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.

In this paper, the author has demonstrated the Data modeling methodology for Apache Cassandra. It is one of the database models which is highly both read and write efficient, leading transactional, scalable, fault-tolerant used to handle the vast amount of data on several clusters across different data centers. Cassandra Query Language (CQL) is the primary query language used to present the queries in the Cassandra model. The proposed Cassandra data model database use keyspace to represent database schema. within keyspace, CQL tables are defined to store and query data. It uses two types of models in Cassandra: The table and the Query model. The table model is a CQL table viewed as a set of partitions that contain rows. It has primary keys which are a combination of partition key (single or composite) and clustering key which is optional. All the rows divided by the partition key are sorted in ascending order of the clustering key. It also has the option of the static attribute, where it is the same value for all the rows in the primary key. The Query model where Queries over tables are expressed in CQL. It supports queries like SQL without the joins and other binary operations for efficiency and scaling. conceptual-to-logical data model mapping is done via data modeling principles, Mapping Rules, and Mapping patterns. Data modeling principles include understanding the data, understanding our query, and ensuring their correct support, Data nesting, and duplicating data across multiple tables. Mapping rules that guide a query-driven transition from a conceptual data model to a logical data model include Entity and relationship types of the map to tables, Equality search attributes map to the prefix columns of a table's primary key, and inequality search attribute maps to a table clustering key column, ordering attributes map to clustering key columns with ascending or descending clustering order, Key attribute types of the map to primary key columns. Mapping Patterns: mapping patterns are designed based on mapping rules that serve as the basis for automating Cassandra database schema design. Physical data modeling is the optimization of a logical data model using optimization techniques like partition splitting, inverted indexes, data aggregation, and concurrent data access optimizations to produce a physical data model. Visualization is done by the Chebotko Diagram, which presents a database schema design as a combination of individual table schemas and query-driven application workflow transitions.

To recapitulate, this paper explains the query-driven data model which uses query-driven schema design, data nesting, and data duplication to model data. Next this model use data modeling principles, mapping rules, and pattern to map conceptual data model to logical data model. Using optimization techniques logical data model is optimized and produces a physical data model. Both logical and physical data models are presented using Chebotko Diagrams. Finally, the paper presented a powerful data modeling tool, called KDM, which automates some of the most complex, error-prone, and time-consuming data modeling tasks, including conceptual-to-logical mapping, logical-to-physical mapping, and CQL generation.

Exercise 2) (3 points)

Step A – Start an EMR cluster

Step B – Install the Cassandra database software and start it
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
```

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

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

Step C – Run the Cassandra interactive command line interface
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:

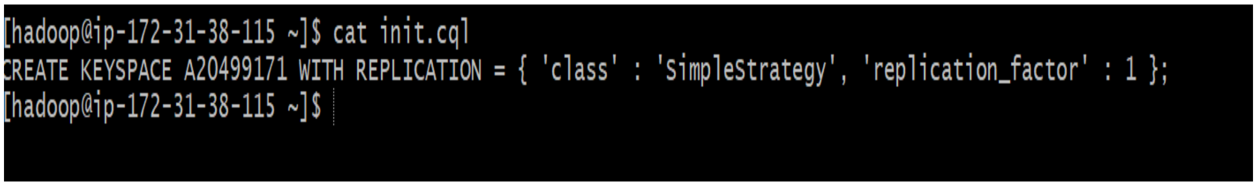
```
apache-cassandra-3.11.2/bin/cqlsh
```

Step D – Prepare to edit your Cassandra code
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.

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:



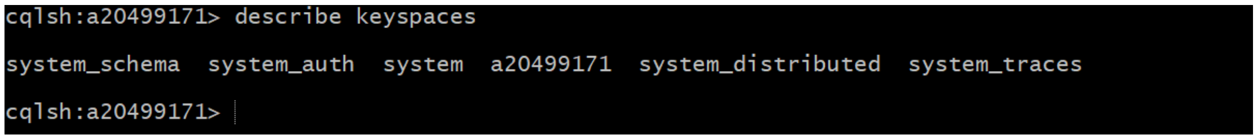
```
[hadoop@ip-172-31-38-115 ~]$ cat init.cql
CREATE KEYSPACE A20499171 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
[hadoop@ip-172-31-38-115 ~]$
```

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

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

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

