

# CASSANDRA - DATA MODEL

The data model of Cassandra is significantly different from what we normally see in an RDBMS. This chapter provides an overview of how Cassandra stores its data.

## Cluster

Cassandra database is distributed over several machines that operate together. The outermost container is known as the Cluster. For failure handling, every node contains a replica, and in case of a failure, the replica takes charge. Cassandra arranges the nodes in a cluster, in a ring format, and assigns data to them.

## Keyspace

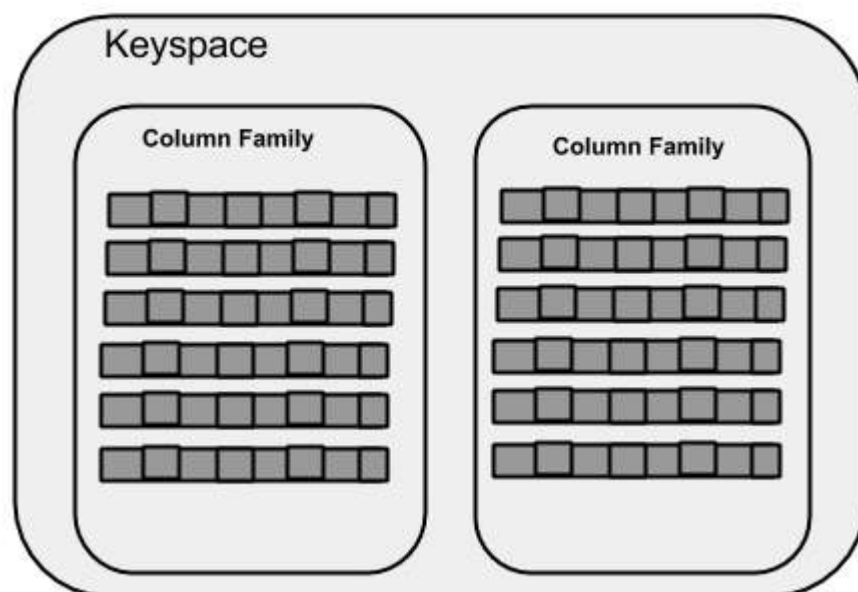
Keyspace is the outermost container for data in Cassandra. The basic attributes of a Keyspace in Cassandra are –

- **Replication factor** – It is the number of machines in the cluster that will receive copies of the same data.
- **Replica placement strategy** – It is nothing but the strategy to place replicas in the ring. We have strategies such as **simple strategy** *rack – awarestrategy*, **old network topology strategy** *rack – awarestrategy*, and **network topology strategy** *datacenter – sharedstrategy*.
- **Column families** – Keyspace is a container for a list of one or more column families. A column family, in turn, is a container of a collection of rows. Each row contains ordered columns. Column families represent the structure of your data. Each keyspace has at least one and often many column families.

The syntax of creating a Keyspace is as follows –

```
CREATE KEYSPACE Keyspace name  
WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 3};
```

The following illustration shows a schematic view of a Keyspace.



## Column Family

A column family is a container for an ordered collection of rows. Each row, in turn, is an ordered collection of columns. The following table lists the points that differentiate a column family from a table of relational databases.

## Relational Table

A schema in a relational model is fixed. Once we define certain columns for a table, while inserting data, in every row all the columns must be filled at least with a null value.

Relational tables define only columns and the user fills in the table with values.

## Cassandra column Family

In Cassandra, although the column families are defined, the columns are not. You can freely add any column to any column family at any time.

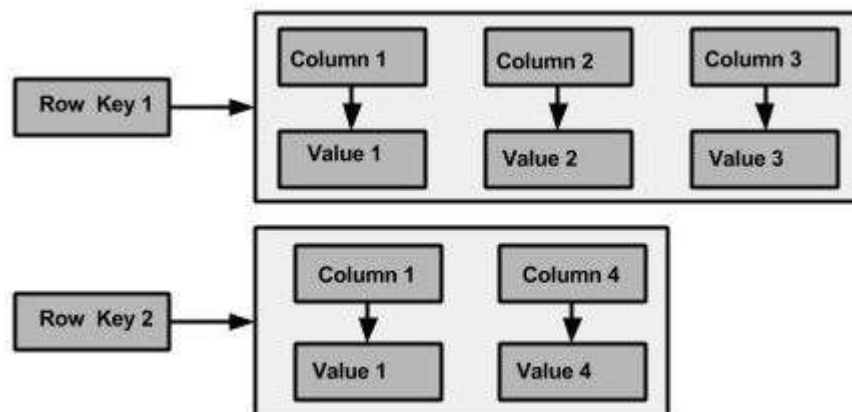
In Cassandra, a table contains columns, or can be defined as a super column family.

A Cassandra column family has the following attributes –

- **keys\_cached** – It represents the number of locations to keep cached per SSTable.
- **rows\_cached** – It represents the number of rows whose entire contents will be cached in memory.
- **preload\_row\_cache** – It specifies whether you want to pre-populate the row cache.

**Note** – Unlike relational tables where a column family's schema is not fixed, Cassandra does not force individual rows to have all the columns.

The following figure shows an example of a Cassandra column family.



## Column

A column is the basic data structure of Cassandra with three values, namely key or column name, value, and a time stamp. Given below is the structure of a column.

Column		
name : byte[]	value : byte[]	clock : clock[]

## SuperColumn

A super column is a special column, therefore, it is also a key-value pair. But a super column stores a map of sub-columns.

Generally column families are stored on disk in individual files. Therefore, to optimize performance, it is important to keep columns that you are likely to query together in the same column family, and a super column can be helpful here. Given below is the structure of a super column.

Super Column	
name : byte[]	cols : map<byte[],

## Data Models of Cassandra and RDBMS

The following table lists down the points that differentiate the data model of Cassandra from that of an RDBMS.

RDBMS	Cassandra
RDBMS deals with structured data.	Cassandra deals with unstructured data.
It has a fixed schema.	Cassandra has a flexible schema.
In RDBMS, a table is an array of arrays. <i>ROW</i> × <i>COLUMN</i>	In Cassandra, a table is a list of “nested key-value pairs”. <i>ROW</i> × <i>COLUMN</i> <i>key</i> × <i>COLUMN</i> <i>value</i>
Database is the outermost container that contains data corresponding to an application.	Keyspace is the outermost container that contains data corresponding to an application.
Tables are the entities of a database.	Tables or column families are the entity of a keyspace.
Row is an individual record in RDBMS.	Row is a unit of replication in Cassandra.
Column represents the attributes of a relation.	Column is a unit of storage in Cassandra.
RDBMS supports the concepts of foreign keys, joins.	Relationships are represented using collections.