THEORY

1. Insert Operations

Q1. What is the purpose of insertOne() in MongoDB?

- insertOne() is used to insert a single document into a collection.
- MongoDB is schema-less, so the document can have any structure.
- If the collection does not exist, MongoDB automatically creates it.

Q2. Write a MongoDB command to insert a single document.

```
db.students.insertOne({
  name: "John Doe",
  course: "Mathematics",
  year: 2
});
```

Q3. How does MongoDB behave if the collection does not exist while using insertOne()?

- MongoDB automatically creates the collection before inserting the document.

Q4. What is insertMany() and how is it different from insertOne()?

- insertMany() inserts multiple documents in one operation.
- Difference: insertOne() inserts a single document, while insertMany() can insert many at once, which is faster for bulk inserts.

Q5. What happens if one document fails during insertMany() execution?

- By default, MongoDB uses ordered inserts (ordered=true). If one document fails, insertion stops immediately.
- If ordered=false, MongoDB continues inserting remaining documents and reports which ones failed.

---

2. Update Operations

Q1. What is the difference between updateOne() and updateMany()?

- updateOne() updates only the first document that matches the filter.

- updateMany() updates all documents that match the filter.
- Useful for controlling whether the update affects one or multiple records.

Q2. Write a command to update only one document's field in MongoDB.

```
db.employees.updateOne(
  { name: "Jane Smith" },
  { $set: { salary: 65000 } }
);
```

- $set modifies the specified field without affecting the rest of the document.

Q3. How do you update multiple documents at the same time in MongoDB?

```
db.products.updateMany(
  { category: "Electronics" },
  { $inc: { price: 50 } }
);
```

- $inc increases numeric fields.
- All products in the "Electronics" category will have their price increased by 50.

Q4. What happens if the filter condition in updateOne() matches multiple documents?

- Only the first matched document is updated; the rest remain unchanged.

Q5. What update operator is used to modify specific fields without replacing the entire document?

- $set → updates specific fields.
- $inc → increments numeric fields.
- $currentDate → updates a field to the current date.

---

3. Add Date (Current Date)

Q1. How do you add the current date to a document while inserting data?

- Use JavaScript new Date():

db.logs.insertOne({ action: "Login", timestamp: new Date() });

Q2. Which MongoDB operator is equivalent to SQL CURDATE()?

- $currentDate sets a field to the current date/time automatically during updates.

Q3. Write a command to add the current date to all existing documents.

```
db.users.updateMany(
  {},
  { $currentDate: { lastUpdated: true } }
);
```

Q4. What is the difference between using new Date() and $currentDate?

- new Date() → generates the exact date when the command runs, used for insert or update.
- $currentDate → is an update operator, automatically sets the field to the system's current date without manually creating a Date object.

---

4. Delete Operations

Q1. What is the difference between deleteOne() and deleteMany()?

- deleteOne() → deletes the first document matching the filter.
- deleteMany() → deletes all documents matching the filter.

Q2. Write a MongoDB command to delete a single document.

db.users.deleteOne({ name: "Ali" });

Q3. What happens if the filter in deleteMany() is empty {}?

- Deletes all documents in the collection.
- The collection itself remains, along with its indexes.

---

5. Drop Operations

Q1. How do you drop a collection (table) in MongoDB?

db.students.drop();

- Deletes the collection completely, including all documents and indexes.

Q2. What is the command to drop an entire database in MongoDB?

db.dropDatabase();

- Deletes the database and all its collections.

Q3. What is the difference between deleteMany({}) and drop()?

- deleteMany({}) → deletes all documents but keeps collection structure and indexes.
- drop() → deletes the collection entirely, including indexes and structure.

PRACTICAL

1. InsertOne (Insert Single Document)

Q1. How do you insert a single student record?
A1: Use insertOne() to insert one document into a collection. If the collection does not exist, MongoDB creates it automatically.

Example:

```
db.students.insertOne({
  name: "John Doe",
  course: "Computer Science",
  year: 2
});
```

Q2. How do you insert a single employee document?
A2: Use insertOne() on the staff collection.

Example:

```
db.staff.insertOne({
  name: "Jane Smith",
  role: "Developer",
  salary: 60000,
```

joiningDate: new Date("2023-06-15")

});

Q3. How do you create a product document?
A3: Use insertOne() on the products collection.

Example:

```
db.products.insertOne({
  name: "Laptop",
  price: 1200,
  category: "Electronics"
});
```

---

2. InsertMany (Insert Multiple Documents)

Q1. How do you insert multiple employee records at once?
A1: Use insertMany() to add multiple documents in a single operation.