```
describe keyspaces;
```



```
cqlsh:a20499171> describe keyspaces

system_schema  system_auth  system  a20499171  system_distributed  system_traces

cqlsh:a20499171>
```

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>;
```

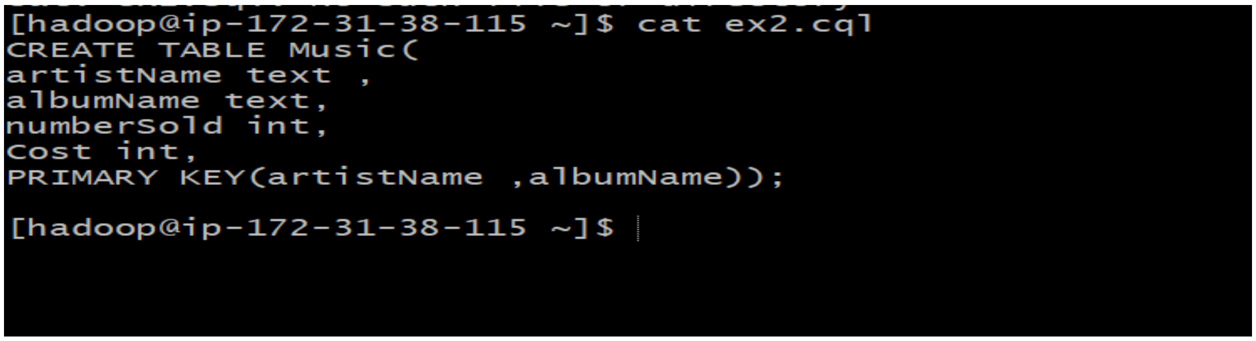
For example,

```
USE A1234567;
```

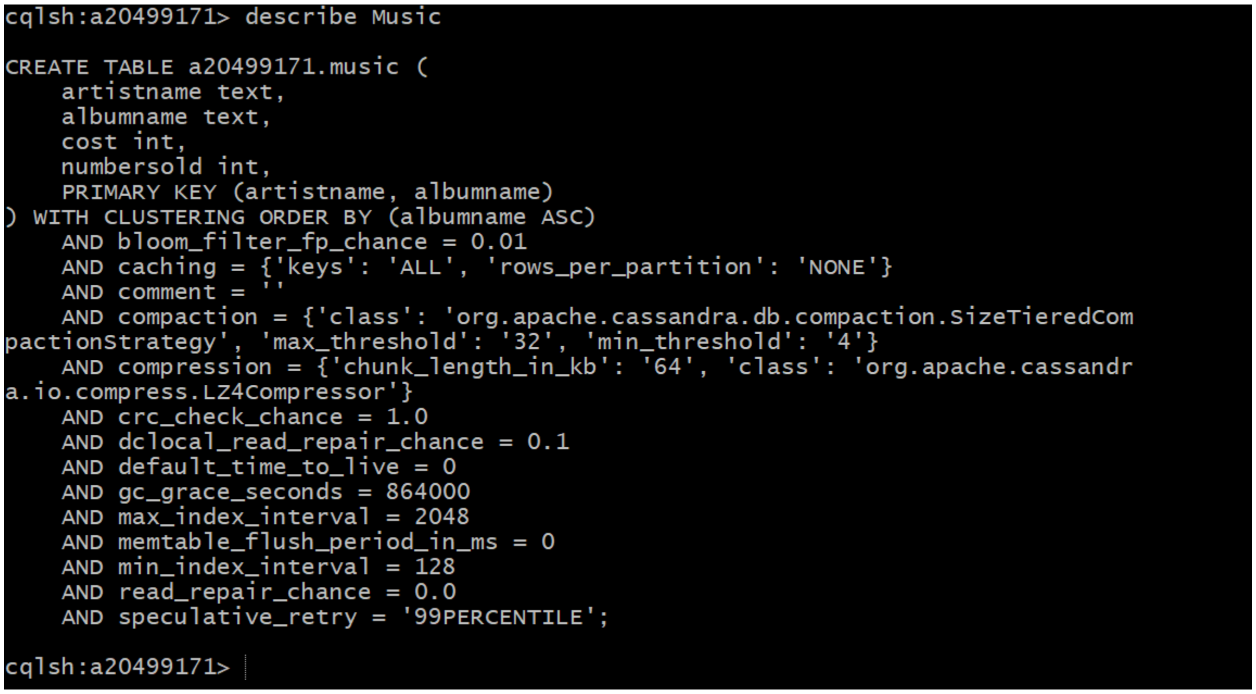
In this file write the command to create a table named 'Music' with the following characteristics:

