ASSIGNMENT 2

Name : Nibedita Karmakar

Reg No : 20BDS0229

Technology Track : Modern Application Development (Java Spring Boot)

1. **Create, update, delete commands in mysql.**

```
CREATE TABLE students (
  id INTEGER PRIMARY KEY,
  name VARCHAR(30) NOT NULL,
  gender CHAR(1) NOT NULL
);

-- insert some values
INSERT INTO students VALUES (1, 'Ryan', 'M');
INSERT INTO students VALUES (2, 'Joanna', 'F');
INSERT INTO students VALUES (3, 'Rita', 'F');

-- fetch some values
SELECT * FROM students;

-- update
update students set name='Meetali' WHERE id=1;
SELECT * FROM students;

-- delete
DELETE FROM students where id=2;
```

```
SELECT * FROM students;
```

2. Create tables and perform joins.

```
-- Create table for school_fees
CREATE TABLE school_fees (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(50),
  fee_amount DECIMAL(10, 2),
  payment_status VARCHAR(20)
);

-- Create table for students
CREATE TABLE students (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(50),
  class VARCHAR(10),
  section VARCHAR(10),
  address VARCHAR(100)
);

-- Insert sample data into school_fees table
INSERT INTO school_fees (student_id, student_name, fee_amount, payment_status)
VALUES (1, 'John Doe', 1000.00, 'Paid'),
       (2, 'Jane Smith', 1500.00, 'Pending'),
       (3,'joe', 1001.00,'pending');
```

```
-- Insert sample data into students table
INSERT INTO students (student_id, student_name, class, section, address)
VALUES (1, 'John Doe', '10th Grade', 'A', '123 Main St'),
    (2, 'Jane Smith', '11th Grade', 'B', '456 Elm St'),
    (4,'kim','12th Grade','B','176 ememt st');
-- Perform INNER JOIN between school_fees and students table
SELECT school_fees.student_id, school_fees.student_name, school_fees.fee_amount,
students.class, students.section, students.address
FROM school_fees
INNER JOIN students ON school_fees.student_id = students.student_id;

-- Perform LEFT JOIN between school_fees and students table
SELECT school_fees.student_id, school_fees.student_name, school_fees.fee_amount,
students.class, students.section, students.address
FROM school_fees
LEFT JOIN students ON school_fees.student_id = students.student_id;

-- Perform RIGHT JOIN between school_fees and students table
SELECT school_fees.student_id, school_fees.student_name, school_fees.fee_amount,
students.class, students.section, students.address
FROM school_fees
RIGHT JOIN students ON school_fees.student_id = students.student_id;

-- Perform FULL JOIN between school_fees and students table
SELECT school_fees.student_id, school_fees.student_name, school_fees.fee_amount,
students.class, students.section, students.address
FROM school_fees
FULL JOIN students ON school_fees.student_id = students.student_id;
```

3. **Create, update and delete commands in mongodb.**

```
// Connect to MongoDB
const MongoClient = require('mongodb').MongoClient;
const uri = 'mongodb://localhost:27017';
const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });

// Create a document in the school_fees collection
const createDocument = async () => {
  try {
    await client.connect();
    const database = client.db('your_database_name');
    const collection = database.collection('school_fees');

    const newDocument = {
      student_id: 1,
      student_name: 'John Doe',
      fee_amount: 500.00,
      payment_status: 'Paid'
    };

    await collection.insertOne(newDocument);
    console.log('Document created successfully.');
  } finally {
    await client.close();
  }
};

// Update a document in the school_fees collection
const updateDocument = async () => {
```

```
  try {
    await client.connect();
    const database = client.db('your_database_name');
    const collection = database.collection('school_fees');

    const filter = { student_id: 1 };
    const update = { $set: { fee_amount: 750.00 } };

    await collection.updateOne(filter, update);
    console.log('Document updated successfully.');
  } finally {
    await client.close();
  }
};

// Delete a document from the school_fees collection
const deleteDocument = async () => {
  try {
    await client.connect();
    const database = client.db('your_database_name');
    const collection = database.collection('school_fees');

    const filter = { student_id: 1 };

    await collection.deleteOne(filter);
    console.log('Document deleted successfully.');
  } finally {
    await client.close();
  }
```

```
};

// Call the functions to perform operations
createDocument()
  .then(() => updateDocument())
  .then(() => deleteDocument())
  .catch(console.error);
```