

✓ Case Study 1: Online Course Registration System

📌 Objective:

Allow students to register/unregister for courses and view course details.

📌 Table Structure:

```
CREATE DATABASE course_db;
```

```
USE course_db;
```

```
CREATE TABLE courses (  
    course_id INT PRIMARY KEY,  
    course_name VARCHAR(100),  
    faculty VARCHAR(100),  
    credits INT  
);
```

📌 JDBC Operations:

INSERT: Add new courses.

SELECT: List available courses.

UPDATE: Modify faculty or credit values.

DELETE: Remove obsolete courses.

-----Insert Operation-----

```
Connection con = DriverManager.getConnection(  
    "jdbc:mysql://localhost:3306/course_db", "root", "password");
```

```
String query = "INSERT INTO courses VALUES (?, ?, ?, ?)";
```

```
PreparedStatement ps = con.prepareStatement(query);
```

```
ps.setInt(1, 101);
```

```
ps.setString(2, "Core Java");
```

```
ps.setString(3, "Dr. Mehta");
```

```
ps.setInt(4, 4);
```

```
ps.executeUpdate();
```

```
System.out.println("Course inserted successfully.");
```

-----Select Operation-----

```
String query = "SELECT * FROM courses";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {
    System.out.println("Course ID: " + rs.getInt("course_id"));
    System.out.println("Course Name: " + rs.getString("course_name"));
    System.out.println("Faculty: " + rs.getString("faculty"));
    System.out.println("Credits: " + rs.getInt("credits"));
    System.out.println("-----");
}
```

-----Update Operation-----

```
String query = "UPDATE courses SET faculty = ?, credits = ? WHERE course_id = ?";
PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, "Dr. Sharma");
ps.setInt(2, 5);
ps.setInt(3, 101);

ps.executeUpdate();
System.out.println(" Course updated successfully.");
```

-----Delete Operation-----

```
String query = "DELETE FROM courses WHERE course_id = ?";
PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1, 101);

ps.executeUpdate();
System.out.println(" Course deleted successfully.");
```

✓ Case Study 2: Product Inventory System

📌 Objective:

Track product stock in a retail store.

📌 Table Structure:

```
CREATE DATABASE inventory_db;  
USE inventory_db;
```

```
CREATE TABLE products (  
    product_id INT PRIMARY KEY,  
    product_name VARCHAR(100),  
    quantity INT,  
    price DECIMAL(10,2)  
);
```

📌 JDBC Operations:

INSERT: Add new products to inventory.

SELECT: View stock levels and prices.

UPDATE: Update quantity after sale/purchase.

DELETE: Remove discontinued products.

-----Insert Operation-----

```
Connection con = DriverManager.getConnection(  
    "jdbc:mysql://localhost:3306/inventory_db", "root", "password");
```

```
String query = "INSERT INTO products VALUES (?, ?, ?, ?)";  
PreparedStatement ps = con.prepareStatement(query);
```

```
ps.setInt(1, 201);  
ps.setString(2, "Laptop");  
ps.setInt(3, 10);  
ps.setDouble(4, 49999.99);
```

```
ps.executeUpdate();  
System.out.println("Product inserted successfully.");
```

-----Select Operation-----

```

String query = "SELECT * FROM products";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {
    System.out.println("Product ID: " + rs.getInt("product_id"));
    System.out.println("Product Name: " + rs.getString("product_name"));
    System.out.println("Quantity: " + rs.getInt("quantity"));
    System.out.println("Price: " + rs.getDouble("price"));
    System.out.println("-----");
}

```

-----Update Operation-----

```

String query = "UPDATE products SET quantity = ? WHERE product_id = ?";
PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1, 8); // updated after sale
ps.setInt(2, 201);

ps.executeUpdate();
System.out.println("Product quantity updated.");

```

-----Delete Operation-----

```

String query = "DELETE FROM products WHERE product_id = ?";
PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1, 201);

ps.executeUpdate();
System.out.println(" Product deleted from inventory.");

```