Mongo DB Commands

mongosh: connects with the local MongoDB database on the localhost:27017

show dbs: it will display a list of all the available databases

use database\_name: to create a database if it does not exist, else, it returns to the already existing database.

db.createCollection(name, options): MongoDB creates a collection for an inserted command automatically if no similar collection already exists in the MongoDB database

show collections: To obtain a list of MongoDB collections, we need to use the Mongo shell command show collections

MongoDB CRUD Operations

Create

db.collection.insertOne(document, options):method in MongoDB is used to insert a single document into a specified collection.

db.collection.insertMany(document,document, options): method in MongoDB is used to insert a multiple document into a specified collection

Read

db.collection.find().pretty():Configures the cursor to display results in an easy-to-read format.

Update

db.collection.updateOne(

{ <filter> },

{ <update> }

) method to update one documents in a collection

updateMany() for more precise updates.

<filter>: Specifies the selection criteria for the update

<update>: Specifies the modifications to apply. This can include $set to update specific fields, $inc to increment/decrement a numeric value, and other update operators.

E.g. db.myCollection.updateMany(

{ status: "active" }, // Update documents with 'status' equal to "active"

{ $set: { status: "inactive" } } // Set the 'status' field to "inactive"

)

To delete only one field or element within a document in MongoDB, you can use the $unset operator in combination with the updateOne()

db.myCollection.replaceOne(

{ name: "John Doe" }, // Replace the document with the specified name

{ name: "Jane Doe", age: 32, email: "jane.doe@example.com" } // New document

)

If you want to completely replace a document in MongoDB, you can use the replaceOne() method. This method replaces the first document that matches a given filter with a new document

If you want to replace multiple documents, you can use replaceMany()

Delete

db.collection.deleteOne()

Delete at most a single document that match a specified filter even though multiple documents may match the specified filter.

db.collection.deleteMany()

Delete all documents that match a specified filter.

db.collection.remove()

Delete a single document or all documents that match a specified filter.

db.collection\_name.drop():delete a table, or collection as it is called in MongoDB, by using the drop() method.

db.dropDatabase():To delete database