MongoDB - collection

Topics : MongoDB
Written on December 30, 2023

A MongoDB collection is a grouping of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database and does not enforce a schema. In MongoDB, documents within a collection can have different fields and data types.

1. **Dynamic Schema:**
    - MongoDB is schema-less, meaning documents within a collection do not need to have the same structure. Each document can have different fields.
2. **Creation:**
    - Collections are created implicitly when the first document is inserted. MongoDB creates a collection dynamically as soon as data is inserted into it.
3. **Accessing Collections:**
    - In the MongoDB shell, you can access a collection using the db.collection_name notation. For example, db.users refers to the "users" collection.
4. **Inserting Documents:**
    - You can insert documents into a collection using the insertOne or insertMany methods.

      db.collection_name.insertOne({ field1: value1, field2: value2 });
5. **Querying Documents:**
    - The find method is used to query documents in a collection. It returns a cursor to the documents that match the query criteria.

      db.collection_name.find({ field: value });
6. **Updating Documents:**
    - You can update documents in a collection using the updateOne or updateMany methods.

db.collection_name.updateOne({ field: value }, { $set: { updated_field: new_value } });

7. **Deleting Documents:**

   - Documents can be deleted using the deleteOne or deleteMany methods.

     db.collection_name.deleteOne({ field: value });

8. **Indexing:**

   - Indexes can be created on fields to improve query performance.

     db.collection_name.createIndex({ field: 1 });

9. **Aggregation:**

   - MongoDB supports aggregation pipelines for complex data transformations.

```
db.collection_name.aggregate([
  { $group: { _id: "$field", count: { $sum: 1 } } }
]);
```