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.



```
[hadoop@ip-172-31-38-115 ~]$ cat ex2.cql
CREATE TABLE Music(
artistName text ,
albumName text,
numberSold int,
cost int,
PRIMARY KEY(artistName ,albumName));

[hadoop@ip-172-31-38-115 ~]$
```



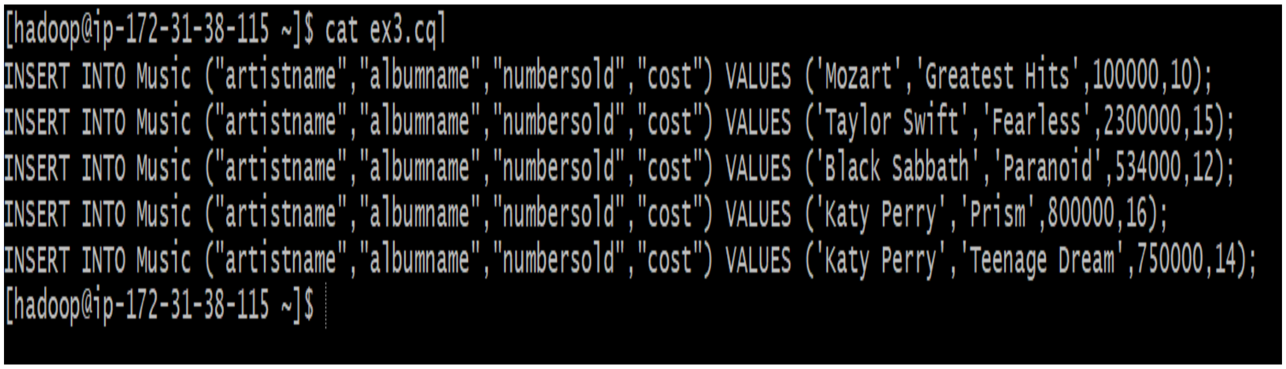
```
cqlsh:a20499171> describe Music

CREATE TABLE a20499171.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.SizeTieredCom
pactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandr
a.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';

cqlsh:a20499171>
```

Exercise 3) (3 points)

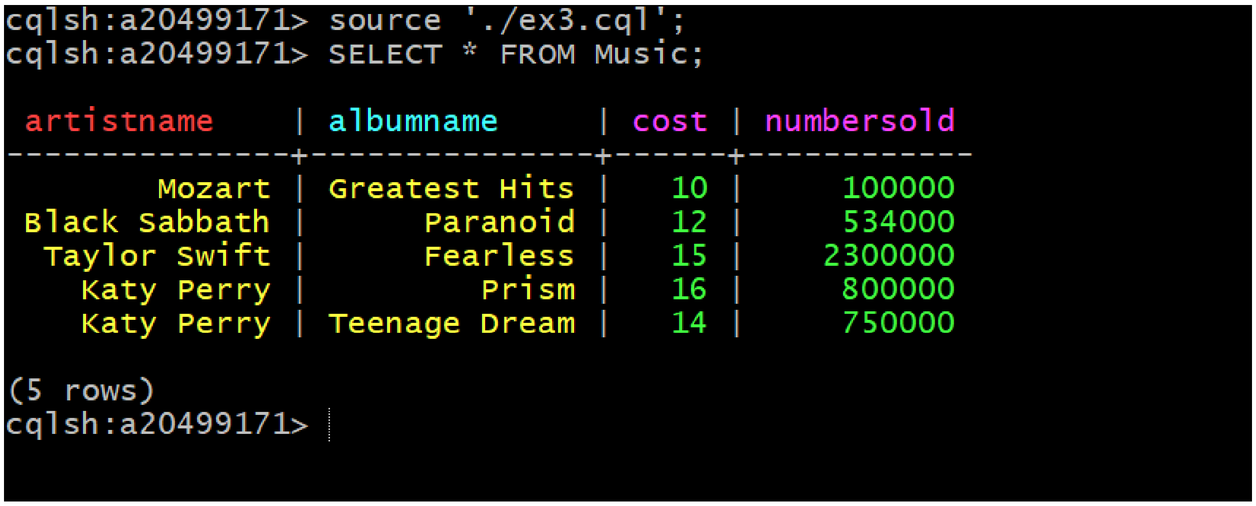
In this file write the commands to insert the following records into table 'Music'…

