

Assignment - B2

Title :- Design & develop MongoDB queries using CRUD operations.

Problem Statement :-

Design & develop MongoDB queries using CRUD operations (use CRUD operation, SAVE method, logical operations, comparative operations & embedded documents).

Objective :- To understand & implement CRUD operations in MongoDB.

S/W & H/W :- MongoDB, Fedora OS.

Outcome :-

- Implement the commands on two tiers.
- Implement the database in MongoDB.

Theory :-

CRUD operations

c - create - mongo store the data in form of JSON objects. so every

record for a collection, in mongo is called a document. If the collection does not currently exist, insert operations will create the collection. We can ~~inter~~ insert documents into collection in 3 ways.

- i) `insert_one()`
- ii) `insert_many()`
- iii) `insert()`

R - Read -

We can retrieve the document from collection using 2 methods.

- `find()`
- `find_one()`

`find()` functional will return all the documents in that collection by default. It returns a cursor objects.

`find_one()`: returns the first document in the collection.

D - Delete -

We can delete the document in that collection using following methods.

- `delete_one()`
- `delete_many()`

Both of these methods will return a DeleteResult object the general syntax As above methods

<method.name> (condition)

U - Update -

We can update the documents from the collection with the following methods.

update(), updateOne(),
updateMany(), update
replace-one().

The general syntax for all the above method is

<methodname> (condition, update, upsert = false,
bypass : document_validation = false),

Logical query operators

\$OR :- Joins query clauses with a logical OR returns all documents that match the condition of either clause.

{ \$or [{ <expression> }, { <expression> }, ...] }

\$and - Joins query clauses with a logical AND returns all documents that match the conditions of both clauses

$$\{ \$and : [\{ <expression> \}, \{ <expression> \}, \dots] \}$$

\$not - Inverts the effect of a query expression & returns documents that do not match the query expression.

$$\{ field : \{ \$not : \{ <operator> <expression> \} \} \}$$

\$nor - Joins query clauses with a logical NOR return all documents that fail to match both clauses.

$$\{ \$nor : [\{ <expression> \}, \{ <expression> \}, \dots] \}$$

Conclusion :

Implemented CRUD operations successfully using logical, comparator operators.