CSP 554 Big Data Technologies

Assignment #12

Student ID: A20549927

Exercise 1:

Summary of "A Big Data Modeling Methodology for Apache Cassandra"

The document introduces a novel methodology for modeling data in Apache Cassandra, a widely adopted NoSQL database system. Unlike the conventional relational databases that prioritize normalization and data-centric design, this innovative approach places emphasis on understanding the specific queries that an application will execute. By doing so, it aims to address challenges associated with data distribution, partitioning, and replication within the Cassandra environment, leading to a more efficient process from initial business requirements to the implementation of physical data models.

The proposed methodology involves several steps, including identifying key business needs, devising a logical data model, and transforming it into a physical model tailored to Cassandra's architecture. A case study provided in the document illustrates the advantages of this approach, showcasing enhanced performance and scalability for an application handling large volumes of data from various sources. This practical example underscores the effectiveness of adopting a query-driven strategy, which simplifies data modeling while accommodating high scalability demands.

In Cassandra, data modeling requires a departure from traditional SQL practices due to its query language, CQL, which revolves around concepts like keyspaces and tabular structures. While this offers flexibility, it necessitates careful design to prevent data duplication and optimize data retrieval efficiency.

Furthermore, the paper outlines future directions for expanding the methodology to incorporate newer Cassandra features such as user-defined data types and global indexes. Additionally, there is a plan to explore its applicability in analytical contexts. The authors also intend to delve into schema evolution in Cassandra, recognizing its impact on long-term data modeling strategies.

The proposed approach facilitates the alignment of conceptual data models with logical data models, minimizing the reliance on normalization and enabling the creation of more efficient and scalable structures for Cassandra-based applications. These insights hold significant value for database designers and developers seeking to construct resilient, scalable systems leveraging Cassandra's capabilities.

Exercise 2:

Loaded all the contents:
Step A – Start an EMR cluster:

Started an EMR cluster selecting the "Spark Interactive" configuration.

Step B – Install and start Cassandra:

Connected to the primary node of the EMR cluster via terminal (Cass-Term).

Downloaded Cassandra 4.1.4 and extracted it using the provided commands.

Navigated to the Cassandra directory.

Started Cassandra using the command bin/cassandra &.

Waited for Cassandra to start (approximately two to three minutes).

Step C – Run the Cassandra interactive command line interface:

Opened a second terminal connection to the EMR primary node (Cqlsh-Term).

Started the Cassandra CQL shell using the command apache-cassandra-4.1.4/bin/cqlsh.

Step D – Prepare to edit Cassandra code:

Opened a third terminal connection to the EMR primary node (Edit-Term).

Ready to use the 'vi' editor to create or edit Cassandra code files.

With these steps completed, I'm now set up to interact with Cassandra on the EMR cluster.

Connected to Cassandra cqlsh:

```
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
```

Executing init.cql file:

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

a20549927  system_auth         system_schema  system_views
system     system_distributed  system_traces  system_virtual_schema
```

Executing ex2.cql file:

To display the contents of ex2.cql file:

```
CREATE TABLE Music (
artistName text,
albumName text,
numberSold int,
Cost int,
PRIMARY KEY (artistName, albumName));[hadoop@ip-172-
```

To describe Music table:

```
cqlsh:a20549927> DESCRIBE TABLE Music;

CREATE TABLE a20549927.music (
    artistname text,
    albumname text,
    cost int,
    numbersold int,
    PRIMARY KEY (artistname, albumname)
) WITH CLUSTERING ORDER BY (albumname ASC)
    AND additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCom
pactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandr
a.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    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 = 'BLOCKING'
    AND speculative_retry = '99p';
```

Exercise 3:

To display the content of ex3.cql file:

```
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-172-31-20-43 ~]$
```

Output while executing ex3.cql file:

```
cqlsh:a20549927> source './ex3.cql';
cqlsh:a20549927> 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)
```

To display the content of ex4.cql file:

```
Select * from Music where artistName='Katy Perry';
```

Output while executing ex4.cql file:

```
cqlsh:a20549927> source './ex4.cql';

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

(2 rows)
```

To display the content of ex5.cql file:

```
Select * from Music where numberSold >= 700000 ALLOW FILTERING;
```

Output while executing ex5.cql file:

```
cqlsh:a20549927> 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)
```