CSP554—Big Data Technologies

Assignment #12 – Cassandra

Worth: 12 points

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.

- The stable, query-driven data processing technique of Apache Cassandra is discussed in this article.
- This paper outlines recommendations for mapping and patterning to shift from technology-independent conceptual data models to realistic practical data models for Cassandra, as well as the fundamental concepts of data modeling for Cassandra.
- The methodology is fundamentally different from the traditional relational data processing approach in a number of ways, including query-driven schema design, data replication, and data nesting.
- The process of modeling physical data is also defined, as well as a novel visualization technique called Chebotko Diagrams, which can be used to capture complex conceptual and physical data models.
- Finally, it includes KDM, a powerful data modeling tool that automates some of the most complex, error-prone, and time-consuming data modeling tasks such as logical-to-physical mapping, CQL generation, and conceptual-to-logical mapping.
- Apache Cassandra is a scalable, common transactional, and user-friendly cloud database.
- Cassandra's widespread use in big data applications is attributed to a number of factors, including its fault-tolerant peer-to-peer architecture, flexible, a versatile and compact data model derived from the Big-Table data model, the user-friendly Cassandra Query Language, declarative, and extremely efficient read and write access paths that enable sensitive large data applications to stay online at all times.
- Including thousands of nodes distributed through many data centers, this is believed to be capable of hosting some of the world's largest cluster datasets.
- Cassandra data management use cases include social and messaging networking, suggestion, sensor data and the Internet of Things, playlists and product catalogs, fraud detection, configuration, and a host of other applications involving time-series data.

Exercise 2) (2 points)

a) Follow the instructions "Installing and Starting Cassandra" included on the blackboard with this assignment.

b) You might want to open two terminal windows and connect them (via ssh) to the Cassandra VM you set up in step "a". This will allow you to edit files in one window and start cqlsh in the other to execute them.

c) Create a file in your working directory called init.cql and enter the following commands. 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 };

```
ubuntu@ip-172-31-20-76:~$ touch init.cql
ubuntu@ip-172-31-20-76:~$ vim init.cql
ubuntu@ip-172-31-20-76:~$ cat init.cql
CREATE KEYSPACE A20451170 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
ubuntu@ip-172-31-20-76:~$
```

d) Then execute this file in the CQL shell as follows…

source './init.cql';

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

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

USE <IIT id>;

```
cqlsh> USE A20451170;
cqlsh:a20451170>
```

Now create a file in your working directory called ex2.cql. 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

```
ubuntu@ip-172-31-20-76:~$ touch ex2.cql
ubuntu@ip-172-31-20-76:~$ vim ex2.cql
ubuntu@ip-172-31-20-76:~$ cat ex2.cql
CREATE TABLE A20451170.Music (
        artistName text,
        albumName text,
        numberSold int,
        cost int,
        PRIMARY KEY(artistName, albumName)
) WITH CLUSTERING ORDER BY (albumName DESC);
ubuntu@ip-172-31-20-76:~$
```

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

```
cqlsh:a20451170> source './ex2.cql';
cqlsh:a20451170> DESCRIBE TABLE Music

CREATE TABLE a20451170.music (
    artistname text,
    albumname text,
    cost int,
    numbersold int,
    PRIMARY KEY (artistname, albumname)
) WITH CLUSTERING ORDER BY (albumname DESC)
    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_threshold': '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';

cqlsh:a20451170>
```

Exercise 3) (2 points)

Now create a file in your working directory called ex3.cql. 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.

```
ubuntu@ip-172-31-20-76:~$ touch ex3.cql
ubuntu@ip-172-31-20-76:~$ vim ex3.cql
ubuntu@ip-172-31-20-76:~$ 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);
ubuntu@ip-172-31-20-76:~$
```

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

```
cqlsh:a20451170> source './ex3.cql';
cqlsh:a20451170> SELECT * FROM Music;

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

(5 rows)
cqlsh:a20451170>
```

Exercise 4) (2 points)

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.

```
ubuntu@ip-172-31-20-76:~$ touch ex4.cql
ubuntu@ip-172-31-20-76:~$ vim ex4.cql
ubuntu@ip-172-31-20-76:~$ cat ex4.cql
SELECT * FROM Music WHERE artistName = 'Katy Perry';
ubuntu@ip-172-31-20-76:~$
```

```
cqlsh:a20451170>
cqlsh:a20451170> source './ex4.cql';

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

(2 rows)
cqlsh:a20451170>
```

Exercise 5) (2 points)
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.

```
ubuntu@ip-172-31-20-76:~$ touch ex5.cql
ubuntu@ip-172-31-20-76:~$ vim ex5.cql
ubuntu@ip-172-31-20-76:~$ cat ex5.cql
SELECT * FROM Music WHERE numberSold >= 700000 ALLOW FILTERING;
ubuntu@ip-172-31-20-76:~$
cqlsh:a20451170>
cqlsh:a20451170> source './ex5.cql';

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

(3 rows)
cqlsh:a20451170>
```