Databases and Collections

On this page

Overview

MongoDB stores data records as documents (specifically BSON documents) which are gathered together in collections. A database stores one or more collections of documents.

Databases

In MongoDB, databases hold one or more collections of documents. To select a database to use, in the `mongo` shell, issue the `use <db>` statement, as in the following example:

```
use myDB
```

Create a Database

If a database does not exist, MongoDB creates the database when you first store data for that database. As such, you can switch to a non-existent database and perform the following operation in the `mongo` shell:

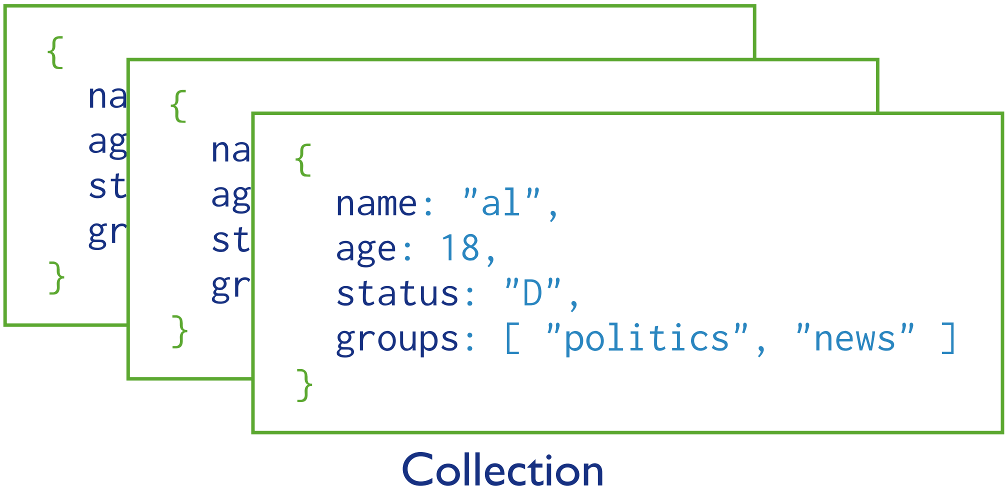
```
use myNewDB

db.myNewCollection1.insertOne( { x: 1 } )
```

The `insertOne()` operatio
they do not already exist. Be
Restrictions.

Collections

MongoDB stores documents in collections. Collections are analogous to tables in relational databases.



click to enlarge

Create a Collection

If a collection does not exist, MongoDB creates the collection when you first store data for that collection.

```
db.myNewCollection2.insertOne( { x: 1 } )
db.myNewCollection3.createIndex( { y: 1 } )
```

Both the `insertOne()` and the `createIndex()` operations create their respective collection if they do not already exist. Be sure that the collection name follows MongoDB Naming Restrictions.

Explicit Creation

MongoDB provides the `db.createCollection()` method to explicitly create a collection with various options, such as setting the maximum size or the documentation validation rules. If you are not specifying these options, you do not need to explicitly create the collection since MongoDB creates new collections when you first store data for the collect

To modify these collection options, see `collMod`.

Document Validation

New in version 3.2.

By default, a collection does not require its documents to have the same schema; i.e. the documents in a single collection do not need to have the same set of fields and the data type for a field can differ across documents within a collection.

Starting in MongoDB 3.2, however, you can enforce document validation rules for a collection during update and insert operations. See Schema Validation for details.

Modifying Document Structure

To change the structure of the documents in a collection, such as add new fields, remove existing fields, or change the field values to a new type, update the documents to the new structure.

Unique Identifiers

New in version 3.6.

> **NOTE:**
>
> The `featureCompatibilityVersion` must be set to `"3.6"` or greater. For more information, see View FeatureCompatibilityVersion.

Collections are assigned an immutable UUID (Universally unique identifier). The collection UUID remains the same across all members of a replica set and shards in a sharded cluster.

To retrieve the UUID for a collection, run either the listCollections command or the `db.getCollectionInfos()` method.