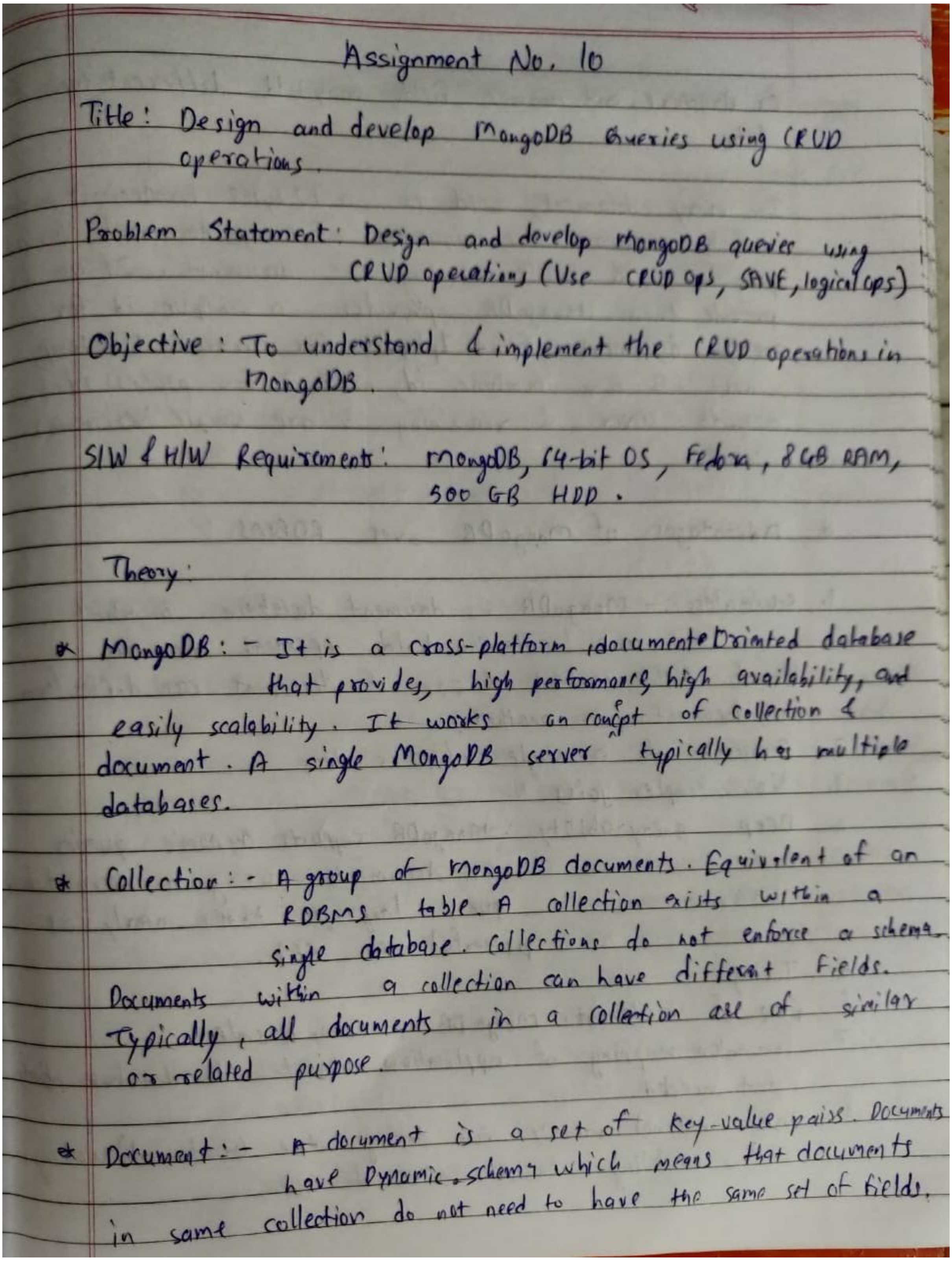
//31139 (Durvesh)



Assignment No. 10

Title: Design and develop MongoDB Queries using CRUD operations.

Problem Statement: Design and develop mongoDB queries using CRUD operations (Use CRUD ops, SAVE, logical ops)

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

S/W & H/W Requirements: mongoDB, 64-bit OS, Fedora, 8GB RAM, 500 GB HDD.

Theory:

- MongoDB: - It is a cross-platform, document oriented database that provides, high performance, high availability, and easily scalability. It works on concept of collection & document. A single MongoDB server typically has multiple databases.

- Collection: - A group of MongoDB documents. Equivalent of an RDBMS table. A collection exits within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

- Document: - A document is a set of key-value pairs. Documents have Dynamic schema which means that documents in same collection do not need to have the same set of fields.

Or structure, and common fields may hold different types of data.

In every document, _id is a 12 bytes hexadecimal number which assures the uniqueness of every document. You can provide _id while inserting the document. If you didn't provide then MongoDB provides a unique id for every document. First 4 bytes for current timestamp, next 3 for machine id, next 2 for process id of mongodb server & remaining 3 are simple incremental value.

* Advantages of MongoDB over RDBMS :

1. Schemaless :- MongoDB is document database in which one collection holds different documents. No. of fields, content & size of document can differ from one document to another.
2. Structure of a single objet is clear
3. No complex joins
4. Deep query-ability :- MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
5. Tuning
6. Ease of scale-out: MongoDB is easy to scale
7. Conversion/mapping of application objects to database objects not needed.
8. Use internal memory for storing the (windowed) working set, enabling faster access of data.

