



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.

```
package Assignment;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.Statement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;

import java.util.Scanner;

public class Assignment1 {

    public static void main(String[] args) {

        String url = "jdbc:mysql://localhost:3306/course_db";

        String user = "root";

        String password = "Akshay#2001";

        Scanner sc = new Scanner(System.in);

        try (Connection con = DriverManager.getConnection(url, user, password))
        {

            Statement stmt = con.createStatement();

            while (true) {

                System.out.println("\n1. Add Course\n2. View Course\n3. Update
Course\n4. Delete Course\n5. Exit");

                System.out.print("Choose option: ");

                int choice = sc.nextInt();
            }
        }
    }
}
```

```
sc.nextLine();

switch (choice) {

    case 1:

        System.out.print("Course ID: ");

        int cid = sc.nextInt();

        sc.nextLine();

        System.out.print("Course Name: ");

        String cname = sc.nextLine();

        System.out.print("Faculty: ");

        String faculty = sc.nextLine();

        System.out.print("Credits: ");

        int credits = sc.nextInt();

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

        try (PreparedStatement pstmt =
con.prepareStatement(insertQuery)) {

            pstmt.setInt(1, cid);

            pstmt.setString(2, cname);

            pstmt.setString(3, faculty);

            pstmt.setInt(4, credits);

            pstmt.executeUpdate();
        }
    }
}
```



```
break;

case 3:

    System.out.print("Enter Course ID to Update: ");

    int updateId = sc.nextInt();

    sc.nextLine();

    System.out.print("New Faculty: ");

    String newFaculty = sc.nextLine();

    System.out.print("New Credits: ");

    int newCredits = sc.nextInt();

    String updateQuery = "UPDATE courses SET faculty = ?, credits = ? WHERE course_id = ?";

    try (PreparedStatement pstmt =
con.prepareStatement(updateQuery)) {

        pstmt.setString(1, newFaculty);

        pstmt.setInt(2, newCredits);

        pstmt.setInt(3, updateId);
```

```
        pstmt.executeUpdate();

        System.out.println("Course updated!");

    }

    break;

case 4:

    System.out.print("Enter Course ID to Delete: ");

    int delId = sc.nextInt();

    String deleteQuery = "DELETE FROM courses WHERE course_id = ?";

    try (PreparedStatement pstmt =
con.prepareStatement(deleteQuery)) {

        pstmt.setInt(1, delId);

        pstmt.executeUpdate();

        System.out.println("Course deleted!");

    }

    break;
```

```
case 5:  
  
    System.out.println("Exiting...");  
  
    return;  
  
default:  
  
    System.out.println("Invalid choice!");  
  
}  
  
}  
  
}  
  
} catch (SQLException e) {  
  
    e.printStackTrace();  
  
}  
  
}  
  
}
```

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.

```
package Assignment;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
import java.sql.Statement; //  Correct import
import java.sql.SQLException;
import java.util.Scanner;
```

```
public class Assignment2 {
```

```
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/inventory_db";
        String user = "root";
```

```
String password = "Akshay#2001";

Scanner sc = new Scanner(System.in);

try (Connection con = DriverManager.getConnection(url, user, password))
{
    Statement stmt = con.createStatement(); //  No casting needed

    while (true) {

        System.out.println("\n1. Add Product\n2. View Product\n3. Update
Quantity\n4. Delete Product\n5. Exit");

        System.out.print("Choose option: ");

        int choice = sc.nextInt();

        sc.nextLine();

        switch (choice) {

            case 1:

                System.out.print("Product ID: ");

                int pid = sc.nextInt();

                sc.nextLine();

                System.out.print("Product Name: ");

                String pname = sc.nextLine();

                System.out.print("Quantity: ");

                int qty = sc.nextInt();

                System.out.print("Price: ");

                double price = sc.nextDouble();

                String insertQuery = "INSERT INTO products VALUES (?, ?, ?, ?)";


```

```
try (PreparedStatement pstmt =  
con.prepareStatement(insertQuery)) {  
  
    pstmt.setInt(1, pid);  
  
    pstmt.setString(2, pname);  
  
    pstmt.setInt(3, qty);  
  
    pstmt.setDouble(4, price);  
  
    pstmt.executeUpdate();  
  
    System.out.println("Product added!");  
  
}  
  
break;
```

case 2:

```
ResultSet rs = stmt.executeQuery("SELECT * FROM products");  
  
while (rs.next()) {  
  
    System.out.println(rs.getInt("product_id") + " | " +  
        rs.getString("product_name") + " | " +  
        rs.getInt("quantity") + " | " +  
        rs.getDouble("price"));  
  
}  
  
break;
```

case 3:

```
System.out.print("Enter Product ID to Update: ");  
  
int updateId = sc.nextInt();  
  
System.out.print("New Quantity: ");  
  
int newQty = sc.nextInt();
```

```
String updateQuery = "UPDATE products SET quantity = ? WHERE
product_id = ?";

try (PreparedStatement pstmt =
con.prepareStatement(updateQuery)) {

    pstmt.setInt(1, newQty);

    pstmt.setInt(2, updateId);

    pstmt.executeUpdate();

    System.out.println("Quantity updated!");

}

break;
```

case 4:

```
System.out.print("Enter Product ID to Delete: ");

int delId = sc.nextInt();

String deleteQuery = "DELETE FROM products WHERE
product_id = ?";

try (PreparedStatement pstmt =
con.prepareStatement(deleteQuery)) {

    pstmt.setInt(1, delId);

    pstmt.executeUpdate();

    System.out.println("Product deleted!");

}

break;
```

case 5:

```
System.out.println("Exiting...");

return;
```

default:

```
System.out.println("Invalid choice!");  
}  
}  
  
}  
}  
  
} catch (SQLException e) {  
    e.printStackTrace();  
}  
}  
}
```