JDBC Ứng dụng: Quản lý nhân viên

```
======Menu======
1.Get Employee by ID
2.Add Employee
3.Update Employee
4.Delete Employee
5.Get All Employee
6.Exit
```

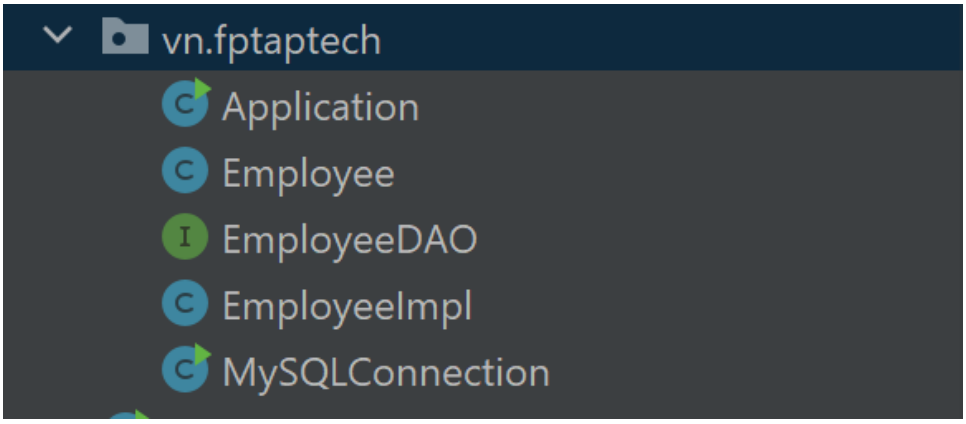
Create: Thêm nhân viên mới

Edit: Sửa bản ghi

Delete: Xóa bản ghi

Read: Đọc toàn bộ bản ghỉ OR đọc theo ID

Cấu trúc project



(File MySQLConnection là file kết nói với CSDL các bạn đã làm rồi, chỉ cần bổ sung thêm method tạo bảo cho project này)

1. Class **MySQLConnection.java** có bổ sung thêm method tạo bảo Employee(id varchar, name varchar, salary varchar)

```java
public static void createEmployeeTable() throws SQLException{
    //1. Tạo connection
    Connection connection = getMySQLConnection();
    Statement statement = connection.createStatement();
    //Thực thi các chỉ thị với Database
    //CREATE TABLE
    statement.execute( sql: "DROP TABLE IF EXISTS employee ");
    statement.execute( sql: "CREATE TABLE employee(id varchar(10) primary  key," +
            "name varchar(20)," +
            "salary varchar(20))");


}
```

2. Class **Employee.java**

Có 3 fields ID, name, salary tương đương với 3 columns trong bảng employee. Có constructor và getter/setter đầy đủ của các fields này.

```java
package vn.fptaptech;
//Employee Entity aka Employee Table
public class Employee {
    private String ID;
    private String name;
    private String salary;

    public Employee() {
    }

    public Employee(String ID, String name, String salary) {...}

    public String getID() { return ID; }

    public void setID(String ID) { this.ID = ID; }
```

3. Interface: **Employee.java**

Định nghĩa các method sẽ được triển khai trong project

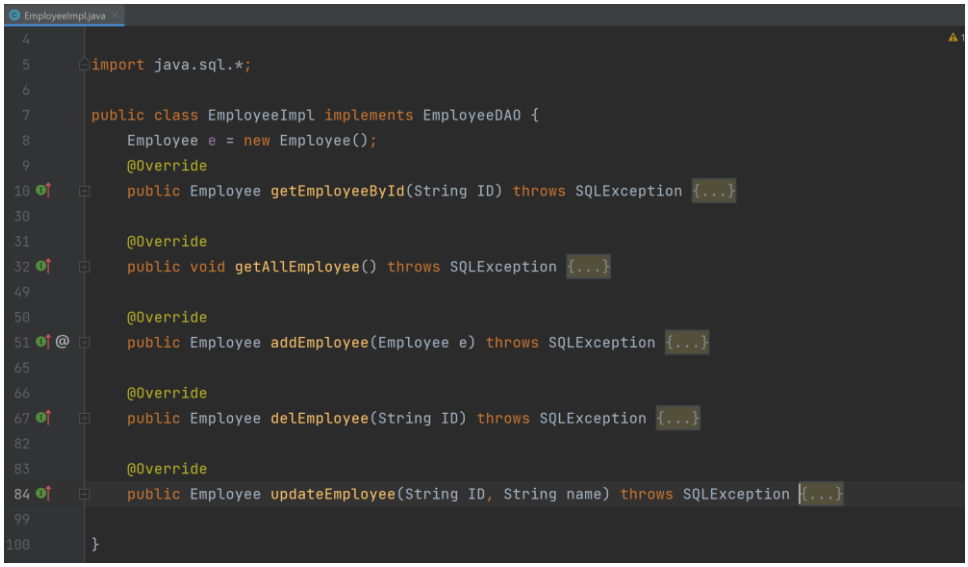
```java
import java.sql.SQLException;
import java.util.ArrayList;

//define all the functions
public interface EmployeeDAO {
    //return 1 employee
    public Employee getEmployeeById(String ID) throws SQLException;
    //Return 1 list of employees
    //public ArrayList<Employee> getAllEmployee();
    public void getAllEmployee() throws SQLException;
    public Employee addEmployee(Employee e) throws SQLException;
    public Employee delEmployee(String ID) throws SQLException;
    public Employee updateEmployee(String ID, String name) throws SQLException;

}
```

