

Bài 07: Java Swing and MySQL

Giảng Viên: ThS. Giang Hào Côn

7.1/ Giới thiệu

- **MySQL** là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến.
- **Java** là ngôn ngữ lập trình hướng đối tượng mạnh mẽ, được sử dụng rộng rãi.
- Kết hợp **MySQL** và **Java** giúp xây dựng ứng dụng web mạnh mẽ, hiệu quả. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu MySQL trong Java.

7.2/ MySQL

- a) Giới thiệu
- b) Điểm nổi bật của MySQL
- c) Ứng dụng của MySQL
- d) Cấu trúc cơ bản, các kiểu dữ liệu và Các câu lệnh SQL cơ bản:
SELECT, INSERT, UPDATE, DELETE.

7.3/ Kết nối Java với Mysql

a) Thư viện JDBC

JDBC là viết tắt của Java Database Connectivity, là một API (Application Programming Interface) tiêu chuẩn được sử dụng để kết nối và truy cập dữ liệu từ các cơ sở dữ liệu khác nhau trong các ứng dụng Java.

b) Chức năng của JDBC

c) Lợi ích

7.3/ Kết nối Java với Mysql

c) Thành phần:

- **DriverManager**: Lớp quản lý danh sách các trình điều khiển JDBC (JDBC driver) cho các loại cơ sở dữ liệu khác nhau.
- **Connection**: Giao diện đại diện cho một kết nối với cơ sở dữ liệu.
- **Statement**: Giao diện để thực thi các câu lệnh SQL.
- **ResultSet**: Lớp lưu trữ kết quả của một truy vấn SQL.

7.3/ Kết nối Java với Mysql

d) Cách sử dụng:

- 1) Tải xuống trình điều khiển JDBC
- 2) Thêm trình điều khiển JDBC vào project
- 3) Tạo kết nối với cơ sở dữ liệu
- 4) Thực thi câu lệnh SQL
- 5) Xử lý kết quả:
- 6) Đóng kết nối

7.3/ Kết nối đến CSDL MySQL

```
import java.sql.*;

public class Connect {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.jdbc.Driver");
        Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/database_name",
                                                            "username", "password");

        Statement statement = connection.createStatement();

        // Thực thi các truy vấn SQL

        statement.close();
        connection.close();
    }
}
```

7.3/ Kết nối Java với Mysql

e) Ví dụ








```
import java.sql.*;

public class Vidu {

    public static void main(String[] args) throws ClassNotFoundException, SQLException {
        // Tải xuống và thêm trình điều khiển JDBC MySQL vào project
        // Tạo kết nối với cơ sở dữ liệu MySQL
        Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/testdb", "root", "");

        // Thực thi câu lệnh SELECT
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery("SELECT * FROM student");
        // Xử lý kết quả
        while (resultSet.next()) {
            System.out.println(resultSet.getString("name"));
        }

        // Đóng kết nối
        resultSet.close();
        statement.close();
        connection.close();
    }
}
```

- ✓  Referenced Libraries
- >  jgoodies-forms-1.8.0.jar
 - >  mysql-connector-j-8.3.0.jar
 - >  Hình
 - >  images
 - ✓  lib
 -  mysql-connector-j-8.3.0.jar

7.3/ Thêm dữ liệu vào bảng

```
String sql = "INSERT INTO table_name (column1, column2, ...) VALUES (?, ?, ...)";  
PreparedStatement ps = connection.prepareStatement(sql);  
  
ps.setString(1, value1);  
ps.setString(2, value2);  
// ...  
ps.executeUpdate();
```

7.3/ Đọc dữ liệu của bảng

```
String sql = "SELECT * FROM table_name";  
Statement statement = connection.createStatement();  
ResultSet resultSet = statement.executeQuery(sql);  
  
while (resultSet.next()) {  
    // Lấy dữ liệu từ resultSet  
}  
  
resultSet.close();  
statement.close();
```

7.3/ cập nhật dữ liệu của bảng

```
String sql = "UPDATE table_name SET column1 = ?, column2 = ?, ... WHERE id = ?";  
PreparedStatement ps = connection.prepareStatement(sql);
```

```
ps.setString(1, value1);
```

```
ps.setString(2, value2);
```

```
// ...
```

```
ps.setInt(n, id);
```

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

7.3/ xoá dữ liệu của bảng

```
String sql = "DELETE FROM table_name WHERE id = ?";  
PreparedStatement ps = connection.prepareStatement(sql);  
  
ps.setInt(1, id);  
  
ps.executeUpdate();
```

7.4/ Demo mẫu

Tạo ứng dụng quản lý sinh viên đơn giản với các chức năng:

- Thêm mới sinh viên.
- Sửa thông tin sinh viên.
- Xóa sinh viên.
- Tìm kiếm sinh viên.

Tạo ứng dụng đăng nhập với chức năng:

- Kiểm tra tài khoản hợp lệ.
- Hiển thị thông báo đăng nhập thành công hoặc thất bại.

