

CSP – 554 Big Data Technologies

Assignment #12

Rutul Mehta

A20476293

Exercise 1

Firstly, paper describe the difference between traditional relational data modeling methodology and Cassandra data modeling and explain how traditional approach is not suitable for developing robust Cassandra data model. Then brief description of the Cassandra solution architect is given which encompass the role of designing data and application.

- ➔ Build conceptual data model and defining an application workflow.
- ➔ Workflows describe the access pattern or queries, require run against the database.
- ➔ Based on access patter, mapping between conceptual data model and logical data model can be possible.
- ➔ Logical data model, specify Cassandra tables that can support application queries.
- ➔ Finally, physical data module is produced based on physical optimization concerning data types, keys, partition size and ordering.

Now I summarize following topics,

Cassandra data model:

- ➔ **Table Model:** You can think of a CQL table as a group of partitions containing rows with a similar structure. Each partition in a table uniquely identifies through partition key and row identifies via clustering key and the combination of both is known as primary key. The set of columns and their assigned datatype either primitive (int, text, etc.), complex or counter help to define the table schema.
- ➔ **Query Model:** CQL, which has a SQL-like syntax, is used to express queries over tables. CQL does not support binary operations such as joins and includes a set of query predicates rules that ensure efficiency and scalability.

Conceptual data modelling and application workflow modeling

Understanding of the data and understating of how it is used in data driven application help to design a Cassandra database schema. The ER diagram depicts the former. Application workflow diagrams, which define data access patterns for application tasks, capture the latter.

Logical data modeling:

Based on queries defined in an application workflow, mapping perform between conceptual and logical data model.

- ➔ **Data modeling principle:** Four data modeling principle, DMP1 (know your data), DMP2 (know your queries), DMP3 (data nesting), and DMP4 (data duplication) use to map conceptual to logical data model.
- ➔ **Mapping Rules:** Query driven transition from conceptual data model to logical data model is possible because of five mapping rules which is rely on above data modeling principle. MR1 (Entities and relationship), MR2 (Equality search attribute), MR3 (inequality search attribute), MR4 (ordering attribute), MR5 (Key attribute).
- ➔ **Mapping patterns:** Based on mapping rules, we design mapping patterns that serves basic for automating Cassandra database schema design.

Physical data modeling:

The final step is the analysis and optimization of a logical data model to produce a physical data model.

Exercise – 2

```
hadoop@ip-172-31-47-216:~  
cqlsh> USE a20476293;  
cqlsh:a20476293> source './ex2.clq'  
Could not open './ex2.clq': [Errno 2] No such file or directory: './ex2.clq'  
cqlsh:a20476293> source './ex2.cql'  
cqlsh:a20476293> DESCRIBE TABLE Music  
  
CREATE TABLE a20476293.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_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:a20476293> |
```

Exercise – 3

```
hadoop@ip-172-31-47-216:~  
ess.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:a20476293> source './ex3.cql'  
cqlsh:a20476293> SELECT * FROM Music  
  ... exit  
  ... clear  
  ... exit  
  ... SELECT * FROM Music;  
SyntaxException: line 2:0 no viable alternative at input 'exit' (SELECT * FROM [Music]exit  
...)  
cqlsh:a20476293> 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:a20476293> ;
```

Exercise – 4

```
hadoop@ip-172-31-47-216:~  
    AND read_repair_chance = 0.0  
    AND speculative_retry = '99PERCENTILE';  
  
cqlsh:a20476293> source './ex3.cql'  
cqlsh:a20476293> SELECT * FROM Music  
    ... exit  
    ... clear  
    ... exit  
    ... SELECT * FROM Music;  
SyntaxException: line 2:0 no viable alternative at input 'exit' (SELECT * FROM [Music]exit  
...)  
cqlsh:a20476293> 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:a20476293> source './ex4.cql'  
  
  artistname | albumname | cost | numbersold  
-----  
Katy Perry | Prism | 16 | 800000  
Katy Perry | Teenage Dream | 14 | 750000  
  
(2 rows)  
cqlsh:a20476293> |
```

Exercise – 5

```
hadoop@ip-172-31-47-216:~  
SyntaxException: line 2:0 no viable alternative at input 'exit' (SELECT * FROM [Music]exit  
...)  
cqlsh:a20476293> 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:a20476293> source './ex4.cql'  


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

  
(2 rows)  
cqlsh:a20476293> 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:a20476293> |
```