International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 2

ISSN: 2454-4248
177 – 180

# Cassandra Data Modeling

Prof. Minal P. Patil
Department of Information Technology
MIT COE, Pune
Pune, India
*minal.patil@mitcoe.edu.in*

***Abstract*—**To work with large amount of data consisting of 4 v's velocity, variety, volume and veracity that is nothing but big data, so the need arises to find out the solution to work on such large scale data with high performance. NoSQL databases helps in this scenario. With Cassandra we can efficiently manage the large amount of structured data. It supports dynamic control over the data. In this paper, we present relational and Cassandra data modeling.

***Keywords*-***Cassandra, Data modeling, CQL, Mapping Patterns.*

*****

## I. INTRODUCTION

Over last years, the volume of data has been increased exponentially. Huge amount of data is produced by multiple sources and its representation in different format becomes a challenging task. So the need arises to handle such a big data.
NoSQL is the most popular approach which can deal with such problem. It allows to store and manage large scale datasets which are designed to scale horizontally. According to Brewer's CAP theorem, NoSQL offers consistency in favor of availability and partition tolerance, as a result there is great improvement in performance and scalability as compared to traditional ACID guarantees[2].

- There are different NoSQL databases are available such as HBase, MongoDB, Hypertable, HyperGraphDB, Riak, Memcached, Voldemort, Tokyo Cabinet, Neo4J, Redis, Cassandra and CouchDB.
- But when the question arises for handling the big data and their applications few of above are used as HBase, Hypertable, Riak, Voldemort, Cassandra.
- BigTable is used by HBase and Hypertable. Dynamo is used by Riak and Voldemort. Cassandra is daughter of both BigTable and Dynamo[3].

Depending upon the database properties, NoSQL is divided into four categories as [4],[5]:

### *1. Key Value Database*

Data is stored in decentralized manner. It uses <key, value> pair for storing the values into database. Here, *key* consists in a unique alphanumeric identifier that works like an index, *value* can be simple text strings or complex structures. Key-value stores ideally suited to retrieve information in a very fast, available and scalable way.
For instance, Amazon's Dynamo makes extensive use of a Key-value store system, to manage the products in its shopping cart[6]. Also Linkedin uses Amazon's Dynamo and Voldemort[2]. Dynamo uses Consistent Hashing where each node is assigned to a random position on the ring, node is chosen by walking clockwise from the hash location.

### *2. Document Oriented Database*

It is used for storing, retrieving and managing semi-structured data. A document contains a description of data type and value for it. Documents are grouped to form collection which functions same as relational table.
For instance, MongoDB and CouchDB are two popular document oriented databases[2].

### *3. Column Family Database*

Data is stored in centralized manner and ordered by a row key. It uses multidimensional sorted map as map is indexed by a row key and a column key, and ordered by a row key. Columns are grouped into sets called column families.
For instance, Google's BigTableis is one of the most popular column family database[7].

### *4. Graph Database*

In graph database data is represented as a network of nodes[2]. A graph database works with nodes and relationships along with properties. It contains connected entities as nodes. Relationship provides directed connection among two node entities. Every relationship has a start node and end node.
For instance, Neo4j and Allegro are most popular examples of graph database[2], where Neo4j is an open-source NoSQL graph database implemented in java and scala.

## II. DATA MODELING CONCEPT

Data modeling is a science, which applies tested methodologies, make improvements and reproducible. In relational databases, traditional data modeling methodology is used [8], [9], [10]. This process is purely data driven process. As depicted in following Figure 1, flow starts from conceptual data modeling, mapping it into relational data model and at last get relational database schema. In this process first preference is given to organize the data into relation by understanding and querying it and also take care of data redundancy and data duplication. SQL supports normalization, joins, data aggregation, nested queries.

 ISSN: 2454-4248
177 – 180

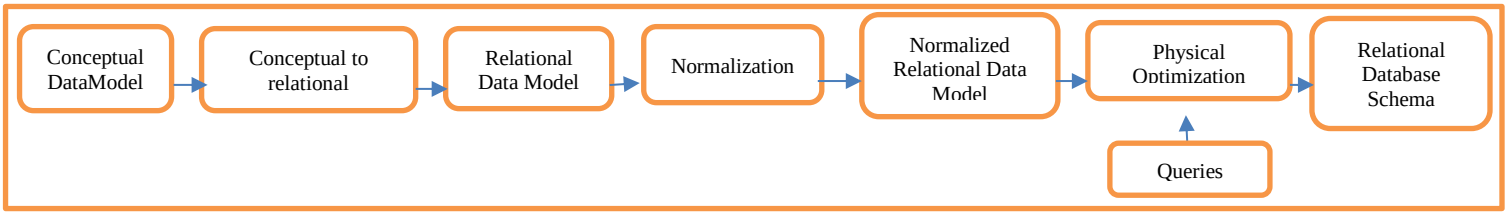



Figure 1: Relational Data Modeling [1]

