Dao

package com.morningstar.dao;

import java.sql.SQLException;

import java.util.List;

import com.morningstar.model.Employee;

public interface EmployeeDao {

public int AddEmployee(Employee employee) throws SQLException;

public List<Employee> ReadAllEmployee() throws SQLException;

public Employee FindById(int id) throws SQLException;

public Employee FindByName(String name) throws SQLException;

public int DeleteEmployee(int id) throws SQLException;

public int UpdateEmployee(Employee employee) throws SQLException;

}

================================================================================

DaoImpl

package com.morningstar.dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.morningstar.model.Employee;

public class EmployeeDaoImpl implements EmployeeDao {

private static Connection connection = null;

static {

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

connection.setAutoCommit(false);

} catch (SQLException | ClassNotFoundException e) {

System.out.println("There is some problem");

e.printStackTrace(); // log the stack trace

}

}

public static Connection getConnection() {

return connection;

}

public static void setConnection(Connection connection) {

EmployeeDaoImpl.connection = connection;

}

@Override

public int AddEmployee(Employee employee) throws SQLException {

String sql = "Insert Into Employee( EMPLOYEE\_ID, EMPLOYEE\_NAME,EMPLOYEE\_SALARY,DEPARTMENT\_ID) Values(?,?,?,?)";

PreparedStatement pstmt = connection.prepareStatement(sql);

pstmt.setInt(1, employee.getEmployeeId());

pstmt.setString(2, employee.getEmployeeName());

pstmt.setDouble(3, employee.getEmployeeSalary());

pstmt.setInt(4, employee.getDepartmentId());

int result = pstmt.executeUpdate();

return result;

}

@Override

public List<Employee> ReadAllEmployee() throws SQLException {

String sql="Select \* from Employee";

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

List<Employee> list=new ArrayList<>();

while (rs.next())

{

int empId=rs.getInt("EMPLOYEE\_ID");

String empName=rs.getString("EMPLOYEE\_NAME");

double empSal=rs.getDouble("EMPLOYEE\_SALARY");

int empDept=rs.getInt("DEPARTMENT\_ID");

Employee employee=new Employee(empId,empName,empSal,empDept);

list.add(employee);

}

return list;

}

@Override

public Employee FindById(int id) throws SQLException {

String sql="Select \* from Employee Where Employee\_Id="+id;

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

Employee employee=null;

while (rs.next())

{

int empId=rs.getInt("EMPLOYEE\_ID");

String empName=rs.getString("EMPLOYEE\_NAME");

double empSal=rs.getDouble("EMPLOYEE\_SALARY");

int empDept=rs.getInt("DEPARTMENT\_ID");

employee=new Employee(empId,empName,empSal,empDept);

}

return employee;

}

@Override

public Employee FindByName(String name) throws SQLException {

String sql="Select \* from Employee Where EMPLOYEE\_NAME="+name;

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

Employee employee=null;

while (rs.next())

{

int empId=rs.getInt("EMPLOYEE\_ID");

String empName=rs.getString("EMPLOYEE\_NAME");

double empSal=rs.getDouble("EMPLOYEE\_SALARY");

int empDept=rs.getInt("DEPARTMENT\_ID");

employee=new Employee(empId,empName,empSal,empDept);

}

return employee;

}

@Override

public int DeleteEmployee(int id) throws SQLException {

String sql="DELETE from Employee where Employee\_Id="+id;

PreparedStatement pstmt = connection.prepareStatement(sql);

int result = pstmt.executeUpdate();

return result;

}

@Override

public int UpdateEmployee(Employee employee) throws SQLException {

String sql="UPDATE EMPLOYEE SET EMPLOYEE\_NAME=?,EMPLOYEE\_SALARY=?,DEPARTMENT\_ID=? Where EMPLOYEE\_ID=?";

PreparedStatement pstmt = connection.prepareStatement(sql);

pstmt.setString(1, employee.getEmployeeName());

pstmt.setDouble(2, employee.getEmployeeSalary());

pstmt.setInt(3, employee.getDepartmentId());

pstmt.setInt(4, employee.getEmployeeId());

int result = pstmt.executeUpdate();

return result;

}

}

================================================================================

Service

package com.morningstar.service;

import java.util.List;

import com.morningstar.model.Employee;

public interface EmployeeService {

public boolean addEmployee(Employee employee);

public List<Employee> readAllEmployee();

public Employee findById(int id);

public Employee findByName(String name);

public boolean deleteEmployee(int id);

public boolean updateEmployee(Employee employee);

}

================================================================================

Service Impl

package com.morningstar.service;

import java.sql.SQLException;

import java.util.List;

import com.morningstar.dao.EmployeeDao;

import com.morningstar.dao.EmployeeDaoImpl;

import com.morningstar.model.Employee;

public class EmployeeServiceImpl implements EmployeeService {

private EmployeeDao dao=null;

public EmployeeDao getDao() {

return dao;

}

public void setDao(EmployeeDao dao) {

this.dao = dao;

}

@Override

public boolean addEmployee(Employee employee) {

try {

int result=dao.AddEmployee(employee);

if(result>=1)

{

EmployeeDaoImpl.getConnection().commit();

return true;

}

} catch (SQLException e) {

try {

EmployeeDaoImpl.getConnection().rollback();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

e.printStackTrace();

}

return false;

}

@Override

public List<Employee> readAllEmployee() {

List<Employee> list=null;

try {

list=dao.ReadAllEmployee();

} catch (SQLException e) {

}

return list;

