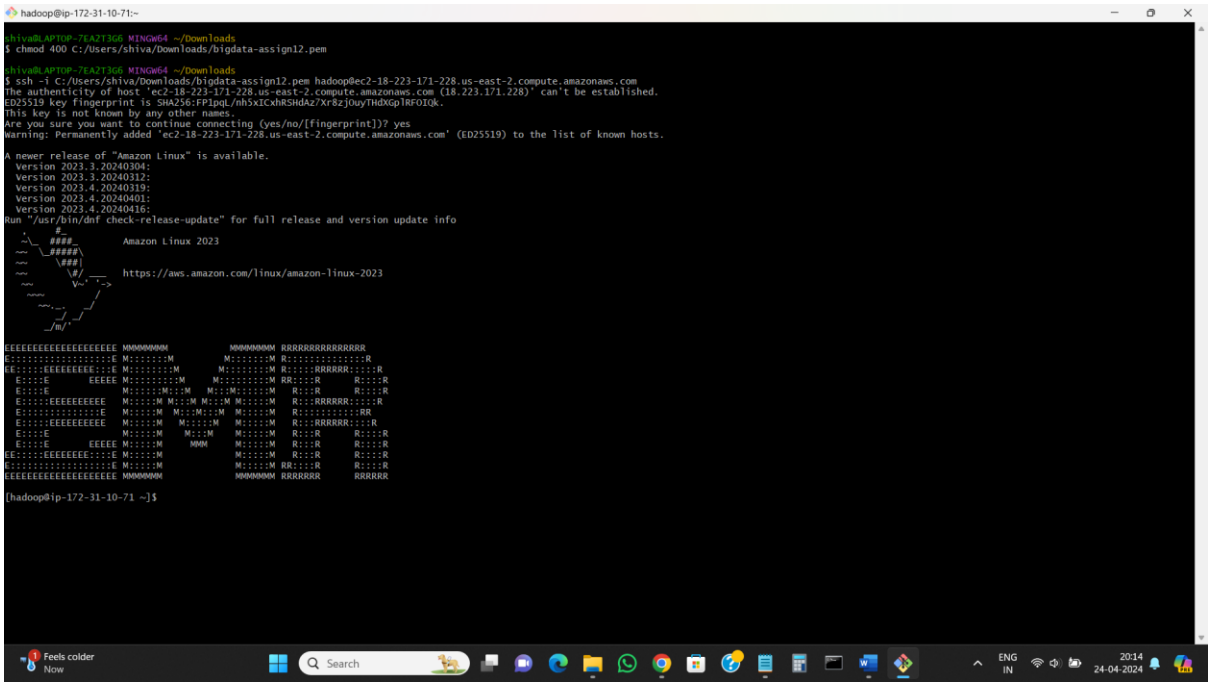
logical and physical data models. They introduced KDM, a potent data modeling tool that automates some of the trickiest, time-consuming, and error-prone data modeling activities, such as conceptual-to-logical mapping, logical-to-physical mapping, and CQL.

Exercise 2) (3 points)

Step A – Start an EMR cluster Start up a EMR cluster as previously, but instead of choosing the “Core Hadoop” configuration chose the “Spark Interactive” configuration (see below), otherwise proceed as before.



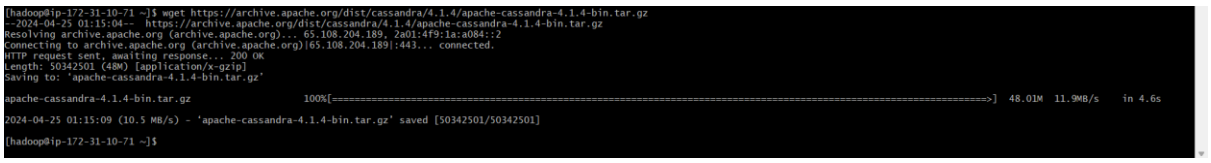
Step B – Install the Cassandra database software and start it Open up a terminal connection to your EMR primary node. Over the course of this exercise, you will need to open up three separate terminal connections to your EMR primary node. This is the first, which we will call Cass-Term.

