**CSP 554 – Assignment #12**

Name: Adarsh Mathad Vijayakumar

CWID: A20424847

Email ID: avijayakumar@hawk.iit.edu

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.

***A Big Data Modeling Methodology for Apache Cassandra***

Apache Cassandra is a leading transactional, scalable, and highly-available distributed database. This paper proposes the first query driven data modelling methodology for Cassandra. Cassandra data management use cases include product catalogs and playlists, sensor data and Internet of Things, messaging, social networking, recommendation, personalization, fraud detection, and numerous other applications that deal with time series data.

Traditional data modeling is a purely data-driven process, where data access patterns are only taken into account to create additional indexes and occasional materialized views to optimize the most frequently executed queries. In contrast, the Cassandra data model is designed to achieve superior write and read performance for a specified set of queries that an application needs to run. Data modeling for Cassandra starts with application queries.

The most important aspect of the methodology proposed in this paper is that the application workflow and the access patterns become first-class citizens. Cassandra database design revolves around both the application workflow and the data, and both are of paramount importance. The main contributions of the paper are: a first-of-its-kind data modeling methodology; a set of modeling principles, mapping rules, and mapping patterns that guide a logical data modeling process; a visualization technique, called Chebotko Diagrams; a data modeling tool, called KDM, that automates Cassandra database schema design according to the proposed methodology.

The following four data modeling principles provide a foundation for the mapping of conceptual to logical data models: DMP1 (Know Your Data), DMP2 (Know Your Queries), DMP3 (Data Nesting) and DMP4 (Data Duplication).

The five mapping rules are: MR1 (Entities and Relationships), MR2 (Equality Search Attributes), MR3 (Inequality Search Attributes), MR4 (Ordering Attributes) and MR5 (Key Attributes).

Based on the above mapping rules, they design mapping patterns that serve as the basis for automating Cassandra database schema design. 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. The final step of the methodology is the analysis and optimization of a logical data model to produce a physical data model.

The paper also automated the proposed methodology using a Web-based data modeling tool, called KDM2. The tool relies on the mapping patterns and the proprietary algorithms to automate the most complex, error prone and time-consuming data modeling tasks.

This paper introduces us to the paradigm shift in data modelling approach since the evolution of NoSQL databases. Data modelling is a crucial step and needs to be done differently for different database solutions. Automated tools help reduce the effort of data modelling, the right methodology needs to be built into the tool to take benefits offered by the underlying database.

Exercise 2)

At this point you have created a keyspace unique to you. So make that keyspace the default by entering:

USE A20424847;

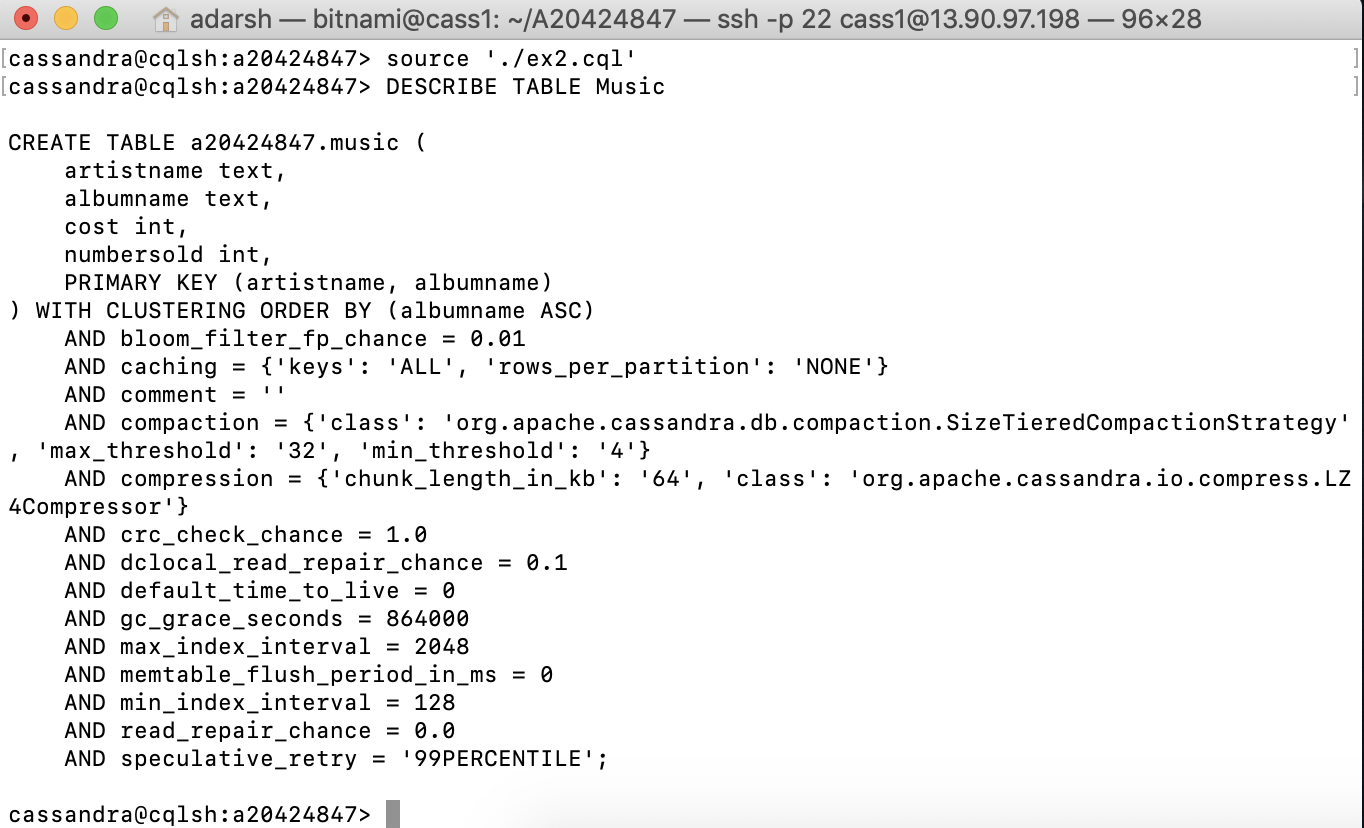
Now create a file in your working directory called ex2.cql. In this file write the command to create a table named ‘Music’.