Data modeling in Cassandra differs from traditional data modeling. It is based on conceptual data model alone. CQL doesn't support data aggregation and joins. As single table is able to answer all queries, CQL supports denormalization. As depicted in following figure[1], flow starts from conceptual data model along with application workflow mapping it into logical data model and at last get physical data model. In this process first preference is given to conceptual data only excluding queries [1].

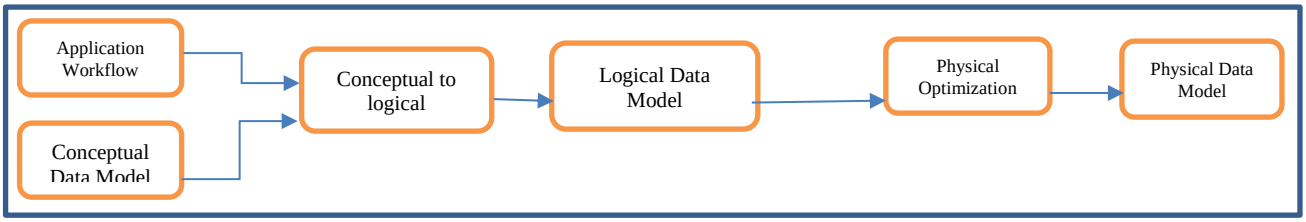



Figure 2. Cassandra Data Modeling [1]

***2.1 Application Workflow:***

User queries are defined in application workflow. In Cassandra database is termed as keyspace and contains collection of different tables. To work on such tables the queries are expresses in CQL ( Cassandra Query Language) having SQL like syntax.

The syntax for creating a keyspace is as follo[ws:

```
CREATE KEYSPACE keyspace_name WITH
replication = {'class': "Replica Placement
Strategy",'replecation factor': "No. of machines""};
```

- Where class represents strategy used to place replicas in ring. These strategies are simple strategy, old network topology strategy and network topology strategy.
- Each keyspace contains column family, each column family contains ordered rows and each row contains ordered collection of columns.

A Cassandra column family has following attributes

- Keys_cached – no of locations to keep cached per SSTable
- Rows_cached – no of rows whose contents will be cached in memory
- Preload_row_cache – used to pre-populates row cache

To verify whether table has been created or not, we have to use DESCRIBE.

ALTER KEYSPACE is used to alter the properties, the syntax for that is as

```
ALTER KEYSPACE "keyspace_name" WITH
replication = {'class':'Strategy name',
'replication_factor': ' no. of replicas'};
```

DROP KEYSPACE is used to delete a keyspace, the syntax for that is as

```
DROP KEYSPACE "keyspace_name"
```

TABLE operations:

```
CREATE TABLE table_name( column1_name datatype
<OPTION as PRIMARYKEY>, column2_name datatype,
column3_namedatatype,PRIMARY KEY(column1_name))
```

To verify we can use SELECT, as

```
SELECT * from table_name;
```

```
ALTER TABLE table_name ADD
newcolumn  namedatatype;
```

ALTER TABLE table_name DROP column_name;

```
DROP TABLE table  name;
```

To delete table permanently use TRUNCATE command, as

```
TRUNCATE table_name
```

 ISSN: 2454-4248
177 – 180

**CRUD Operations :**
**Creating Data in table(insert):**

```
INSERT INTO table_name( column1_name, column2_name, …) values(vlaue1, value2,…..)USING option;
```

**Reading Data from table(select):**

```
SELECT column_names FROM table_name;
```

**Updating Data in table(update):**

```
UPDATE table_name SET column_name = new value WHERE condition:
```

**Deleting Data from table(delete):**

```
DELETE column_name FROM table_name WHERE condition;
```

### *2.2 Conceptual Data Model:*

Conceptual data model gives E-R Diagram representation to understand the relationship between different entities with respect to attributes, cardinalities and constraints. It uses a top down approach which can be algorithmically defined. Conceptual data modeling mean not only understanding of to be managed data but also understanding of the ways data driven applications accesses them [1].