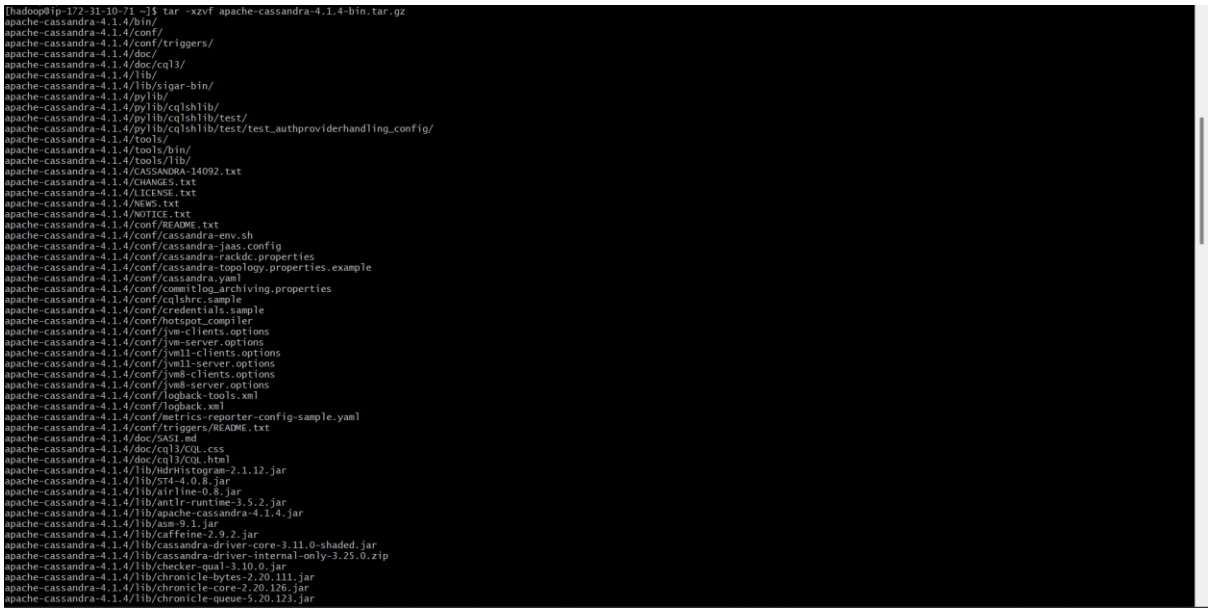


Enter the following two commands:

wget https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz

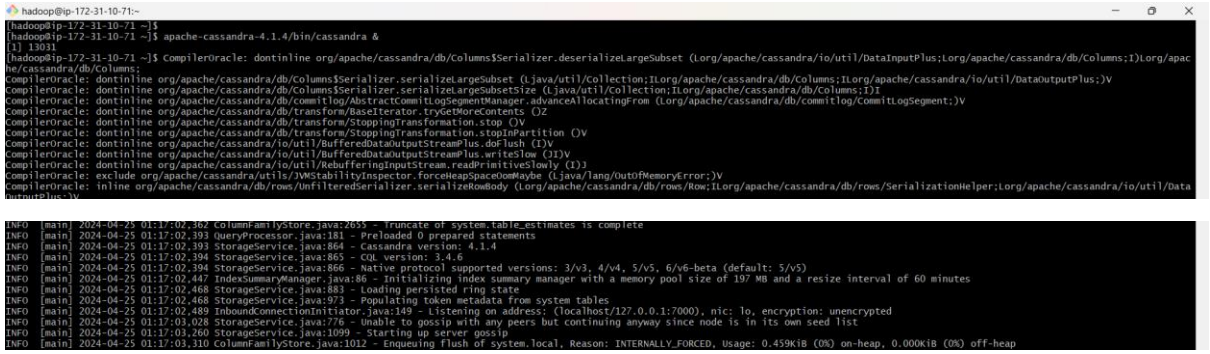
tar -xzvf apache-cassandra-4.1.4-bin.tar.gz