Example:

```
db.staff.insertMany([
  { name: "Alice", age: 25, role: "HR" },
  { name: "Bob", age: 30, role: "Manager" },
  { name: "Charlie", age: 28, role: "Developer" },
  { name: "David", age: 35, role: "Tester" },
  { name: "Eva", age: 22, role: "Intern" }
]);
```

Q2. How do you insert multiple products of different categories?
A2: Use insertMany() for the products collection.

Example:

```
db.products.insertMany([
  { name: "Phone", price: 700, category: "Electronics" },
```

```
  { name: "Shirt", price: 40, category: "Clothing" },
  { name: "Book", price: 15, category: "Stationery" },
  { name: "Headphones", price: 120, category: "Electronics" },
  { name: "Shoes", price: 80, category: "Footwear" }
]);
```

Q3. How do you insert multiple orders?
A3: Use insertMany() in the orders collection.

Example:

```
db.orders.insertMany([
  { orderId: 101, employeeName: "Alice", totalAmount: 150 },
  { orderId: 102, employeeName: "Bob", totalAmount: 300 },
  { orderId: 103, employeeName: "Charlie", totalAmount: 250 },
  { orderId: 104, employeeName: "David", totalAmount: 400 },
  { orderId: 105, employeeName: "Eva", totalAmount: 100 }
]);
```

---

3. UpdateOne (Update Single Document)

Q1. How do you update the salary of one employee?
A1: Use updateOne() with $set to update a specific field.

Example:

```
db.staff.updateOne(
  { name: "Jane Smith" },
  { $set: { salary: 65000 } }
);
```

Q2. How do you add a new field status to one employee?
A2: Use $set with updateOne().

Example:

```
db.staff.updateOne(
  { name: "Alice" },
  { $set: { status: "active" } }
);
```

Q3. How do you change the city of one employee?
A3: Use updateOne() with $set.

Example:

```
db.staff.updateOne(
  { name: "Bob" },
  { $set: { city: "New York" } }
);
```

---

4. UpdateMany (Update Multiple Documents)

Q1. How do you add updatedDate to all employee records?
A1: Use updateMany() with $set and new Date().

Example:

```
db.staff.updateMany(
  {},
  { $set: { updatedDate: new Date() } }
);
```

Q2. How do you increase the price of all products in a category?
A2: Use $inc in updateMany().

Example:

```
db.products.updateMany(
  { category: "Electronics" },
  { $inc: { price: 50 } }
);
```

Q3. How do you add a country field to all employees?
A3: Use $set with updateMany().

Example:

```
db.staff.updateMany(
  {},
  { $set: { country: "USA" } }
);
```

---

5. Add Date (Current Date)

Q1. How to insert a document with the current date?
A1: Use new Date() during insertion.

Example:

```
db.logs.insertOne({
  action: "Employee login",
  timestamp: new Date()
});
```

Q2. How do you update all documents with today's date?
A2: Use updateMany() with $set and new Date().

Example:

```
db.staff.updateMany(
  {},
  { $set: { lastUpdated: new Date() } }
);
```

Q3. How do you convert a numeric year field into a date field?
A3: Use $dateFromParts with updateMany().

Example:

```
db.students.updateMany(
```

```
  {},
  [
    { $set: { yearDate: { $dateFromParts: { year: "$year", month: 1, day: 1 } } } }
  ]
);
```

---

6. DeleteOne (Delete Single Document)

Q1. How do you delete a specific employee?
A1: Use deleteOne() with a filter.

Example:

```
db.staff.deleteOne({ name: "Ali" });
```

Q2. How do you delete a single product by _id?
A2: Use deleteOne() with ObjectId.

Example:

```
db.products.deleteOne({ _id: ObjectId("PUT_PRODUCT_ID_HERE") });
```

---

7. DeleteMany (Delete Multiple Documents)

Q1. How do you delete all employees from a city?
A1: Use deleteMany() with a filter.

Example:

```
db.staff.deleteMany({ city: "New York" });
```

Q2. How do you delete all student records where year < 2020?
A2: Use deleteMany() with a condition.

Example:

```
db.students.deleteMany({ year: { $lt: 2020 } });
```

---

8. Drop Operations

Q1. How do you drop a collection?
A1: Use drop() on the collection.

Example:

```
db.students.drop();
```

Q2. How do you drop the entire employees database?
A2: Use dropDatabase().

Example:

```
db.dropDatabase();
```