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);

```
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.**

```
String sql = "INSERT INTO courses (course_id, course_name, faculty, credits) VALUES (?, ?, ?, ?)";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setInt(1, 101);
stmt.setString(2, "Java Programming");
stmt.setString(3, "Dr. Smith");
stmt.setInt(4, 4);
stmt.executeUpdate();
```

- **SELECT: List available courses**

```
String sql = "SELECT * FROM courses";
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) {
    System.out.println("Course: " + rs.getString("course_name"));
}
```

UPDATE: Modify faculty or credit values.

```
String sql = "UPDATE courses SET faculty = ? WHERE course_id = ?";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setString(1, "Dr. Johnson");
stmt.setInt(2, 101);
stmt.executeUpdate();
```

- DELETE: Remove obsolete courses.

```
String sql = "DELETE FROM courses WHERE course_id = ?";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setInt(1, 101);
stmt.executeUpdate();
```

✓ Case Study 2: Product Inventory System

🎯 Objective: Track product stock in a retail store.

📊 Table Structure:C 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));

```
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.

```
String sql = "INSERT INTO products (product_id, product_name, quantity, price) VALUES (?, ?, ?, ?)";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setInt(1, 1);
stmt.setString(2, "Laptop");
stmt.setInt(3, 10);
stmt.setBigDecimal(4, new BigDecimal("749.99"));
stmt.executeUpdate();
```

- SELECT: View stock levels and prices.

```
String sql = "SELECT * FROM products";
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) {
   System.out.println("Product: " + rs.getString("product_name") + " | Price: " + rs.getBigDecimal("price"));
}
```

- UPDATE: Update quantity after sale/purchase.

```
String sql = "UPDATE products SET quantity = quantity - ? WHERE product_id = ?";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setInt(1, 1); // sale of 1 unit
stmt.setInt(2, 1);
stmt.executeUpdate();
```

- DELETE: Remove discontinued products

```
String sql = "DELETE FROM products WHERE product_id = ?";
PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setInt(1, 1);
stmt.executeUpdate();
```