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

Code written inside the ex2.cql to create the table Music:

CREATE TABLE Music(artistName text, albumName text, numberSold Int, cost Int, PRIMARY KEY(artistName, albumName));

Executing the shell command: DESCRIBE TABLE Music;



Exercise 3)

Now create a file in your working directory called ex3.cql. In this file write the commands to insert the records.

1. Execute ex3.cql. Provide the content of this file as the result of this exercise.
2. Execute the command ‘SELECT \* FROM Music;’ and provide the output of this command as another result of the exercise.

Content of the file ex3.cql:

INSERT INTO Music(artistName, albumName, numberSold, cost) VALUES ('Mozart', 'Greatest Hits', 100000, 10);

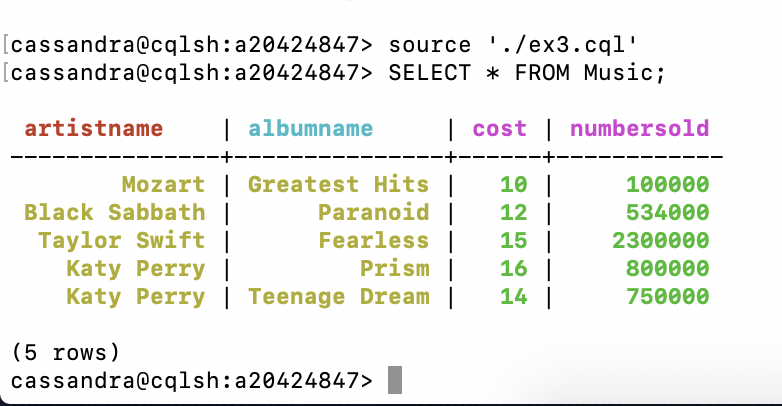
INSERT INTO Music(artistName, albumName, numberSold, cost) VALUES ('Taylor Swift', 'Fearless', 2300000, 15);

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);

Output of the command ‘SELECT \* FROM Music’:



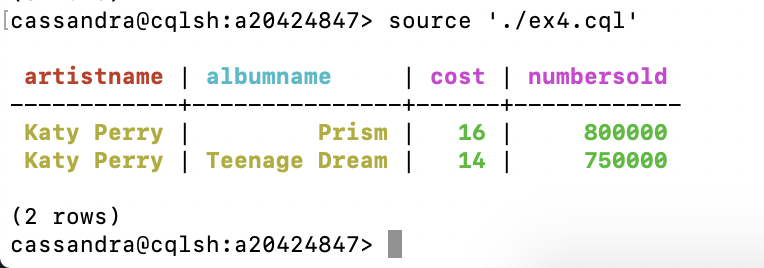
Exercise 4)

Now create a file in your working directory called ex4.cql. In this file write the commands to query only Katy Perry songs. Execute ex4.cql. Provide the content of this file and result of executing this file as the result of this exercise.

Code written inside the ex4.cql:

SELECT \* FROM Music WHERE artistName = 'Katy Perry';

Output:



Exercise 5)

Now create a file in your working directory called ex5.cql. 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 result of executing this file as the result of this exercise.

Code written inside the ex4.cql:

SELECT \* FROM Music WHERE numberSold > 700000 ALLOW FILTERING;

Output:

