MongoDB

Commands

CRUD Operations in MongoDB

- To start working with MongoDB databases

- c:\>mongod

- MongoD is the host for MongoDB databases. It must exist as background system process to work with the databases.

- To open theMongoDB shell

- c:/>mongo

- Mongo is the MongoDB shell where we can type commands to work with documents, collections and databases.

Listing Databases

- To check for the databases already present
- >show dbs
- OR
- >show databases

Creating Database

- >use database_name
- Ex:
- >use school
- Will create a database called school
- MongoDB is case sensitive – 'school' is different from 'SCHOOL'
- To clear the contents of the command prompt screen
- >cls
- To display the currently selected database
- >db

Creating and displaying collection

- Like table is created in SQL database, collection is created in MongoDB database.

- As records are inserted in SQL table, documents are inserted in MongoDB collection.

- >db.createCollection(name, options)

- name-name of the document (always string), options – size, type and indexing

- Options parameter is optional.

- >db.createCollection("student")

Creating documents

- Insert() method is used to insert one or more than one document record in a collection.

- db.collection_name.insert({name:"value"})

- Example:

- >db.student.insert({RollNo:1, Name:"Vijay", marks:90})

- Documents in MongoDB are dictionary items having the syntax:

- {key1:value1,key2:value2,...}

- MongoDB creates collection automatically when we insert some documents.

Inserting multiple documents – using insertMany()

▶ Example

▶ >db.student.insertMany([{RollNo:3, Name:"Surya", Marks:95},

{RollNo:4, Name:"Ram", Marks:65},{RollNo:5, Name:"Radhika", Marks:97}])

Note: On executing the above command, three documents shall be inserted in the "student" collection and the ObjectId for all 3 documents shall be automatically generated by MongoDB.

Listing/Querying documents

▶ To query data from MongoDB collection,

▶ >db.collection_name.find()

▶ Example:

▶ >db.student.find()

Note: Data in MongoDB has a flexible schema. The documents can be different in the same collection. They do not need to have the same set of fields or structure, and common fields. A collection's document may hold different types of data.

Updating documents – using update()

▶ update() method is used to modify values in the existing document.

▶ Example

▶ >db.student.update({RollNo:8},{$set:{'Garde':'B'}})

▶ The above command will update grade as 'B' for the student whose rollno is 8.

Updating documents – using save()

▶ Save() method replaces the existing document with the new document passed in the save() method.

▶ While using save(), if the document exists it will be replaced otherwise it will be created.

▶ Example:

▶ >db.student.save({_id:100, Name:'Deepa'})

▶ To search for the record,

▶ >db.student.find()

Removing a document – using remove()

- remove() method is used to remove a document from the collection.

- Example:

- To delete the document from the student collection whose result is "pass", the command is,

- >db.student.remove({Result":"pass"})

- If criterion is missing, all documents will be deleted

- To remove all the documents,

- >db. Remove()

Removing/Dropping a collection

- drop() method is used to remove a collection permanently from a database.
- Example:
- >db.student.drop()
- >show collections
- …………. (empty)

Removing/Dropping a database

- dropDatabase() method is used to remove a MongoDB database permanently including its contents.

- Example:

- >db.dropDatabase()

- In order to drop 'school' database, ensure that the current database is 'school' by giving command >db.

- dropDatabase() method always removes/deletes the currently selected database.

- We can confirm the deletion using

- >show db

Exiting MongoDB

- To move out of MongoDB server,

- >exit

- Will show the default message

- Bye

- As output