MongoDB CRUD operations with Python (Pymongo)

Pymongo is a python distribution that contains tools for working with mongoDB, ~~so in this~~

Install Pymongo:

```
pip install pymongo
```

Connecting to Database with PyMongo

```
>>> from pymongo import MongoClient
>>> client = MongoClient('localhost',27017)
    db = client.test                          (test database)
    col = db.person                           (person collection)
```

CRUD operations:-

C - Create : mongo stores the data in the 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 insert the documents into collection in 3 ways.

1. insert_one() :- inserts a single document into a collection.
2. insert_many() :- inserts many documents into a collection.
3. insert() :- insert() can be used to insert single or array or documents.

R - Read: We can retrieve the documents from a collection using 2 methods.

1. find() :- will return with all the documents in that collection. By default it returns a cursor object.

2. find_one() :- returns the first document in the collection.

D - Delete :- We can delete the documents in the collection using following methods.

1. delete_one()
2. delete_many()

Both these methods will return a DeleteResult object.

U - Update :- We can update the documents from the Collection with the following methods.

1. update()
2. update_one()
3. update_many()
4. replace_one()

General syntax :-

<method_name>(condition, update_or_replace_document, upsert=false, bypass_document_validation = False)

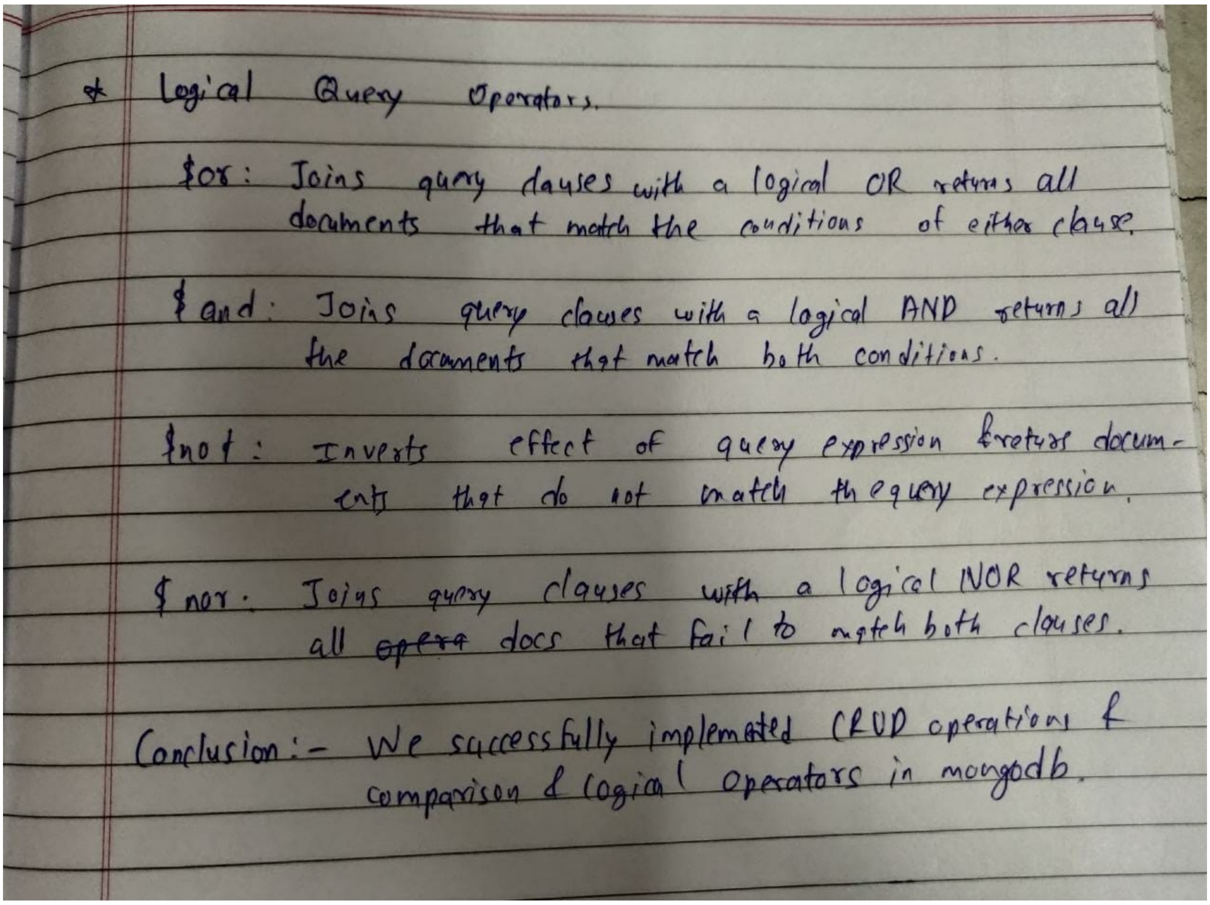
Here,

condition : A query that matches the document to replace.

update_or_replace_document : The new document

upsert (optional) : If True, perform an insert if no documents match the filter.

bypass_document_validation : (optional) If True, allows the write to opt-out of document level validation. Default is False.



* Logical Query Operators.

$or: Joins query clauses with a logical OR returns all documents that match the conditions of either clause.

$and: Joins query clauses with a logical AND returns all the documents that match both conditions.

$not: Inverts effect of query expression & returns documents that do not match the query expression.

$nor: Joins query clauses with a logical NOR returns all ~~opera~~ docs that fail to match both clauses.

Conclusion:- We successfully implemated CRUD operations & comparison & logical operators in mongodb.