Java - JDBC

Nội dung

1. Java-JDBC là gì
2. Lớp Connection
3. Lớp Statement, PreparedStatement
4. Lớp ResultSet

Java – JDBC là gì?

Ngôn ngữ SQL là gì ?

- JDBC - Java Database Connectivity
- Là thư viện tiêu chuẩn dùng để tương tác với các loại CSDL quan hệ.
- Là một tập hợp các class và interface dùng cho ứng dụng Java nói chuyện với các CSDL

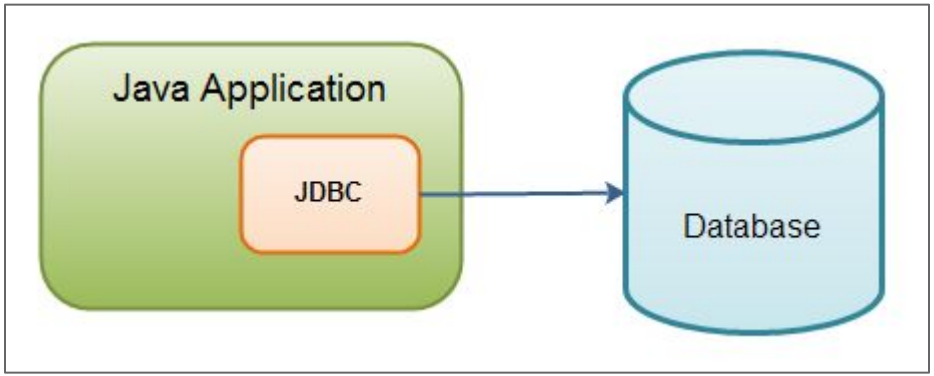
Ngôn ngữ SQL là gì ?

Các thành phần cơ bản bao gồm

- DriverManager
- Driver
- Connection
- Statement
- ResultSet

Nguyên tắc kết nối

- Để kết nối với CSDL nào cần có Driver của CSDL đó.
- Ví dụ:
  - Kết nối Oracle cần Driver của Oracle
  - Kết nối MySQL cần Driver của MySql



Lớp Connection

Lớp Connection

- Tạo ra phiên làm việc giữa Java và CSDL
- Dùng để tạo ra Statement, PreparedStatement,...
- Ngoài ra còn cung cấp nhiều phương thức quản lý transaction

Lớp Connection

Các phương thức cơ bản

createStatement	Tạo ra 1 đối tượng Statement
prepareStatment	Tạo ra 1 đối tượng prepareStatment
createsetAutoCommit	Được sử dụng để thiết lập trạng thái commit
commit	Lưu các thay đổi được thực hiện trước đó
rollback	Loại bỏ tất cả các thay đổi trước đó
close	Đóng kết nối và giải phóng tài nguyên

Tạo ra Connection với MySQL

- **Tải Driver**
  - *Driver của MySQL đã có sẵn trong Netbean nên sẽ không cần tải.*
- **Nạp thư viện sử dụng “Class.forName”**
- **Chuẩn bị chuỗi kết nối**
- **Tạo kết nối qua lớp “DriverManager”**

Tạo ra Connection với MySQL

```
Class.forName("com.mysql.jdbc.Driver");
String connStr = "jdbc:mysql://localhost:3306/mydb";
Connection conn =
DriverManager.getConnection(connStr, "root", "123");
```

- **Connection String** :

  jdbc:mysql://server_name:port/db_name

Lớp Statement & PreparedStatement

Lớp Statement

- Là một **INTERFACE**
- Được tạo ra từ thể hiện của đối tượng **Connection**
- Cung cấp các phương thức để thực thi các câu lệnh truy vấn với CSDL
- Cung cấp phương thức để tạo ra đối tượng ResultSet

Lớp Statement

Các phương thức cơ bản

`ResulteSet executeQuery(String sql)`	Thực hiện câu lệnh SELECT
`Int executeUpdate(String sql)`	Thực hiện các câu lệnh insert, update, delete
`Boolean execute(String sql)`	Thực thi các câu truy vấn trả về nhiều kết quả
`Int[] executeBatch();`	Thực thi 1 tập lệnh

Câu lệnh Statement

- Ví dụ: **Thực thi câu Select**

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    String connStr = "jdbc:mysql://localhost:3306/omsdb";
    String user = "root";
    String pass = "root";
    Connection conn = DriverManager.getConnection(connStr, user, pass);
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM Users");
} catch (Exception e) {
    e.printStackTrace();
}
```

Lớp PreparedStatement

- Là một **SUB-INTERFACE** của **Statement**
- Được sử dụng để thực hiện các truy vấn có tham số.
- Được tạo ra từ thể hiện của **Connection**
- Hỗ trợ cho việc phòng tránh SQLInjection

Lớp PreparedStatement

Các phương thức cơ bản

`ResulteSet executeQuery()`	Thực hiện câu lệnh SELECT
`Int executeUpdate()`	Thực hiện các câu lệnh insert, update, delete
`setInt(int index,int value)`	Đặt giá trị int cho tham số
`setString(int index, String value)`	Đặt giá trị string cho tham số
`setFloat(int index, float value)`	Đặt giá trị float cho tham số
`setDouble(int index, double value)`	Đặt giá trị double cho tham số

Câu lệnh PreparedStatement

- Ví dụ: **Thực thi câu Insert**

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    String connStr = "jdbc:mysql://localhost:3306/omsdb";
    String user = "root";
    String pass = "root";
    Connection conn = DriverManager.getConnection(connStr, user, pass);
    PreparedStatement pstmt =
        conn.prepareStatement("INSERT INTO Users (name, email) VALUES (?,?)");
    pstmt.setString(1, "hoang");
    pstmt.setString(2, "hoang@gmail.com");
    int executeUpdate = pstmt.executeUpdate();
} catch (Exception e) {
    e.printStackTrace();
}
```

Lớp ResultSet

Lớp ResultSet

- Cho phép duyệt kết quả trả về từ các câu truy vấn SELECT
- Duy trì một con trỏ trỏ đến một hàng của một bảng
- Ban đầu con trỏ trỏ đến hàng đầu tiên
- Có thể điều khiển con trỏ đi theo hướng tiếp theo hoặc quay lại bản ghi phía trước.

Lớp ResultSet

Các phương thức cơ bản

`Int getInt(int index)`	Lấy gia trị kiểu INT ở cột thứ index
`Int getInt(String columnName)`	Lấy giá trị kiểu INT ở cột có tên chỉ định
`String getString(int index)`	Lấy giá trị kiểu String ở cột thứ index
`String getString(String columnName)`	Lấy giá trị kiểu String ở cột có tên chỉ định
`Boolean next()`	Di chuyển con trỏ đến hàng tiếp theo nếu có trả về TRUE nếu không có trả về FALSE

Lớp ResultSet

- Ví dụ:

```
PreparedStatement pstmt = conn.prepareStatement("select * from users");
ResultSet rs = pstmt.executeQuery();

while (rs.next()) {
    System.out.println(rs.getString("name"));
    System.out.println(rs.getString("email"));
    System.out.println(rs.getString("password"));
    System.out.println(rs.getString("phone"));
    System.out.println("==================================");
}
```