# MONGODB CRUD TASK

### ****1. Create Operation****

#### ****1.1 Create a new database and switch to it****

use library

**Explanation:**

The use command switches to the specified database. If the database does not exist, it is created.

#### ****1.2 Insert a document into the "books" collection****

db.books.insertOne({

title: "To Kill a Mockingbird",

author: "Harper Lee",

published\_year: 1960

})

**Explanation:**

The insertOne method adds a single document to the "books" collection. The collection is created automatically if it doesn’t exist.

### ****2. Operation Read****

#### ****2.1 Retrieve all documents from the "books" collection****

db.books.find()

**Explanation:**

The find method without arguments retrieves all documents from the collection.

#### 2.2 Find documents where the author is "J.K. Rowling"

db.books.find({ author: "J.K. Rowling" })

**Explanation:**

The find method with a query filter retrieves documents matching the specified condition.

#### ****2.3 Fetch the document with the earliest published year****

db.books.find().sort({ published\_year: 1 }).limit(1)

**Explanation:**

The sort method orders documents in ascending order by the published\_year. The limit method ensures only the earliest document is returned.

### ****3. Update Operation****

#### ****3.1 Update the published year of "The Catcher in the Rye" to the current year****

db.books.updateOne(

{ title: "The Catcher in the Rye" },

{ $set: { published\_year: 2015 } })

**Explanation:**

The updateOne method updates the first document matching the query filter. The $set operator modifies the specified field.

#### ****3.2 Add a new field "genre" with the value "Mystery" to all documents****

db.books.update({}, { $set: { genre: "Mystery" } })

**Explanation:**

The updateMany method updates all documents in the collection. The $set operator adds or updates the specified field.

### ****4. Delete Operation****

#### ****4.1 Remove the document with the title "1984"****

db.books.deleteOne({ title: "1984" })

**Explanation:**

The deleteOne method removes the first document matching the query filter.

#### ****4.2 Delete all documents where the published year is before 2000****

db.books.deleteMany({ published\_year: { $lt: 2000 } })

**Explanation:**

The deleteMany method removes all documents matching the query filter. The $lt operator matches documents with a published\_year less than 2000.

### ****5. Advanced Query****

#### ****5.1 Find the top 3 recently published books****

db.books.find().sort({ published\_year: -1 }).limit(3)

**Explanation:**

The sort method orders documents in descending order of the published\_year. The limit method ensures only the top 3 documents are returned.

#### 5.2 Retrieve documents where the title contains "MongoDB"

db.books.find({ title: /MongoDB/i })

**Explanation:**

The query db.books.find({ title: /MongoDB/i }) uses a **regular expression** (/MongoDB/i) to search for documents in the books collection where the title field contains the word "MongoDB".

The /i flag ensures the search is **case-insensitive**, so it will match titles like "MongoDB Basics", "Learn mongodb", or "Working with MongoDB".

.