4. **Class EmployeeImpl.java** là class sẽ triển khai code biz chính

Đây là hình ảnh tổng quát toàn bộ các methods trong class EmployeeImpl.java implements các methods trong interface (chi tiết từng method sẽ có trong tường ảnh riêng)



- Chi tiết phần code sẽ triển khai trong class **EmployeeImpl**

EmployeeImpl.java

```java

import java.sql.*;

public class EmployeeImpl implements EmployeeDAO {
    Employee e = new Employee();
    @Override
    public Employee getEmployeeById(String ID) throws SQLException {
        Connection conn = MySQLConnection.getMySQLConnection();
        String query  = "select * from employee where id = "+ID;
        Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(query);
        while (rs.next()){
           String id = rs.getString( columnIndex: 1);
           e.setID(id);//id của Employee sẽ có giá trị
            String name = rs.getString( columnIndex: 2);
            e.setName(name);
            String sal = rs.getString( columnIndex: 3);
            e.setSalary(sal);
            System.out.println("      Employee Data      ");
            System.out.println("=========================");
            System.out.println("ID: "+id);
            System.out.println("Name: "+name);
            System.out.println("Salary: "+sal);
        }
        return e;
    }

    @Override
    public void getAllEmployee() throws SQLException {
        String query = "select * from Employee";
        Connection conn = MySQLConnection.getMySQLConnection();
        Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(query);

        while(rs.next())
        {
            String id = rs.getString( columnIndex: 1);
            e.setID(id);
            String name = rs.getString( columnIndex: 2);
            e.setName(name);
            String sal = rs.getString( columnIndex: 3);
            e.setSalary(sal);
            System.out.printf("|%5s|%-10s|%-10s|\n", id, name, sal);
        }
    }
    @Override
    public Employee addEmployee(Employee e) throws SQLException {
        Connection conn = MySQLConnection.getMySQLConnection();
        PreparedStatement ps = conn.prepareStatement( sql: "INSERT INTO Employee VALUES (?, ?, ?)");
        ps.setString( parameterIndex: 1, e.getID());
        ps.setString( parameterIndex: 2, e.getName());
        ps.setString( parameterIndex: 3, e.getSalary());
        int newCount =  ps.executeUpdate();
        if(newCount > 0) {
            System.out.println("Employee added successfully...!");
        }
        else
            System.out.println("Failed to add a record........!");
        return null;
    }

```

```java
    @Override
    public Employee delEmployee(String ID) throws SQLException {
        Connection conn = MySQLConnection.getMySQLConnection();
        String sql = "DELETE FROM Employee WHERE ID=?";
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString( parameterIndex: 1, ID);

        int i = ps.executeUpdate();
        if (i > 0) {
            System.out.println("Data deleted successfully!");
        }
        else
            System.out.println("Failed to delete data..!");

        return null;
    }

    @Override
    public Employee updateEmployee(String ID, String name) throws SQLException {
        Connection conn = MySQLConnection.getMySQLConnection();
        String sql = "UPDATE Employee SET Name= ? WHERE ID=?";
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString( parameterIndex: 1, name);
        ps.setString( parameterIndex: 2, ID);

        int i = ps.executeUpdate();
        if (i > 0) {
            System.out.println("An existing user was updated successfully!");
        }
        else
            System.out.println("Failed to update data..!");
        return null;
    }

}
```

5. **Class Application.java** chương trình chính để start project

Application.java

```java
package vn.fptaptech;
import java.sql.SQLException;
import java.util.Scanner;
import static java.lang.System.exit;
public class Application {
    public static void main(String[] args) throws SQLException {
        int choice,num;
        EmployeeImpl dao = new EmployeeImpl();
        Employee e;//default null
        do{
            System.out.println("======Menu======");
            System.out.println("1.Get Employee by ID");
            System.out.println("2.Add Employee");
            System.out.println("3.Update Employee");
            System.out.println("4.Delete Employee");
            System.out.println("5.Get All Employee");
            System.out.println("6.Exit");
            Scanner sc = new Scanner(System.in);
            choice = sc.nextInt();
            switch (choice){
                case 1: System.out.println("Enter the ID of Employee: ");
                    String id1 = sc.next();
                    e = dao.getEmployeeById(id1);
                    break;

                case 2: System.out.println("Enter Employee Details: ");
                    System.out.println("Enter Employee ID: ");
                    String id2 = sc.next();
                    System.out.println("Enter Employee Name: ");
                    String name = sc.next();
                    System.out.println("Enter Salary: ");
                    String salary = sc.next();
                    Employee e1 = new Employee(id2, name, salary);
                    e = dao.addEmployee(e1);
                    break;

                case 3: System.out.println("Enter the ID to change: ");
                    String id3 = sc.next();
                    System.out.println("Enter the new Name: ");
                    String name1 = sc.next();
                    e = dao.updateEmployee(id3, name1);
                    break;
                case 4: System.out.println("Enter the ID to delete: ");
                    String id4 = sc.next();
                    e = dao.delEmployee(id4);
                    break;
                case 5: System.out.println("Employee Data");
                    System.out.println("----------------------------");
                    System.out.printf("|%5s|%-10s|%-10s|\n", "ID", "Name", "Salary");
                    System.out.println("----------------------------");
                    dao.getAllEmployee();
                    System.out.println("----------------------------");
                    break;
                case 6:
                    exit( status: 0);
                default:
                    System.out.println("Wrong input..!");
                    break;
            }
            System.out.println("Enter 6 to continue..!");
            num = sc.nextInt();
        }while(num == 6);
    }
}

```

==========End of Tutorial==========