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

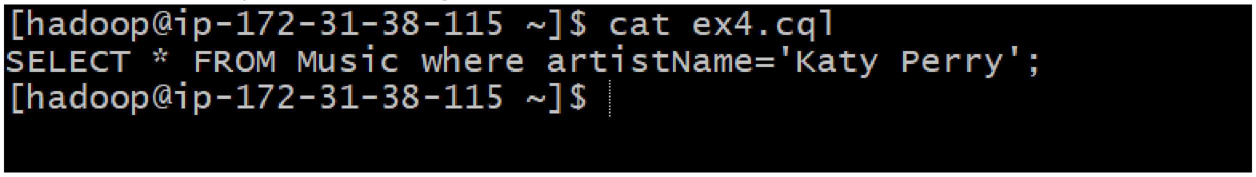
a) Execute ex3.cql. Provide the content of this file as the result of this exercise.

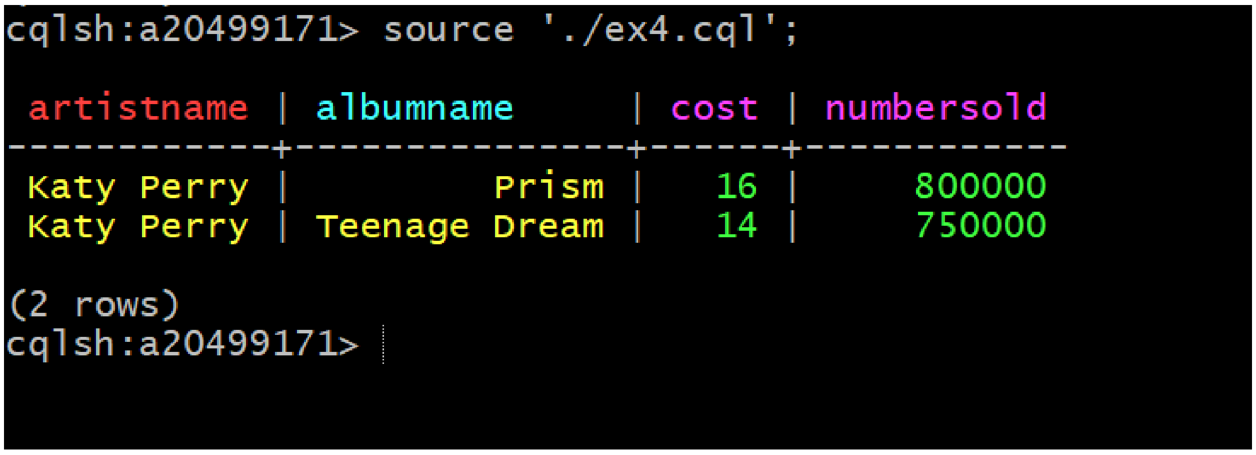

```
[hadoop@ip-172-31-38-115 ~]$ cat ex3.cql
INSERT INTO Music ("artistname","albumname","numbersold","cost") VALUES ('Mozart','Greatest Hits',100000,10);
INSERT INTO Music ("artistname","albumname","numbersold","cost") VALUES ('Taylor Swift','Fearless',2300000,15);
INSERT INTO Music ("artistname","albumname","numbersold","cost") 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);
[hadoop@ip-172-31-38-115 ~]$
```

b) Execute the command ‘SELECT * FROM Music;’ and provide the output of this command as another result of the exercise.


```
cqlsh:a20499171> source './ex3.cql';
cqlsh:a20499171> 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:a20499171>
```