Note, this will create a new directory (apache-cassandra-4.1.4) holding the Cassandra software release. Then enter this command to start Cassandra (lots of diagnostic messages will appear):

apache-cassandra-4.1.4/bin/cassandra &



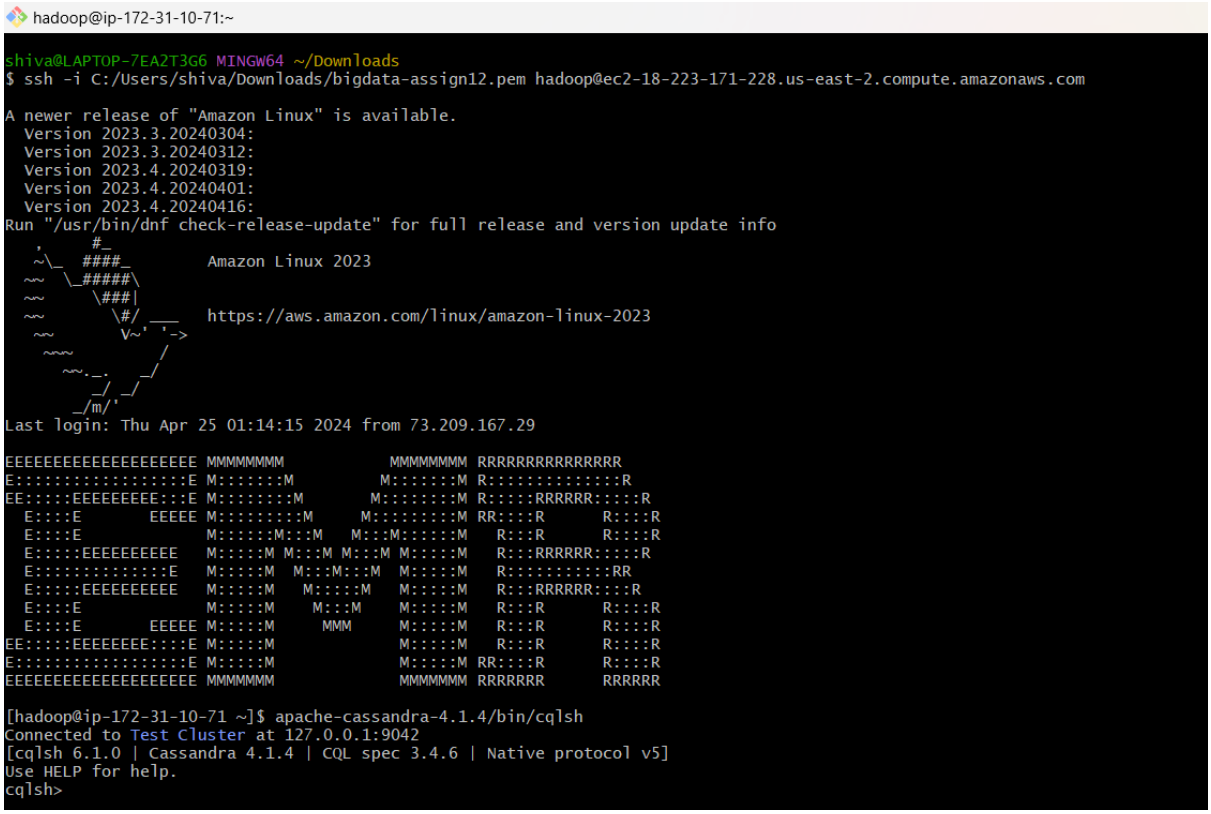
Now wait two or three minutes for Cassandra to start.

