﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Luong

{

public partial class Lương : Form

{

SqlConnection connection;

SqlCommand command;

string str = "Data Source=TTTT\\TTTT;Initial Catalog=NhanSu;Integrated Security=True";

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

void loaddata()

{

command = connection.CreateCommand();

command.CommandText = "select \* from Luong";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv2.DataSource = table;

}

private void dgv1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

//tb\_MaLuong.ReadOnly = true;

int i;

i = dgv2.CurrentRow.Index;

tb\_MaLuong.Text = dgv2.Rows[i].Cells[0].Value.ToString();

tb\_Thang.Text = dgv2.Rows[i].Cells[1].Value.ToString();

tb\_Nam.Text = dgv2.Rows[i].Cells[2].Value.ToString();

tb\_Thuong.Text = dgv2.Rows[i].Cells[3].Value.ToString();

tb\_MaNhanVien.Text = dgv2.Rows[i].Cells[4].Value.ToString();

tb\_MaLuong.Enabled = false;

}

public Lương()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

// Phương thức xóa trắng các TextBox

private void ClearInputFields()

{

tb\_MaLuong.Clear();

tb\_Thang.Clear();

tb\_Nam.Clear();

tb\_Thuong.Clear();

tb\_MaNhanVien.Clear();

tb\_MaLuong.Focus(); // Đưa con trỏ về ô nhập "Mã Lương"

}

private void bt\_Them\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string thang = tb\_Thang.Text.Trim();

string nam = tb\_Nam.Text.Trim();

string maNhanVien = tb\_MaNhanVien.Text.Trim();

string thuong = tb\_Thuong.Text.Trim();

// Kiểm tra nếu một trong các trường bắt buộc bị bỏ trống

if (string.IsNullOrEmpty(thang) || string.IsNullOrEmpty(nam) || string.IsNullOrEmpty(maNhanVien))

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin các trường bắt buộc: Tháng, Năm, Mã Nhân Viên.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra định dạng số cho Tháng và Năm

if (!int.TryParse(thang, out \_) || !int.TryParse(nam, out \_))

{

MessageBox.Show("Vui lòng nhập đúng định dạng số cho Tháng và Năm.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra định dạng số cho Thưởng nếu không để trống

if (!string.IsNullOrEmpty(thuong) && !decimal.TryParse(thuong, out \_))

{

MessageBox.Show("Vui lòng nhập đúng định dạng số cho Thưởng.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra MaNhanVien có tồn tại trong bảng NhanVien không

using (var command = connection.CreateCommand())

{

command.CommandText = "SELECT COUNT(\*) FROM NhanVien WHERE MaNhanVien = @MaNhanVien";

command.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

var exists = (int)command.ExecuteScalar();

if (exists == 0)

{

MessageBox.Show("Mã Nhân Viên không tồn tại trong bảng NhanVien.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// Tạo mã lương tự động tăng dần với định dạng 0000001, 0000002,...

string maLuong;

using (var command = connection.CreateCommand())

{

command.CommandText = "SELECT MAX(CAST(SUBSTRING(MaLuong, 2, 7) AS INT)) FROM Luong";

var result = command.ExecuteScalar();

int nextMaLuong = result == DBNull.Value ? 1 : Convert.ToInt32(result) + 1;

maLuong = nextMaLuong.ToString("D7"); // Đảm bảo mã lương có 7 chữ số (0000001, 0000002,...)

}

// Tạo câu lệnh SQL với tham số hóa

string query = "INSERT INTO Luong (MaLuong, Thang, Nam, Thuong, MaNhanVien) VALUES (@MaLuong, @Thang, @Nam, @Thuong, @MaNhanVien)";

using (var command = connection.CreateCommand())

{

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLuong", maLuong); // Mã lương tự động sinh

command.Parameters.AddWithValue("@Thang", thang); // Tháng

command.Parameters.AddWithValue("@Nam", nam); // Năm

command.Parameters.AddWithValue("@Thuong", string.IsNullOrEmpty(thuong) ? DBNull.Value : (object)thuong); // Thuong có thể trống

command.Parameters.AddWithValue("@MaNhanVien", maNhanVien); // Mã Nhân Viên

// Thực thi lệnh

command.ExecuteNonQuery();

}

// Cập nhật lại dữ liệu hiển thị

loaddata();

// Hiển thị thông báo thành công

MessageBox.Show("Thêm bản ghi Lương mới thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Xóa trắng các TextBox sau khi thêm

ClearInputFields();

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void Nam\_Click(object sender, EventArgs e)

{

}

private void bt\_Sua\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ các TextBox

string maLuong = tb\_MaLuong.Text.Trim();

string thuong = tb\_Thuong.Text.Trim();

// Kiểm tra nếu Mã Lương không được nhập

if (string.IsNullOrEmpty(maLuong))

{

MessageBox.Show("Vui lòng nhập Mã Lương để cập nhật thông tin!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra nếu Thưởng không được nhập

if (string.IsNullOrEmpty(thuong))