### *2.2 Conceptual to logical mapping:*

The input is available in the form of conceptual data model and map it into logical data model with the help of application workflow and queries in it. It can be done with data modeling principles, Mapping Rules and mapping patterns [1].

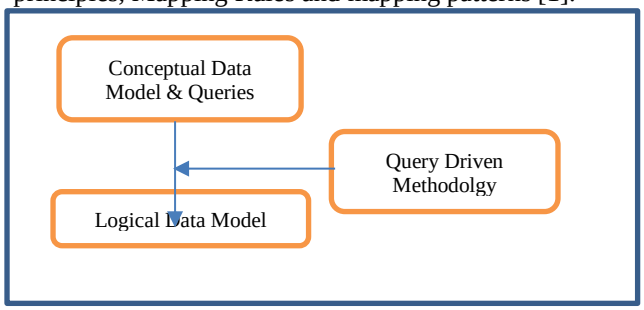
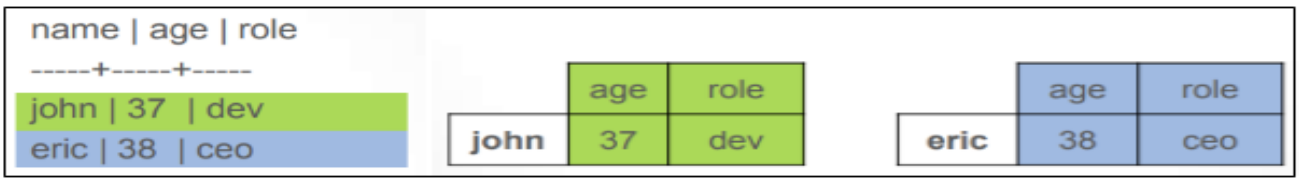



Figure 3. Conceptual to logical mapping

Suppose we have the query given as

**CREATE TABLE employees (name text PRIMARY KEY, age int, role text);**
Then the mapping will results into this [11],




### *2.3Logical Data Model:*

***2.3.1 Data modeling Principles [1]:*** There are four data modeling principles are used for mapping conceptual to logical modeling.

1. Know your data : To organize data correctly, entities, attributes and their relationships must be well known to develop a conceptual data model.
2. Know your queries: To organize data efficiently, queries are used. The best option to be performed is partition per query.
3. Data nesting : To organize multiple entities of same type together on known criterion, data nesting is used. It is used to retrieve multiple entities from a single partition.
4. Data Duplication : It is always better to have data duplication over joins in Cassandra as it helps efficiently supports different queries over the same data.

***2.3.2 Mapping Rules [1]:*** There are five mapping rule, which are used along with data modeling principles.

1. Entities and Relationships
2. Equality search attributes
3. Inequality search attributes
4. Ordering Attributes
5. Key Attributes

***2.3.3 Mapping Patterns:*** Mapping patterns gives the final schema design, so it serves as the basis for automating Cassandra database schema design [1].
Three important points to be consider for Cassandra [11]:

1. Cassandra finds rows fast
2. Cassandra scans column fast
3. Cassandra does not scan rows.

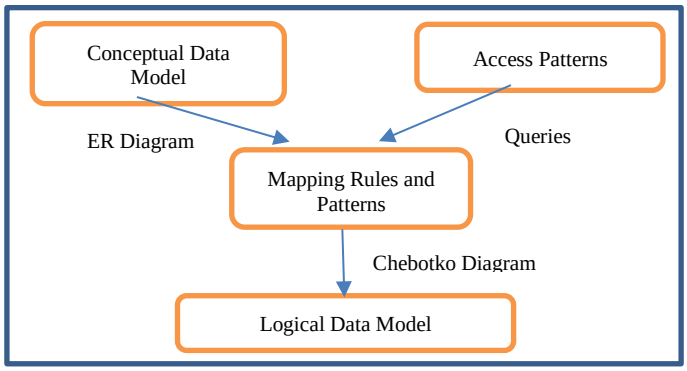
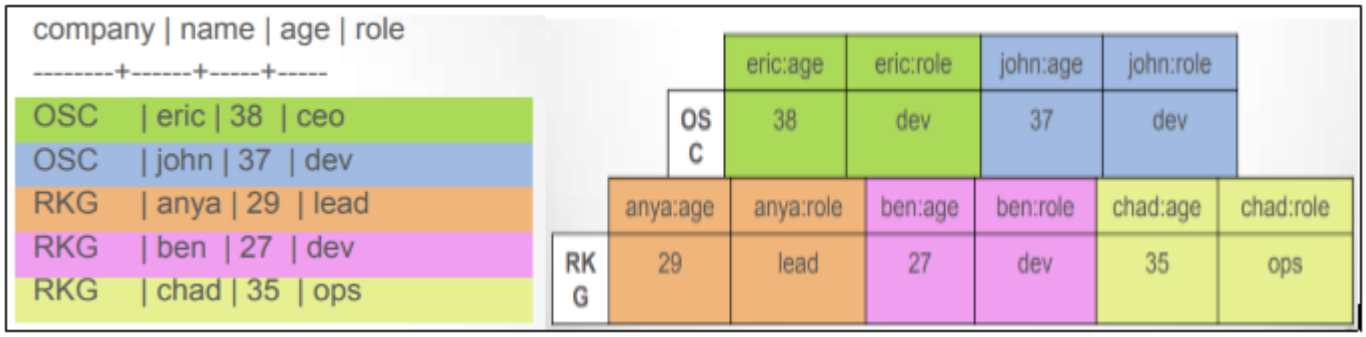