Step C – Run the Cassandra interactive command line interface Open a second terminal connection to the EMR primary node. Going forward we will call this terminal connection: Cqlsh-Term.

Note, if you are using the git bash shell on your PC, open a new terminal window by right clicking on the title bar of the program and select ‘New Window’ or enter Alt+F2.

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

apache-cassandra-4.1.4/bin/cqlsh



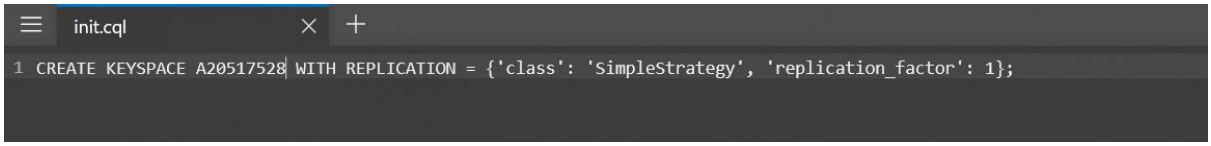
Step D – Prepare to edit your Cassandra code

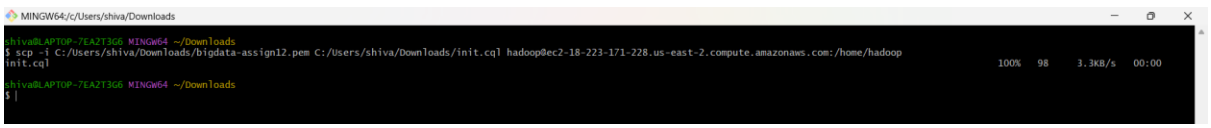
Open a third terminal connection to the EMR primary node. Going forward we will call this terminal connection: Edit-Term.

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.

- Create a file in your working (home) directory on the primary EMR node called **init.cql** using your Edit-term (or using your PC/MAC and then scp it to the EMR primary 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 };

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

source ‘./init.cql’;

```
cqlsh> source './init.cql';
cqlsh>
```

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

describe keyspaces;

```
cqlsh> describe keyspaces;

a20517528  system_auth         system_schema  system_views
system     system_distributed  system_traces  system_virtual_schema

cqlsh>
```

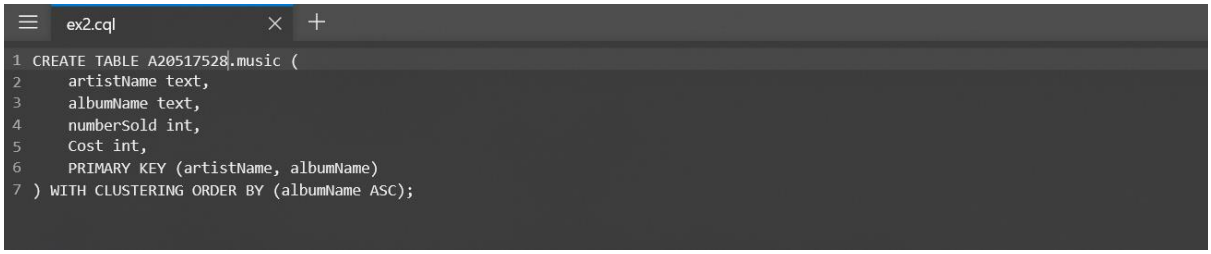
At this point you have created a keyspace unique to you. So, make that keyspace the default by entering the following into the CQL shell:

USE <IIT id>;