7.4/ Demo mẫu

- Bước 01: Tạo một Project tên là LapTrinhCSDLJava.
- Bước 02: Tạo Package tên là JavaMySQL
- Bước 03: Tạo lớp tên là DataProvider và code như sau

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DataProvider {
    //Khai báo tên db cần làm việc
    private final static String DATABASE_LINK = "jdbc:mysql://localhost:3306/qlsv";

    public static Connection ketNoi() throws ClassNotFoundException
    {
        //Khai báo đối tượng kết nối
        Connection conn = null;
        try {
            //Thực hiện kết nối đến db
            conn = DriverManager.getConnection(DATABASE_LINK, "root", "");
        } catch (SQLException ex) {
            System.err.println("Không kết nối được đến MySQL. Chi tiết: " + ex.getMessage());
        }
        //Trả về kết nối
        return conn;
    }
}
```

7.4/ Demo mẫu

- Bước 04: Tạo lớp tên là SinhVien và code như sau

```
import java.util.Date;

public class SinhVien {
    private String maSV = "";
    private String hoTen;
    private String dienThoai;
    private String email;
    private String diaChi;
    private String maKhoa;
    private int gioiTinh;
    private Date ngaySinh;

    public SinhVien(String maSV, String hoTen, String dienThoai, String email, String diaChi)
    {
        this.maSV = maSV;
        this.hoTen = hoTen;
        this.dienThoai = dienThoai;
        this.email = email;
        this.diaChi = diaChi;
    }
}
```

Khai báo tiếp các getter/setter
cho các thuộc tính của lớp này

7.4/ Demo mẫu

- Bước 04: Tạo lớp SinhVienBusiness.java và code như sau:

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class SinhVienBussiness {

    public List<SinhVien> layDanhSach() throws ClassNotFoundException {
        //Khai báo 1 danh sách
        List<SinhVien> lstSinhVien = new ArrayList();
    }
}
```


7.4/ Demo mẫu

- Bước 04: Tạo lớp SinhVienBusiness.java và code như sau:

```
//Khai báo kết nối
Connection conn = null;
try {
    conn = DataProvider.ketNoi();
    //Khai báo 1 công việc
    String strSQL = "Select MaSV, HoTen, DienThoai, Email, DiaChi, MaKhoa from SinhVien";

    Statement comm = conn.createStatement();

    //Thực hiện và trả về kết quả
    ResultSet rs = comm.executeQuery(strSQL);
```

7.4/ Demo mẫu

- Bước 04: Tạo lớp SinhVienBusiness.java và code như sau:

```
//Khai báo đối tượng
SinhVien objSV = null;
//Đọc từng dòng thông tin
while (rs.next()) {
    //Khởi tạo đối tượng
    objSV = new SinhVien();
    //Gán giá trị cho các thuộc tính
    objSV.setMaSV(rs.getString("MaSV"));
    objSV.setHoTen(rs.getString("HoTen"));
    objSV.setDienThoai(rs.getString("DienThoai"));
    objSV.setEmail(rs.getString("Email"));
    objSV.setMaKhoa(rs.getString("MaKhoa"));
    //Thêm vào danh sách
    lstSinhVien.add(objSV);
}

} catch (SQLException ex) {
    System.out.println("Có lỗi xảy ra trong quá trình làm việc với mysql. " + "Chi tiết: "
        + ex.getMessage());
}
```

7.4/ Demo mẫu

- Bước 04: Tạo lớp SinhVienBusiness.java và code như sau:

```
} finally {  
    try {  
        //Đóng kết nối  
        if (conn != null) {  
            conn.close();  
        }  
    } catch (SQLException ex) {  
        System.out.println("abc");  
    }  
}  
  
return lstSinhVien;  
}  
  
}
```

7.4/ Demo mẫu

- Bước 05: Tạo lớp frmDanhSachSV.java và code như sau:

```
private JFrame frmDSSV;  
private JTextField txtTuKhoa;  
private JTable tbDSSV;  
  
/**  
 * Launch the application.  
 */  
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                frmDanhSachSinhVien window = new frmDanhSachSinhVien();  
                window.frmDSSV.setVisible(true);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    });  
}
```

7.4/ Demo mẫu

- Bước 05: Tạo lớp frmDanhSachSV.java và code như sau:

```
private void hienThiDanhSachSinhVien()  
{  
    //Khái báo 1 đối tượng thuộc lớp SinhVienBusiness  
    SinhVienBussiness sinhVienBusiness = new SinhVienBussiness();  
  
    //Lấy danh sách sinh viên từ db của mysql  
    List<SinhVien> lstSinhVien = new ArrayList<>();  
  
    try {  
        lstSinhVien = sinhVienBusiness.layDanhSach();  
    } catch (ClassNotFoundException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
  
    //Khái báo các cột tiêu đề trên jTable  
    String colTieuDe[] = {"Mã SV", "Họ tên", "Điện thoại", "Email", "Địa chỉ"};  
  
    //Khái báo 1 đối tượng để đưa dữ liệu vào jTable  
    DefaultTableModel model = new DefaultTableModel(colTieuDe, 0);
```

7.4/ Demo mẫu

- Bước 05: Tạo lớp frmDanhSachSV.java và code như sau:

```
Object[] row;
for(SinhVien sv : lstSinhVien)
{
    row = new Object[5];
    row[0] = sv.getMaSV();
    row[1] = sv.getHoTen();
    row[2] = sv.getDienThoai();
    row[3] = sv.getEmail();
    row[4] = sv.getDiaChi();
    //Thêm vào model
    model.addRow(row);
}

tbDSSV.setModel(model);
tbDSSV.setSize(500, 0);
}
```

Chạy chương trình thì xuất hiện



Mã SV	Họ tên	Điện thoại	Email	Địa chỉ
1	Giang Vinh Kiệt	0907079837	giangvinhkiem@...	Hà Nội
2	Võ Phi Long	090789678	long@gmail.co...	Hà Nội

Câu hỏi thảo luận

- 1) MySQL là gì ? Trình bày điểm nổi bật, ứng dụng và các câu lệnh thông dụng của MySQL ?
- 2) JDBC là gì ? Trình bày chức năng, lợi ích, các thành phần và cách sử dụng JDBC trong ứng dụng Java Swing ?