Figure 4. Conceptual to logical data model

 ISSN: 2454-4248
177 – 180

Suppose we have the query given as [11],

**CREATE TABLE employees ( company text, name text, age int, role text, PRIMARY KEY (company,name) );**




### 2.4 Physical Data Model:

This model tends to be platform specific as it reflects database schema and platform specific aspects. With the help of conceptual data model we can perceive higher level of abstraction with data entities and their relationships. With the help of logical data model we have to focus on detailing the entities instead of thinking about the implementation details. With the help of physical data model, we can represent data in the DBMS format [12], [13].

## III. ADVANTAGES OF CASSANDRA

- Minimal Administration
- No Single Point of Failure
- Scale Horizontally
- Writes are durable
- Optimized for writes
- Consistency is flexible, can be updated online
- Schema is flexible, can be updated online
- Handles failure gracefully
- Replication is easy, Rack and DC aware

## IV. CONCLUSION

When there is a question on volume and variety of data, to improve performance and scalability, we need to reinvent the ways in which data is represented and analyzed so data can be extracted efficiently. The NoSQL platform, Cassandra high scalability and fulfills this ideal requirement of massive amount of data storage. It promotes fast retrieval of data irrespective of the size of stored data. In this paper we explained the functioning of Cassandra data modeling over Relational Data Modeling. Cassandra data modeling elaborates all its functionality with the help of modules says, Application Workflow, Conceptual data model, Mapping of conceptual to logical, physical optimization and finally physical data model.

## REFERENCES

[1] ArtemChebotko, AndreyKashlev, Shiyong Lu, " A Big Data Modeling Methodology for Apache Cassandra", 2015 IEEE International Congress on Big Data, 2015.

[2] Andre Ribeiro, Afonso Silva, Alberto Rodrigues da Silva, " Data Modeling and Data Analytics: A Survey from a Big Data Perspective", Journal of Software Engineering and Applications, 2015, 8, 617-634.

[3] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ...& Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 4.Pramod Sadalage, NoSQL Databases: An Overview, thoughtworks.com, October 2014.

[4] Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In Pervasive computing and applications (ICPCA), 2011 6th international conference on (pp. 363-366). IEEE.

[5] Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12-27.

[6] Moniruzzaman, A.B.M. and Hossain, S.A. (2013) NoSQL Database: New Era of Databases for Big data Analytics-Classification, Characteristics and Comparison. *International Journal of Database Theory and Application*, **6**, 1-14.

[7] Chang, F., *et al.* (2006) Bigtable: A Distributed Storage System for Structured Data. *Proceedings of the 7th Symposiumon Operating Systems Design and Implementation* (*OSDI*' 06), Seattle, 6-8 November 2006, 205-218.

[8] E. F. Codd "A relational model of data for large shared data banks",Commun. ACM, vol. 13 no. 6 pp. 377-387 1970.

[9] E. F. Codd "Further normalization of the data base relational model", IBM Research Report, vol. RJ909 1971.

[10] P. P. Chen "The entity-relationship model - toward a unified view of data", ACM Trans. Database Syst., vol. 1 no. 1 pp. 9-36 1976.

[11] John Berryman, "The CQL3/Cassandra Mapping", OpenSource Connections.

[12] Ramakrishnan, R. and Gehrke, J. (2012) Database Management Systems. 3rd Edition, McGraw-Hill, Inc., New York.

[13] Connolly, T.M. and Begg, C.E. (2005) Database Systems: A Practical Approach to Design, Implementation, and Management. 4th Edition, Pearson Education, Harlow.