



MongoDB Basics

Summary: in this tutorial, you'll learn some basic concepts of MongoDB such as documents, collections, databases, and namespaces.

Data formats

In MongoDB, you will often deal with JSON and BSON formats. Therefore, it's important to fully understand them.

JSON

JSON stands for JavaScript Object Notation. JSON syntax is based on a subset of JavaScript ECMA-262 3rd edition.

A JSON document is a collection of fields and values in a structured format. For example:

```
{
  "first_name": "John",
  "last_name": "Doe",
  "age": 22,
  "skills": ["Programming", "Databases", "API"]
}
```

BSON

BSON stands for Binary JSON, which is a binary-coded serialization of JSON-like documents.

Documents

MongoDB stores data records as BSON documents, which are simply called documents.

```
{
  _id: ObjectId("5f339953491024badf1138ec"),
  title: "MongoDB Tutorial",
  isbn: "978-4-7766-7944-8",
  published_date: new Date('June 01, 2020'),
  author: {
    first_name: "John",
    last_name: "Doe"
  }
}
```

A document is a set of field-and-value pairs with the following structure:

```
{
  field_name1: value1,
  field_name2: value2,
  field_name3: value3,
  ...
}
```

In this syntax, the field names are strings and values can be numbers, strings, objects, arrays, etc. For example:

```
{
  _id: ObjectId("5f339953491024badf1138ec"),
  title: "MongoDB Tutorial",
  isbn: "978-4-7766-7944-8",
  published_date: new Date('June 01, 2020'),
  author: { first_name: "John"
, last_name: "Doe"}
}
```

This document has the following field-and-value pairs:

- `_id` holds an `ObjectId`

- `title` holds a string.
- `isbn` holds a string.
- `published_date` holds a value of the `Date` type.
- `author` holds an embedded document that contains two fields `first_name` and `last_name`.

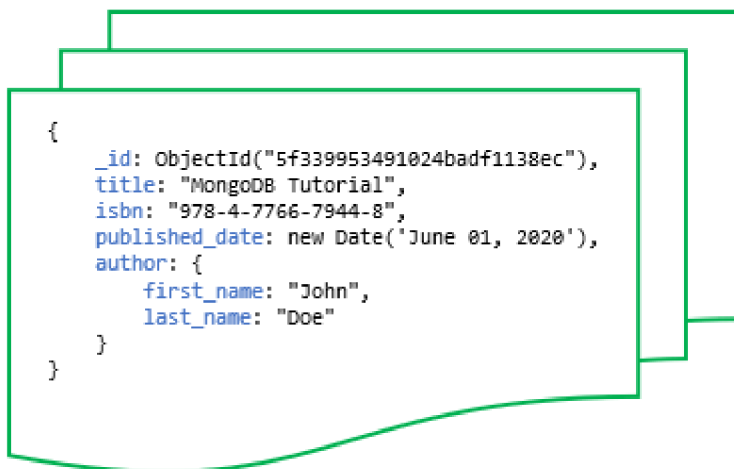
If you are familiar with a relational database management system (RDBMS), you will find that a document is similar to a row in a table, but it is much more expressive.

Field names have the following restrictions:

- MongoDB reserves the field `_id` and uses it to uniquely identify the document.
- Field names cannot contain null characters.
- Top-level field names cannot start with the dollar sign (`$`) character.

Collections

MongoDB stores documents in a collection. A collection is a group of documents.



A collection is analogous to a table in an RDBMS.

MongoDB	RDBMS
Documents	Rows
Collections	Tables

Unlike a table that has a fixed schema, a collection has a dynamic schema.

It means that a collection may contain documents that have any number of different "shapes". For example, you can store the following documents in the same collection:

```
{
  title: "MongoDB Tutorial",
  published_date: new Date('June 01, 2020')
}

{
  title: "MongoDB Basics",
  published_date: new Date('Jan 01, 2021'),
  isbn": "978-4-7766-7944-8"
}
```

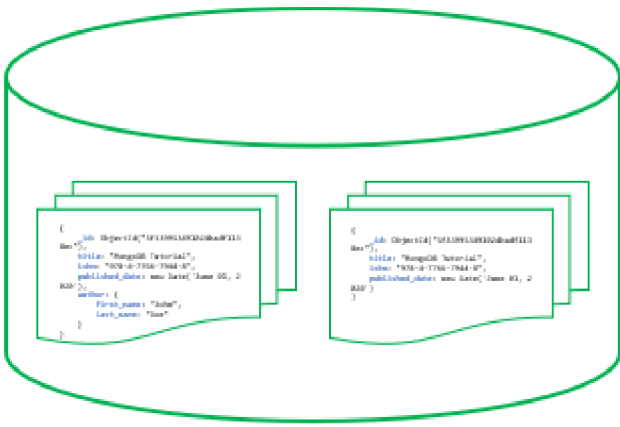
Note that the second document has one more field than the first one. In theory, you can have completely different fields for every document.

A collection has a name e.g., `books` . The collection name cannot:

- contain the dollar sign (`$`)
- contain the null character (`\0`).
- be an empty string.
- begin with the system because MongoDB reserves `system*` for internal collection names.

Databases

MongoDB stores collections into a database. A single instance of MongoDB can host multiple databases.



A database can be referenced by a name for example `bookdb` . The database names cannot:

- Be an empty string (`""`).
- Contain any of these characters: `/, \, ., ", *, <, >, :, |, ?, $`, (a single space), or `\0` (the `null` character).
- Exceed the maximum size which is 64 bytes.

MongoDB also has some reserved database names such as `admin` , `local` , and `config` that you cannot use to create new databases.

Namespace

A namespace is a concatenation of the database name with a collection in that database. Namespaces allow you to fully qualify collections.

For example, if the collection name is `books` and database name is `bookdb` , the namespace of the `books` collection would be `bookdb.books` .

Summary

- MongoDB stores data records as BSON documents. A document is a set of field-and-value pairs.
- MongoDB stores documents in a collection and collections in a database.
- A namespace is a concatenation of the database name and the collection name (`database_name.collection_name`) to fully qualify the collection.