

DataModel

MongoDB make data survive in flexible schema model. In RDBMS, it is must to create table's schema before inserting data, but here in Mongoddb Collections dontenfore document structure.

What are key requirements for any application?

- Performance of database engine during query execution and ease of retrieval of data. Hence the data model should be very important in the application use of data.
- Here everything revolves around documents and their relationships. For this there are 2 types of data models.
 1. Normalized data models
 2. Embedded data models

Normalized Data Model:

- Normalized data model allows us to depict the relationships using references between documents.
- Let us consider the below example where in this normalized data model, the Customer delivery address document contains a reference to the Cust_id document.

Example:

```
{
  _id: "123",
  Customer_name: "Maria",
}
{
  Cust_id:"123"
  street: "123 street",
  city: "london",
  state: "UK",
  zip: "12345"
}
```

These Models can be used in below cases:

- When embedding resulting in duplication of data.

DataModel

- When you need to model large hierarchical data sets and complex many-to-many relationships.

Embedded Data Model:

- Embedded data model allows us to store the relative information coupled together in the same database record. Hence, the applications may need less number of queries to completed regular operations.
- Let us consider the below example again as if the customer delivery address data is frequently retrieved with the customer information, then instead of shooting multiple queries to resolve the reference. The better data model should be to embed the customer delivery address data within the single document.

```
{
  _id: "123",
  Customer_name: "Maria",
}
{
  Cust_id:"123"
  street: "123 street",
  city: "london",
  state: "UK",
  zip: "12345"
}
```

These Models can be used in below cases:

- When you have “contains” relationships between entities (one-to-one).
- When you have “many” or child documents can be enclosed as one (one-to-many).