# MongoDB

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 systems. 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. Collection 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.

Documents – A document is a set of keys– 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 MongoDB.

|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Tuple / Row | Document |
| Column | Field |
| Table Join | Embedded Document |
| Primary key | Primary Key (Default \_id provided by MongoDB itself) |

**Database Server and Client**

|  |  |
| --- | --- |
| Mysqld / Oracle | Mongod |
| Mysql / sqlplus | mongod |

**#Create a database**

Use rst

#check which databases are created show dbs

Show dbs

#show current db

Db

#insert data

Db.movie.insert({“name”:”rst forum”})

#drop a Db:

Use rst

Switched to db mydb

Db.drop.Database()

{“dropped”:”rst”,”ok”:1}

Show dbs