# NAME: Shah Siddh Tejaskumar

# REGNO: 20BCE1937

# JAVA SPRING BOOT ASSIGNMENT 2

**1st question: Create, update, and delete commands in MySQL**

CREATE TABLE test.employees (id INT PRIMARY KEY, name VARCHAR(50),salary DECIMAL(10, 2));

INSERT INTO test.employees (id, name, salary)

VALUES (1, 'John Doe', 5000.00),

(2, 'Jane Smith', 6000.00),

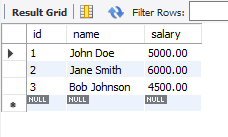
(3, 'Bob Johnson', 4500.00);

SELECT \* FROM test.employees;

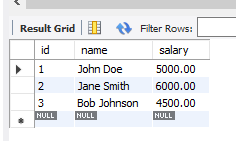
UPDATE test.employees SET salary = 5000.00 WHERE id = 1;

DELETE FROM employees WHERE salary < 4900.00;

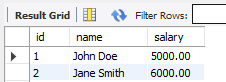
Output:



OUTPUT FOR UPDATE:



OUTPUT FOR DELETE



**2ND QUESTION : CREATE TABLE AND PERFORM JOINS ON THEM**

CREATE TABLE test.departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50)

);

CREATE TABLE test.employees (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(50),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

-- Inserting values into the "departments" table

INSERT INTO test.departments (department\_id, department\_name)

VALUES (1, 'Sales'),

(2, 'Marketing'),

(3, 'Finance');

-- Inserting values into the "employees" table

INSERT INTO test.employees (employee\_id, employee\_name, department\_id)

VALUES (1, 'DATHATREYA', 1),

(2, 'Jane Smith', 2),

(3, 'Bob Johnson', 1);

-- Selecting all rows from the "departments" table

SELECT \* FROM test.departments;

-- Selecting all rows from the "employees" table

SELECT \* FROM test.employees;

SELECT test.employees.employee\_id, test.employees.employee\_name, test.departments.department\_name

FROM test.employees

INNER JOIN departments ON test.employees.department\_id = test.departments.department\_id;

SELECT test.employees.employee\_id, test.employees.employee\_name, test.departments.department\_name

FROM test.employees

LEFT JOIN departments ON test.employees.department\_id = test.departments.department\_id;

SELECT test.employees.employee\_id, test.employees.employee\_name, test.departments.department\_name

FROM test.employees

RIGHT JOIN departments ON test.employees.department\_id = test.departments.department\_id;

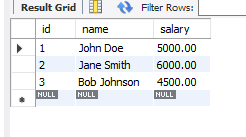
SELECT test.employees.employee\_id, test.employees.employee\_name, test.departments.department\_name

FROM test.employees

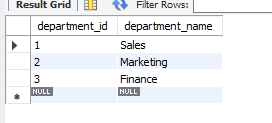
FULL JOIN departments ON test.employees.department\_id = test.departments.department\_id;

**OUTPUT:**

**EMPLOYEE TABLE:**

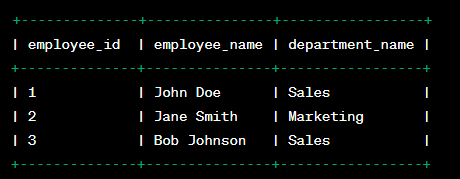


**DEPARTMENT TABLE:**

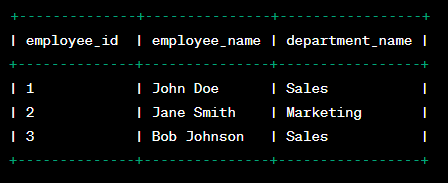


**JOINS:**

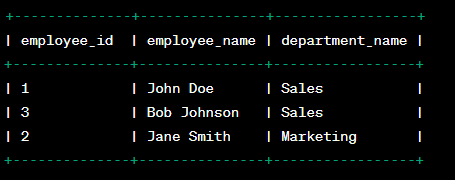
**INNER JOIN:**



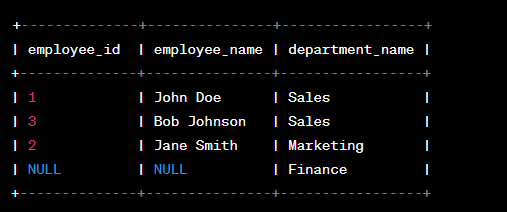
**LEFT JOIN:**



**RIGHT JOIN:**



**FULL JOIN:**



3rd QUESTION:

To perform create, update, and delete commands using the MongoDB shell (monogosh) within MongoDB Compass, follow the steps outlined below:

**Step 1: Launch the MongoDB Shell (monogosh)**

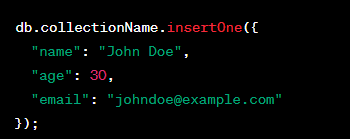
Open MongoDB Compass and connect to your MongoDB server.

Click on the "Collection" tab to select a collection.

At the top-right corner of the collection view, click on the "Open Shell" button. This will launch the MongoDB shell (monogosh).

**Step 2: Create Operation**

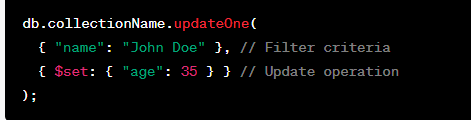
To insert a document into a collection, use the insertOne() or insertMany() methods. Here's an example using insertOne():



Replace collectionName with the actual name of your collection. This command will insert a single document with the specified fields and values into the collection.

**Step 3: Update Operation**

To update a document in a collection, use the updateOne() or updateMany() methods. Here's an example using updateOne():



Replace collectionName with the name of your collection. This command will update the first document that matches the filter criteria (name: "John Doe") and set the "age" field to 35.

**Step 4: Delete Operation**

To delete a document from a collection, use the deleteOne() or deleteMany() methods. Here's an example using deleteOne():



Replace collectionName with the name of your collection. This command will delete the first document that matches the filter criteria (name: "John Doe").

Remember to adjust the collection name and filter criteria according to your specific use case.

After executing each command, you can verify the results by viewing the updated collection data within MongoDB Compass.