# Overview

MongoDB is a **document-oriented database** which works on concept of collection and document. It's a typical **non-SQL** database.

## Basic Concepts

* **Database**: Database is a **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 documents**. Typically, all documents in a collection are of similar or related purpose.
* **Document**:A document is a **set of key-value pairs**. Documents within a collection can have different fields.

## MongoDB vs RDBMS

The following table shows the relationship of RDBMS terminology with MongoDB.

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB** |
| Database | Database |
| Table | Collection |
| Row | Document |
| Column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key  (Default key \_id provided by MongoDB itself) |
| **Database Server and Client** | |
| mysqld/Oracle | mongod |
| mysql/sqlplus | mongo |

# Data Modelling

Data in MongoDB has a flexible schema. 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.

MongoDB provides two types of data models:

## Embedded Data Model

In this model, you can have (embed) all the related data in a single document, it is also known as *de-normalized data model*.

For example, assume we are getting the details of employees in three different documents namely, Personal, Contact and Address. You can embed all the three documents in a single one as shown below:

{

    \_id: ,

    emp\_id: "10025AE336"

    Personal: {

        first\_name: "Radhika",

        last\_name: "Sharma",

        date\_of\_birth: "1995-09-26"

    },

    Contact: {

        e-mail: "radhika\_sharma.123@gmail.com",

        phone: "9848022338"

    },

    Address: {

        city: "Hyderabad",

        area: "Madapur",

        state: "Telangana"

    }

}

If you store these data in RDBMS, you'll need three tables (Personal, Contact and Address). They all have the same column emp\_id which will be used to join the tables. In MongoDB, data is shown from one collection only.

## Normalized Data Model

In this model, you can refer the sub documents in the original document, using references. For example, you can re-write the above document in the normalized model as:

Employee:

{

    \_id: <ObjectId101>,

    emp\_id: "10025AE336"

}

Personal:

{

    \_id: <ObjectId102>,

    emp\_doc\_id: " ObjectId101",

    first\_name: "Radhika",

    last\_name: "Sharma",

    date\_of\_birth: "1995-09-26"

}

Contact:

{

    \_id: <ObjectId103>,

    emp\_doc\_id: " ObjectId101",

    e-mail: "radhika\_sharma.123@gmail.com",

    phone: "9848022338"

}

Address:

{

    \_id: <ObjectId104>,

    emp\_doc\_id: " ObjectId101",

    city: "Hyderabad",

    area: "Madapur",

    state: "Telangana"

}

# Commands

<https://www.tutorialspoint.com/mongodb/mongodb_create_database.htm>

<https://docs.mongodb.com/manual/reference/mongo-shell/>

<https://documents.pub/document/mongodb-cheat-sheet-quick-reference.html>

# MongoDB and PHP

<https://www.tutorialspoint.com/mongodb/mongodb_php.htm>