

exemple complet CRUD Java Swing + JDBC + MySQL

- ✓ CRUD complet : Ajouter, Modifier, Supprimer
- ✓ Table interactive
- ✓ JDBC propre avec DAO
- ✓ Swing simple et fonctionnel

Architecture MVC model/view/controleur

Code sql création table user

```
CREATE DATABASE testdb;
USE testdb;
CREATE TABLE user(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100),
    email VARCHAR(150)
);
```

Architecture du projet :

1. **Database.java** → connexion JDBC
2. **User.java** → modèle
3. **UserDAO.java** → CRUD (Insert, Update, Delete, List)
4. **UserForm.java** → interface Swing (table + formulaire + boutons)

Classe pour la connexion MySQL.

1. FICHIER Database.java

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

public class Database {

    private static final String URL = "jdbc:mysql://localhost:3306/testdb?useSSL=false";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static Connection getConnection() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Classe modèle User.

2. FICHIER User.java

```
public class User {  
    private int id;  
    private String nom;  
    private String email;  
  
    public User() {}  
    public User(int id, String nom, String email) {  
        this.id = id;  
        this.nom = nom;  
        this.email = email;  
    }  
  
    public int getId() { return id; }  
    public String getNom() { return nom; }  
    public String getEmail() { return email; }  
  
    public void setId(int id) { this.id = id; }  
    public void setNom(String nom) { this.nom = nom; }  
    public void setEmail(String email) { this.email = email; }  
}
```

Contrôleur, contient toutes les opérations CRUD.

3. FICHIER UserDao.java

```
import java.sql.*;  
import java.util.*;  
  
public class UserDao {  
  
    public List<User> getAllUsers() {  
        List<User> list = new ArrayList<>();  
        String sql = "SELECT * FROM user";  
  
        try (Connection conn = Database.getConnection());  
            Statement st = conn.createStatement();  
            ResultSet rs = st.executeQuery(sql)) {  
  
                while (rs.next()) {  
                    list.add(new User(  
                        rs.getInt("id"),  
                        rs.getString("nom"),  
                        rs.getString("email"))  
                );  
            }  
        }  
    }  
}
```

```
    }

} catch (Exception e) { e.printStackTrace(); }

return list;
}

public boolean insertUser(User u) {
    String sql = "INSERT INTO user(nom, email) VALUES (?, ?)";

    try (Connection conn = Database.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql)) {

        ps.setString(1, u.getNom());
        ps.setString(2, u.getEmail());
        return ps.executeUpdate() > 0;

    } catch (Exception e) { e.printStackTrace(); return false; }
}

public boolean updateUser(User u) {
    String sql = "UPDATE user SET nom=?, email=? WHERE id=?";

    try (Connection conn = Database.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql)) {

        ps.setString(1, u.getNom());
        ps.setString(2, u.getEmail());
        ps.setInt(3, u.getId());
        return ps.executeUpdate() > 0;

    } catch (Exception e) { e.printStackTrace(); return false; }
}

public boolean deleteUser(int id) {
    String sql = "DELETE FROM user WHERE id=?";

    try (Connection conn = Database.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql)) {

        ps.setInt(1, id);
        return ps.executeUpdate() > 0;

    } catch (Exception e) { e.printStackTrace(); return false; }
}
```

Interface Swing complète.

4. FICHIER UserForm.java

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;

public class UserForm extends JFrame {

    private JTextField txtNom, txtEmail;
    private JTable table;
    private DefaultTableModel model;

    private UserDAO dao = new UserDAO();
    private int selectedId = -1;

    public UserForm() {
        setTitle("Gestion des Utilisateurs - CRUD Swing/JDBC");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // ----- FORMULAIRE -----
        JPanel form = new JPanel(new GridLayout(3, 2, 5, 5));
        form.add(new JLabel("Nom :"));
        txtNom = new JTextField();
        form.add(txtNom);

        form.add(new JLabel("Email :"));
        txtEmail = new JTextField();
        form.add(txtEmail);

        JButton btnAdd = new JButton("Ajouter");
        JButton btnUpdate = new JButton("Modifier");
        JButton btnDelete = new JButton("Supprimer");

        form.add(btnAdd);
        form.add(btnUpdate);
        form.add(btnDelete);

        // ----- TABLEAU -----
        model = new DefaultTableModel(new String[]{"ID", "Nom", "Email"}, 0);
        table = new JTable(model);
        loadUsers();

        // Ajouter Listener pour sélectionner une ligne
        table.addMouseListener(new MouseAdapter() {
```

```

public void mouseClicked(MouseEvent e) {
    int row = table.getSelectedRow();
    selectedId = Integer.parseInt(model.getValueAt(row, 0).toString());
    txtNom.setText(model.getValueAt(row, 1).toString());
    txtEmail.setText(model.getValueAt(row, 2).toString());
}
});

// ----- ACTIONS BOUTONS -----
btnAdd.addActionListener(e -> {
    User u = new User(0, txtNom.getText(), txtEmail.getText());
    if (dao.insertUser(u)) {
        JOptionPane.showMessageDialog(null, "Ajout réussi !");
        loadUsers();
    }
});
}

btnUpdate.addActionListener(e -> {
    if (selectedId != -1) {
        User u = new User(selectedId, txtNom.getText(), txtEmail.getText());
        if (dao.updateUser(u)) {
            JOptionPane.showMessageDialog(null, "Modification réussie !");
            loadUsers();
        }
    }
});
}

btnDelete.addActionListener(e -> {
    if (selectedId != -1) {
        if (dao.deleteUser(selectedId)) {
            JOptionPane.showMessageDialog(null, "Suppression réussie !");
            loadUsers();
        }
    }
});
}

// ----- ASSEMBLAGE -----
add(form, BorderLayout.NORTH);
add(new JScrollPane(table), BorderLayout.CENTER);
}

private void loadUsers() {
    model.setRowCount(0);
    for (User u : dao.getAllUsers()) {
        model.addRow(new Object[]{u.getId(), u.getNom(), u.getEmail()});
    }
}

```

```
public static void main(String[] args) {  
    new UserForm().setVisible(true);  
}  
}
```