package data.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import db.mysql.DBConnect;

import data.dto.memberDto;

public class memberDao {

DBConnect db = new DBConnect();

//insert

//list출력

public List<memberDto> getAllData() {

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

Connection conn = db.getConnection();

PreparedStatement pstmt =null;

ResultSet rs = null;

String sql = "select \* from member";

try {

pstmt=conn.prepareStatement(sql);

rs=pstmt.executeQuery();

while(rs.next()) {

memberDto dto = new memberDto();

dto.setId(rs.getString("id"));

dto.setName(rs.getString("name"));

dto.setProfile(rs.getString("profile"));

dto.setPass(rs.getString("pass"));

dto.setHp(rs.getString("hp"));

dto.setEmail(rs.getString("email"));

dto.setAge(rs.getString("age"));

dto.setGender(rs.getString("gender"));

dto.setArea(rs.getString("area"));

dto.setMbti(rs.getString("mbti"));

dto.setRegister\_day(rs.getTimestamp("register\_day"));

list.add(dto);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(rs,pstmt,conn);

}

return list;

}

//Delete //num값에 따른 멤버 삭제하는 로직

public void deleteMember1(String num) {

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

String sql = "delete from member where num=?";

try {

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, num);

pstmt.execute();

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(pstmt, conn);

}

}

//회원정보수정

public void updateMember(memberDto dto) {

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

String sql = "update member set name=?,profile=?,hp=?,email=?,age=?,gender=?,area=?,mbti=? where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, dto.getName());

pstmt.setString(2, dto.getProfile());

pstmt.setString(3, dto.getHp());

pstmt.setString(4, dto.getEmail());

pstmt.setString(5, dto.getAge());

pstmt.setString(6, dto.getGender());

pstmt.setString(7, dto.getArea());

pstmt.setString(8, dto.getMbti());

pstmt.setString(9, dto.getId());

pstmt.execute();

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(pstmt, conn);

}

}

//id값 하나 가져오기

public memberDto getData(String id) {

memberDto dto = new memberDto();

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

ResultSet rs= null;

String sql = "select \* from member where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, id);

rs=pstmt.executeQuery();

if(rs.next()) {

dto.setId(rs.getString("id"));

dto.setName(rs.getString("name"));

dto.setProfile(rs.getString("profile"));

dto.setPass(rs.getString("pass"));

dto.setHp(rs.getString("hp"));

dto.setEmail(rs.getString("email"));

dto.setAge(rs.getString("age"));

dto.setGender(rs.getString("gender"));

dto.setArea(rs.getString("area"));

dto.setMbti(rs.getString("mbti"));

dto.setRegister\_day(rs.getTimestamp("register\_day"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(rs, pstmt,conn);

}

return dto;

}

//회원탈퇴 처리하는 로직

public boolean deleteMember(String id) {

boolean flag=false;

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

String sql = "delete from member where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, id);

int i = pstmt.executeUpdate();

if(i==1) {

flag=true;

} else {

flag=false;

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(pstmt, conn);

}

return flag;

}

//탈퇴할 때 비번 체크하기 로직

public int isPassCheck(String pass) {

int ispass = 0; //true

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

ResultSet rs = null;

String sql = "select count(\*) from member where pass=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, pass);

rs = pstmt.executeQuery();

if(rs.next()) {

ispass = rs.getInt(1);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(pstmt, conn);

}

return ispass;

}

//로그인할때 id,pass 확인

public boolean isIdPass(String id,String pass) {

boolean check = false;

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

ResultSet rs = null;

String sql = "select\*from member where id=? and pass=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, id);

pstmt.setString(2, pass);

rs=pstmt.executeQuery();

if(rs.next()) {

check = true;

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(rs, pstmt,conn);

}

return check;

}

//아이디에 따른 이름 반환 -> 누구누구 님이 로그인 하셨습니다 뜨게하기 위한 작업

public String getName(String id) {

String name="";

Connection conn = db.getConnection();

PreparedStatement pstmt = null;

ResultSet rs = null;

String sql = "select\*from member where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, id);

rs=pstmt.executeQuery();

if(rs.next()) {

name = rs.getString("name");

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

db.dbClose(rs,pstmt,conn);

}

return name;

}

}