}

@Override

public Employee findById(int id) {

Employee employee=null;

try {

employee=dao.FindById(id);

} catch (SQLException e) {

e.printStackTrace();

}

return employee;

}

@Override

public Employee findByName(String name) {

Employee employee=null;

try {

employee=dao.FindByName(name);

} catch (SQLException e) {

e.printStackTrace();

}

return employee;

}

@Override

public boolean deleteEmployee(int id) {

try {

int result=dao.DeleteEmployee(id);

if(result>=1)

{

return true;

}

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

@Override

public boolean updateEmployee(Employee employee) {

try {

int result=dao.UpdateEmployee(employee);

if(result>=1)

{

EmployeeDaoImpl.getConnection().commit();

return true;

}

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

}

================================================================================

Model

package com.morningstar.model;

public class Employee {

private int EmployeeId;

private String EmployeeName;

private double EmployeeSalary;

private int DepartmentId;

public Employee(int employeeId, String employeeName, double employeeSalary, int departmentId) {

super();

EmployeeId = employeeId;

EmployeeName = employeeName;

EmployeeSalary = employeeSalary;

DepartmentId = departmentId;

}

@Override

public String toString() {

return "EmployeeId :" + EmployeeId + ", EmployeeName :" + EmployeeName + ", EmployeeSalary :"

+ EmployeeSalary + ", DepartmentId :" + DepartmentId ;

}

public int getEmployeeId() {

return EmployeeId;

}

public void setEmployeeId(int employeeId) {

EmployeeId = employeeId;

}

public String getEmployeeName() {

return EmployeeName;

}

public void setEmployeeName(String employeeName) {

EmployeeName = employeeName;

}

public double getEmployeeSalary() {

return EmployeeSalary;

}

public void setEmployeeSalary(double employeeSalary) {

EmployeeSalary = employeeSalary;

}

public int getDepartmentId() {

return DepartmentId;

}

public void setDepartmentId(int departmentId) {

DepartmentId = departmentId;

}

}

App.java (Main)

package com.morningstar;

import java.util.List;

import java.util.Scanner;

import com.morningstar.dao.EmployeeDao;

import com.morningstar.dao.EmployeeDaoImpl;

import com.morningstar.model.Employee;

import com.morningstar.service.EmployeeServiceImpl;

public class App2 {

public static void main(String[] args) {

System.out.println("\*\*Welcome To Employee Management Application\*\*");

Scanner sc = new Scanner(System.in);

int status = 0;

EmployeeServiceImpl service = new EmployeeServiceImpl();

EmployeeDao dao = new EmployeeDaoImpl();

service.setDao(dao);

do {

System.out.println(" 1. Add Employee");

System.out.println(" 2. Read All Employees");

System.out.println(" 3. Find By Id");

System.out.println(" 4. Find by Name");

System.out.println(" 5. Delete Employee");

System.out.println(" 6. Update Employee");

int choice = sc.nextInt();

switch (choice) {

case 1: {

System.out.println("Adding new Employee");

System.out.println("Enter Employee Id");

int id = sc.nextInt();

System.out.println("Enter Employee Name");

String name = sc.next();

System.out.println("Enter Employee Salary");

double salary = sc.nextDouble();

System.out.println("Enter Department Id");

int deptId = sc.nextInt();

Employee employee=new Employee(id,name,salary,deptId);

if(service.addEmployee(employee))

{

System.out.println("New Employee is Added !");

}

else {

System.out.println("Error while adding new Employee");

}

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

case 2: {

System.out.println("\*\*\*\*\*\*\*\*Employees Directory\*\*\*\*\*\*\*\*\*");

List<Employee> result=service.readAllEmployee();

for (Employee employee : result) {

System.out.println(employee);

System.out.println("-------------------------------------------------------");

}

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

case 3: {

System.out.println("Finding Employee by ID");

System.out.println("Enter the Employee Id");

int id=sc.nextInt();

Employee employee=service.findById(id);

if(employee!=null)

{

System.out.println(employee);

}

else

System.out.println("Employee not found with the id :"+id);

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

case 4: {

System.out.println("Finding Employee by Name");

System.out.println("Enter the Employee Name");

String name=sc.next();

Employee employee=service.findByName(name);

if(employee!=null)

{

System.out.println(employee);

}

else

System.out.println("Employee not found with the id :"+name);

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

case 5: {

System.out.println("Deleting Employee");

System.out.println("Enter the employee id");

int id=sc.nextInt();

if(service.deleteEmployee(id))

{

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

}

else

System.out.println("Error While Deleting employee");

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

case 6: {

System.out.println("Updating Employee");

System.out.println("Enter Employee Id");

int id = sc.nextInt();

System.out.println("Enter Employee Name");

String name = sc.next();

System.out.println("Enter Employee Salary");

double salary = sc.nextDouble();

System.out.println("Enter Department Id");

int deptId = sc.nextInt();

Employee employee=new Employee(id,name,salary,deptId);

if(service.updateEmployee(employee))

{

System.out.println("Employee Updated for Employee Id : "+id);

}

else

System.out.println("Error While Updating Employee");

System.out.println("Do you want to continue ! y | n ");

String str=sc.next();

if(str.equals("y"))

{

status=1;

}

else {

status=0;

System.out.println("Thank you !!!!!!!!!!!");

}

break;

}

}

} while (status != 0);

}

}

Sql

Employee Table

Create Table Employee

(

Employee\_Id Number(5) Primary Key,

Employee\_Name varchar2(50) not null,

Employee\_Salary Number(9),

Department\_Id Number(5)

);