NOSQL-COLUMN Family Model

In this article I will discuss about Column Family Data Model. This Column Family Data model use in NoSQL database. Cassandra follows this Column Family Data Model.

What Is Column Family Data Model?

In order to understand Column Family, I will take a small example and try to map it on Relational database then gradually move on to No SQL mode.

Suppose I want to design Employee and Department relationship, so in a RDBMS

We will create Two Tables One for Employee another for Department.

Employee table has dept\_id column which stored Department ID so Department ID is act as Foreign Key

**Table: Employee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Emp\_id | Emp\_name | Emp\_address | Emp\_sex | Dept\_id |
| 1 | Shamik mitra | 1,N D Lane | M | 2 |
| 2 | Ajay Bose | 34,CH Street | M | 1 |
| 3. | Ragini Sil | 45, CT Road | F | 2 |
| 4 | Aniket Dhar | 2,PL Road | M | 2 |

**Table: Department:**

|  |  |  |
| --- | --- | --- |
| Dept\_id | Dept\_name | Dept\_detail |
| 1 | HR | HR Function |
| 2 | IT | Developement |

**The Problem domain, map in to Two tables now suppose we will search for**

**“*Details of Employees Under a Department.”***

One should join Employee and Department using Dept\_id and fetch the necessary information.

**Query should be**

***Select \* Employee e, Department d where e.Dept\_id=d.Dept\_id order by e.name;***

***For a small number of Dataset this will return result very fast but when data increases gradually performance will degrade. In case of 2 million record that will take a good amount of time.***

***Why it takes time?***

To understand this, we need to know How Data are store in RDBMS.

In RDBMS Row wise datas are save in sequential order but individual Rows are not save in sequential they are distributed over disk space***.*** So in above case Shamik Mitra and it’s all related columns are store sequentially say from location

00022 to 00026 in disc.

But for Ajay Bose may it will store in 00134 to 00138.

So now think to find Employees in same department, RDBMS has to hop over here and there in the disc space to collect Employee data who are in same department so obviously it takes time.

Apart from this another bog problem in RDBMS is it has pre-defined Schema so anything Outside the schema would not fit. Like I can say if I want to save hobby for an Employee

I need to change Table Structure of Employee to fit this requirement.

Addressing these problem, NOSQL comes to play

NOSQL Characteristics are

1. It should be Schema less.
2. Data should be stored in distributed manner.
3. Most important It stores Data Aggregation in other way it stores the whole relationship.

mainly there are 4 types Data model in NOSQL data bases

1. Key-value pair
2. Document Base.
3. Column Family.
4. Graph Database

**We will talk about Column Family**.

In Column family style, data store based on column so you can think as

Multiple columns together make a Column Family. In One glance it may be look like same as RDBMS but that is not the case.

Now you can think Table is like Column family. But the main difference is it is Schema less and here Columns are stored sequentially, unlike RDBMS where Rows are store sequentially. As it is schema less we can add any column relate with that column family.

So Here all **name** Column, **dept\_id** column, **sex** column are stores sequentially but

Each column in a row, stores in different location .

So according to definition, All Employee name are stores sequentially in a disc and all dept\_id store sequentially but for a single row name and dept\_id is not sequential.

But one key point should be remembered, each **Row has One Unique Key for a Column family. Key can be same for different Column family**.

By this Unique key we can Identify an employee in Employee column family.

Column Family Model has Three main elements

1. Column Family: Column Family is a single structure that can group Columns and SuperColumns with ease. Thinks as table in RDBMS.
2. Column :  It has an ordered list of elements or tuple with a name and a value defined.
3. Key : Unique Identifier of the record. Keys have different numbers of columns, so the database can scale in an irregular way as it is Schema less.
4. Keyspace:  This defines the outermost level of an organization, typically the name of the application. Think as database schema in RDBMS.
5. Super Column : Super column are store mapping between Keys of different column family.

Let’s take a look How we can map Employee & Department relation in Column Family

**Data Structure :**

**Employee Column Family**

|  |  |
| --- | --- |
| Key-1 |  |
|  | Name : shamik Mitra |
|  | Adress : Nivedita lane |
|  |  |
| Key- 2 |  |
|  | Name : Ajay Bose |
|  | Address : 34 CT Road |
|  | Hobby : Tennis |

**Department Column family**

|  |  |
| --- | --- |
| DEPT\_ID 1 |  |
|  | Name : HR |
|  | Details : HR Function |
| DEPT\_ID 2 |  |
|  | Name : IT |
|  | Details : Developement |

**Mapping of Employee and Department (Super column)**

|  |  |
| --- | --- |
| DEP\_ID 1 |  |
|  | KEY-2 |
|  | KEY…N |
| DEPT\_ID-2 |  |
|  | KEY-1 |

Now think about the query again search Employee under a department.

As Dept\_id column is sequential to search Dept\_id with in large data set is not a problem because no require to hop here and there. Second thing we need to fetch employees under department

But here we need help of Super column as in No SQL data base there is no concept of foreign key or nor we can’t search NOSQL data base by any attribute only we search it through key so to find the Key of Employee we need to find out Employees Key

So we need help from super column and find Employees key then find Employee details