For example, USE A1234567;

```
cqlsh> USE A20517528;
cqlsh:a20517528>
```

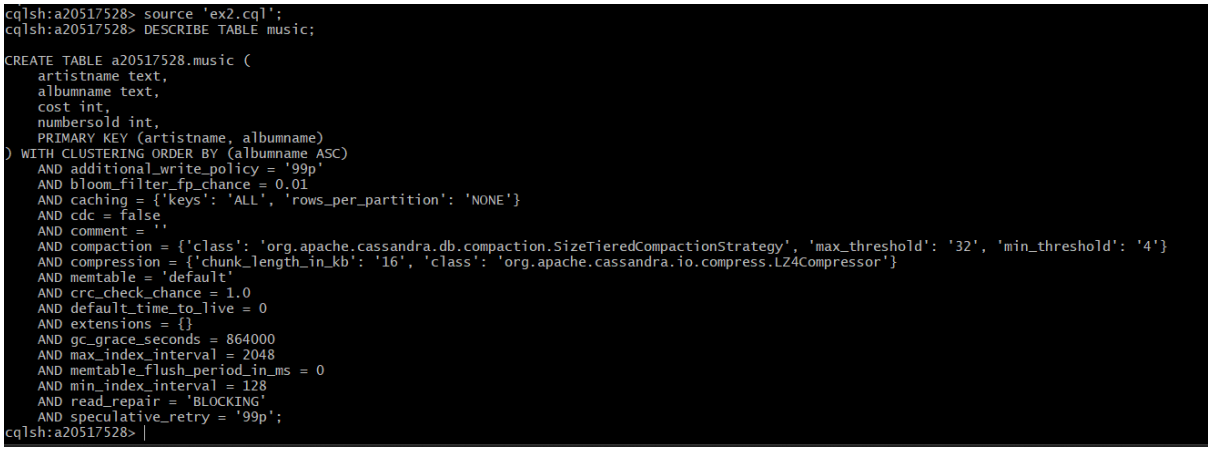
Now create a file in your working directory called ex2.cql using the Edit-Term (or PC/MAC and scp). In this file write the command to create a table named 'Music' with the following characteristics:



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**'.

include (a) the content of the ex2.cql file and (b) a screenshot of the output generated when you then execute 'DESCRIBE TABLE Music' as the result of this exercise.



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'…

```
ex3.cql
1 insert into music (artistName, albumName, numberSold, Cost) values ('Mozart', 'Greatest Hits', 100000, 10);
2 insert into music (artistName, albumName, numberSold, Cost) values ('Taylor Swift', 'Fearless', 2300000, 15);
3 insert into music (artistName, albumName, numberSold, Cost) values ('Black Sabbath', 'Paranoid', 534000, 12);
4 insert into music (artistName, albumName, numberSold, Cost) values ('Katy Perry', 'Prism', 800000, 16);
5 insert into music (artistName, albumName, numberSold, Cost) values ('Katy Perry', 'Teenage Dream', 750000, 14);
```

```
shiva@LAPTOP-7EA2T3G6 MINGW64 ~/Downloads
$ scp -i C:/Users/shiva/Downloads/bigdata-assign12.pem C:/Users/shiva/Downloads/ex3.cql hadoop@ec2-18-223-171-228.us-east-2.compute.amazonaws.com:/home/hadoop
ex3.cql                                                                                    100%  543    21.0KB/s   00:00

shiva@LAPTOP-7EA2T3G6 MINGW64 ~/Downloads
$
```

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

Execute **ex3.cql**.

Provide (a) the content of the ex3.cql file as one result of this exercise, and

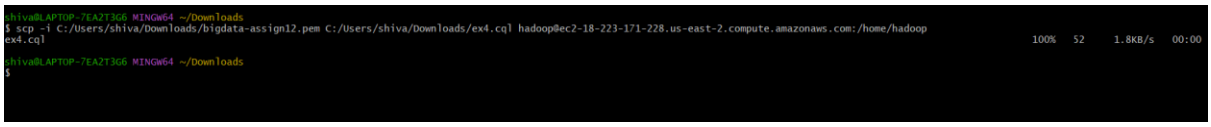
```
cqlsh:a20517528> source 'ex3.cql';
cqlsh:a20517528>
```