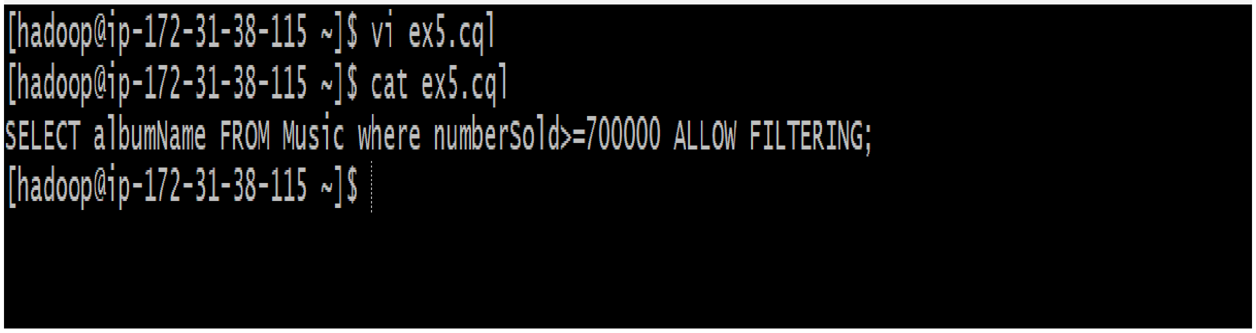
Exercise 4) (2 points)
Now create a file in your working directory called ex4.cql using the Edit-Term. In this file write the commands to query and output only Katy Perry songs. Execute ex4.cql. Provide the content of this file and output of executing this file as the result of this exercise.


```
[hadoop@ip-172-31-38-115 ~]$ cat ex4.cql
SELECT * FROM Music where artistName='Katy Perry';
[hadoop@ip-172-31-38-115 ~]$
```

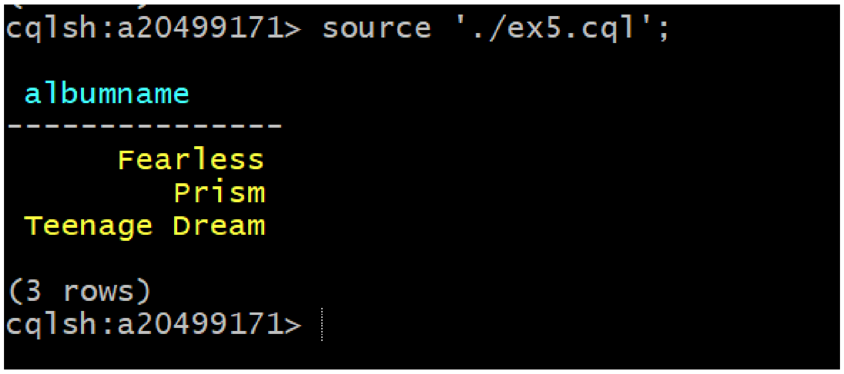


```
cqlsh:a20499171> source './ex4.cql';

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

(2 rows)
cqlsh:a20499171>
```

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-172-31-38-115 ~]$ vi ex5.cql
[hadoop@ip-172-31-38-115 ~]$ cat ex5.cql
SELECT albumName FROM Music where numberSold>=700000 ALLOW FILTERING;
[hadoop@ip-172-31-38-115 ~]$
```



```
cqlsh:a20499171> source './ex5.cql';

 albumname
---------------
      Fearless
         Prism
 Teenage Dream

(3 rows)
cqlsh:a20499171>
```

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