MONGO DB

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

The following table shows the relationship of RDBMS terminology with MongoD :

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key
Database Server and Client	
Mysqld/Oracle	mongodb

Advantages of MongoDB over RDBMS

- *Schema less: MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.*

- *Structure of a single object is clear.*

- *No complex joins.*

- *Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.*

- *Tuning.*

- *Ease of scale-out: MongoDB is easy to scale.*

- *Conversion/mapping of application objects to database objects not needed.*

- *Uses internal memory for storing the (windowed) working set, enabling faster access of data.*

Create and load the dataset using the Mongo Db Using:

- *Go to MongoDB Compass*
- *Click on connect to connect to the server*
- *Then click on + symbol to create a database and enter the database name .*
- *Later enter the collection name and click on create.*
- *Import the data inside the collections using “add data “ where we can import json or csv datasets.*

To see all the records inside the table :

> db.people.find().pretty();

To see the only one record inside the table :

> db.people.findOne().pretty();

Inserting the single record inside the table :

```
> db.people.insert([{"Index": 101,"First Name":"Jeeva","Last
Name":"Anand","Sex":"Male"}]);
```

```
BulkWriteResult({
"writeErrors" : [ ],
"writeConcernErrors" : [ ],
"nInserted" : 1,
"nUpserted" : 0,
"nMatched" : 0,
"nModified" : 0,
"nRemoved" : 0,
"upserted" : [ ]    })
```

Inserting the single record inside the table :

```
> db.people.insertMany([{"Index": 102,"First Name":"Shiyam","Last
Name":"K","Sex":"Male"},{"Index": 103,"First Name":"Chetna","Last
Name":"Shekhawat","Sex":"Female"},{"Index": 104,"First Name":"Thrupthi","Last
Name":"M","Sex":"Female"}]);

{

        "acknowledged" : true,

        "insertedIds" : [

                ObjectId("64bf9e84f5e52b8b4a1d338f"),

                ObjectId("64bf9e84f5e52b8b4a1d3390"),

                ObjectId("64bf9e84f5e52b8b4a1d3391")

        ]

}
```