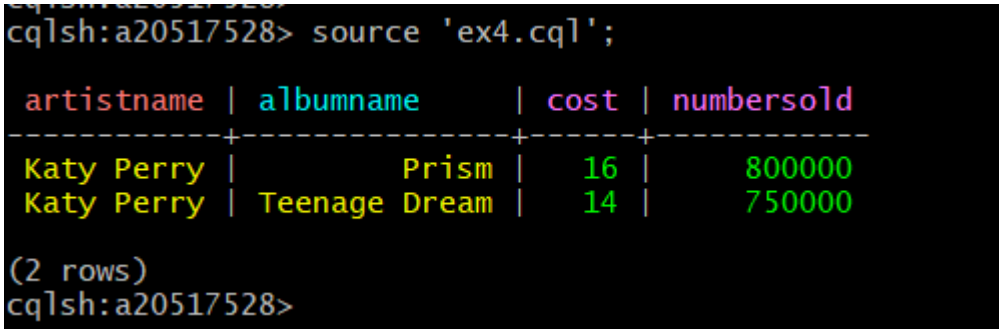
(b) execute the command 'SELECT * FROM Music;' and provide a screenshot of the output of this command as another result of the exercise.

```
cqlsh:a20517528>
cqlsh:a20517528> source 'ex3.cql';
cqlsh:a20517528>
cqlsh:a20517528> 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)
cqlsh:a20517528>
```

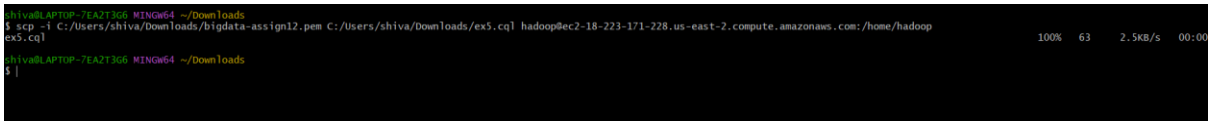
Exercise 4) (2 points)
Now create a file in your working directory called **ex4.cql** using the Edit-Term. In this file write the commands to query and output only Katy Perry songs. Execute **ex4.cql**.

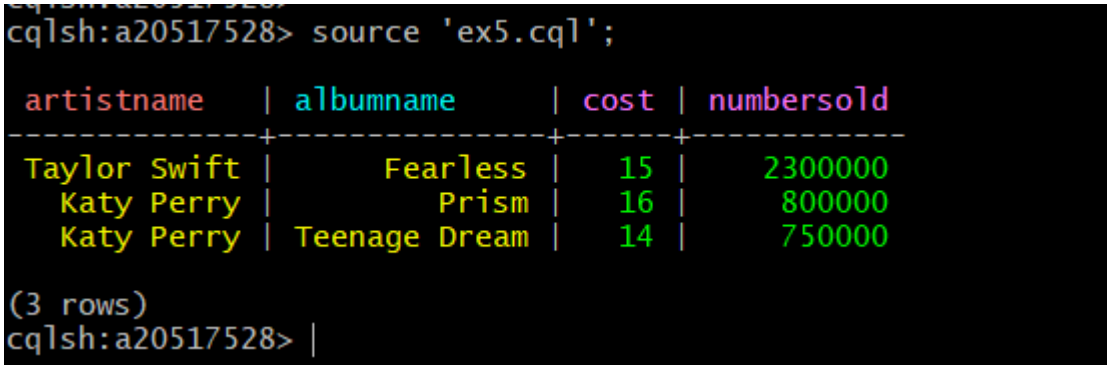
Provide (a) the content of the ex4.cql file and (b) a screenshot of the output of executing this file as the result of this exercise.





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 (a) the content of the ex5.cql file and (b) a screenshot of the output of executing this file as the result of this exercise.





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