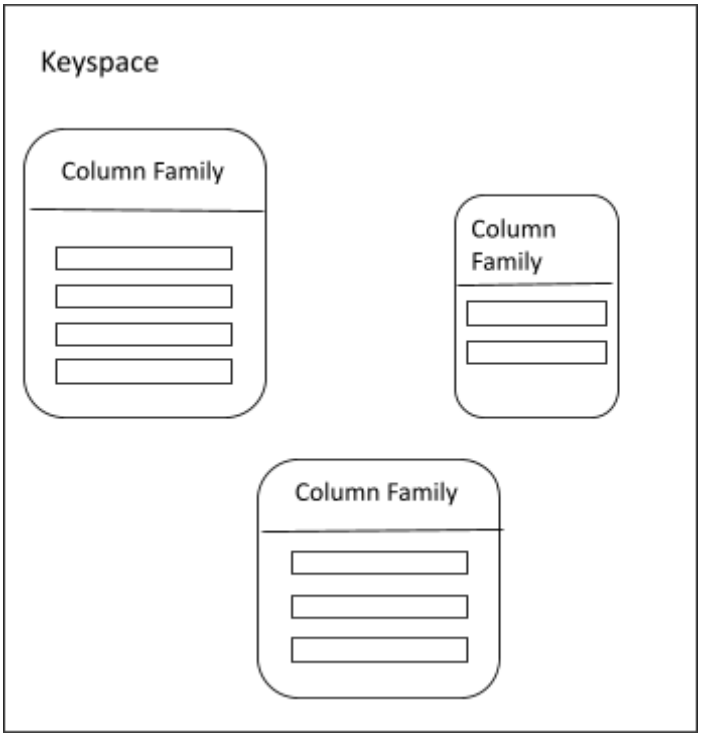
Columnar NoSQL Database:

This kind of database stores data in columns instead of rows. It speeds up the read and write process from the memory to return a query faster. It stores data in a way that greatly improves disk I/O performance.
They are majorly helpful for data analytics and data warehousing.

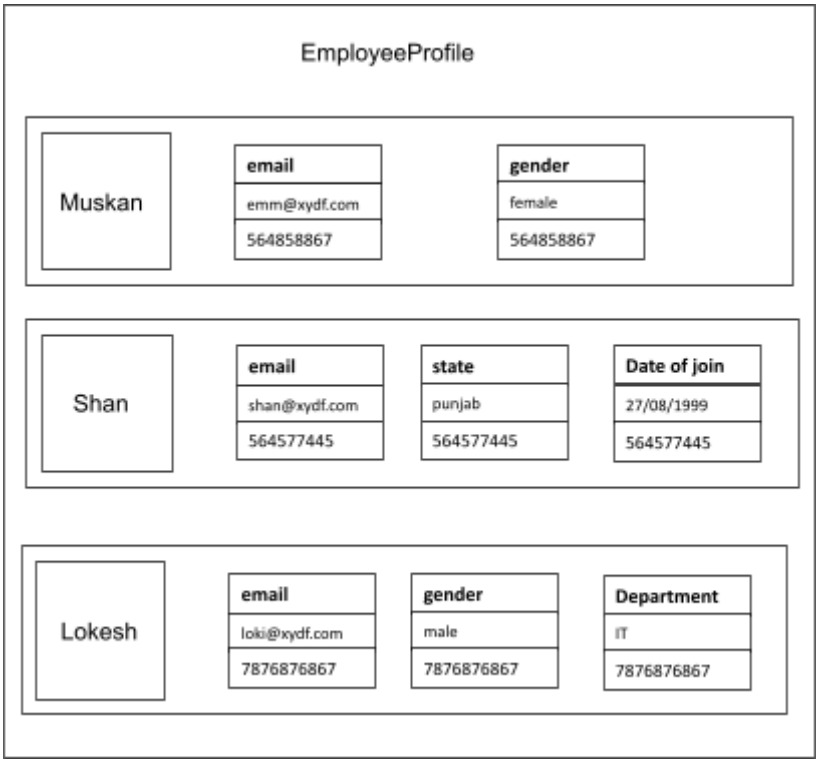
- **How does this work ?**
  Data is organized into columns instead of rows. Hence they function same as those tables in relational databases but being a NoSQL it's model allows it to be much more flexible.
  It uses the concept of keyspace, which is similar to schema in relational models. It contains all the column families, which further contain rows, which then contain columns.

  Below is the representation of column families in the key space,



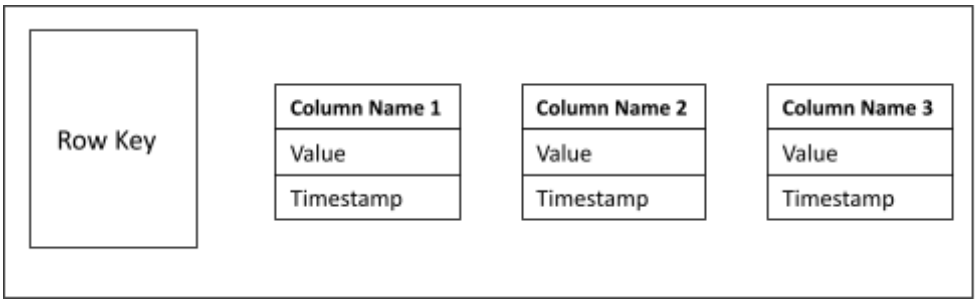
  Each column family contains rows, these rows further contain columns with different names, links, sizes i.e. they don't need to be of the same size, they can vary.
  These columns only exist within their own row and contain a value with the timestamp it was entered on.

Let’s view a one column family. Let it be of an employee of a company:



Here, we see the identifier be the employees name, and each employee has different kinds of data present along with their names.

Now, Let’s view one row for an example.



Row key here is an unique identifier of the row.

➢ Some Columnar NoSQL Database - *Casandra, CosmoDB, HBase*

- **Benefits of Column based NoSQL Database:**
  There are few advantages of using columnar database:
  - Efficient in terms of storage as column stores are good at compression. Hence, we reduce disk resources when handling huge data chunks in a single column.
  - Aggregation queries operate fast.

- Highly scalable, can be scaled out infinitely.
- Load and query time is quick.
- Self-indexing
- Extreme flexibility as columns can be added without disrupting the database.
- Great for analytics and reporting.

- **Limitations:**
  - It’s hard and time consuming to design an index schema for a columnar database.
  - Incremental data loading is not suitable enough.
  - Security issues and vulnerabilities in web apps.
  - Online Transaction Processing (OLTP) applications don't consider column based databases due the fashion in which data is stored here.
  - Can’t write complex queries, it’s hard to connect different columns and extract the desired information.

On observing it could be said that column based databases are nearly identical to SQL methods. The keyspaces here act as schema, thus some kind of schema management is being done. Also, the metadata can be similar to that of a typical relational DBMS.

Such databases are quite impactful but just like everything isn’t perfect they also do offer some limitations as well.
Like, in comparison to relational databases which are written sequentially to the disk on the other hand columnar NoSQL databases lack consistency due to multiple writes to the memory.
Irrespective of all that, they are still one of the most used data models.