{

MessageBox.Show("Vui lòng nhập giá trị Thưởng để cập nhật!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra định dạng số cho Thưởng

if (!decimal.TryParse(thuong, out \_))

{

MessageBox.Show("Vui lòng nhập đúng định dạng số cho Thưởng.",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Tạo câu lệnh SQL với tham số hóa

string query = "UPDATE Luong SET Thuong = @Thuong WHERE MaLuong = @MaLuong";

// Tạo đối tượng SqlCommand

command = connection.CreateCommand();

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLuong", maLuong);

command.Parameters.AddWithValue("@Thuong", thuong);

// Thực thi lệnh

int rowsAffected = command.ExecuteNonQuery();

// Hiển thị thông báo

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật Thưởng thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Làm mới danh sách

// Xóa trắng các TextBox

ClearInputFields();

}

else

{

MessageBox.Show("Không tìm thấy Mã Lương để cập nhật!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Xem\_Click(object sender, EventArgs e)

{

try

{

// Lấy dữ liệu từ TextBox

string maNhanVien = tb\_MaNhanVien.Text.Trim();

string thang = tb\_Thang.Text.Trim();

string nam = tb\_Nam.Text.Trim();

string thuong = tb\_Thuong.Text.Trim();

string maLuong = tb\_MaLuong.Text.Trim();

// Kiểm tra nếu chỉ nhập "Thưởng" hoặc "Mã Lương"

if (string.IsNullOrEmpty(maNhanVien) && string.IsNullOrEmpty(thang) && string.IsNullOrEmpty(nam) &&

(!string.IsNullOrEmpty(thuong) || !string.IsNullOrEmpty(maLuong)))

{

MessageBox.Show("Không thể tìm kiếm với nội dung bạn đã cung cấp.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Kết thúc phương thức

}

// Kiểm tra nếu không nhập bất kỳ thông tin nào

if (string.IsNullOrEmpty(maNhanVien) && string.IsNullOrEmpty(thang) && string.IsNullOrEmpty(nam))

{

MessageBox.Show("Vui lòng nhập ít nhất một thông tin hợp lệ để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Kết thúc phương thức

}

// Xây dựng truy vấn SQL động với các tham số

string query = "SELECT \* FROM Luong WHERE 1=1";

if (!string.IsNullOrEmpty(maNhanVien))

{

query += " AND MaNhanVien = @MaNhanVien";

}

if (!string.IsNullOrEmpty(thang))

{

query += " AND Thang = @Thang";

}

if (!string.IsNullOrEmpty(nam))

{

query += " AND Nam = @Nam";

}

// Tạo đối tượng SqlCommand

command = connection.CreateCommand();

command.CommandText = query;

// Gắn tham số nếu có

if (!string.IsNullOrEmpty(maNhanVien))

{

command.Parameters.AddWithValue("@MaNhanVien", maNhanVien);

}

if (!string.IsNullOrEmpty(thang))

{

command.Parameters.AddWithValue("@Thang", thang);

}

if (!string.IsNullOrEmpty(nam))

{

command.Parameters.AddWithValue("@Nam", nam);

}

// Thực thi truy vấn và hiển thị kết quả

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

// Kiểm tra nếu không có dữ liệu trả về

if (table.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy dữ liệu nào khớp với điều kiện tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

dgv2.DataSource = table;

MessageBox.Show("Hiển thị dữ liệu thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Xoa\_Click(object sender, EventArgs e)

{

try

{

// Lấy Mã Lương từ TextBox

string maLuong = tb\_MaLuong.Text.Trim();

// Kiểm tra nếu Mã Lương không được nhập

if (string.IsNullOrEmpty(maLuong))

{

MessageBox.Show("Vui lòng nhập Mã Lương để xóa!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Câu lệnh SQL xóa bản ghi theo Mã Lương

string query = "DELETE FROM Luong WHERE MaLuong = @MaLuong";

// Tạo đối tượng SqlCommand

command = connection.CreateCommand();

command.CommandText = query;

// Gắn giá trị tham số

command.Parameters.AddWithValue("@MaLuong", maLuong);

// Thực thi lệnh

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra số lượng bản ghi bị xóa

if (rowsAffected > 0)

{

MessageBox.Show("Xóa bản ghi thành công!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Làm mới danh sách sau khi xóa

// Xóa trắng các TextBox

ClearInputFields();

}

else

{

MessageBox.Show("Không tìm thấy Mã Lương để xóa!",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

// Nếu có lỗi, hiển thị thông báo lỗi

MessageBox.Show("Có lỗi xảy ra: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void bt\_Tinh\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection conn = new SqlConnection(str))

{

// Mở kết nối đến cơ sở dữ liệu

conn.Open();

string thang = tb\_Thang.Text.Trim();

string nam = tb\_Nam.Text.Trim();

// Tạo một đối tượng SqlCommand để gọi thủ tục

using (SqlCommand cmd = new SqlCommand("Sp\_3\_ThongTinLuong", conn))

{

cmd.CommandType = CommandType.StoredProcedure;

// Thêm tham số cho thủ tục

cmd.Parameters.AddWithValue("@Thang", thang);

cmd.Parameters.AddWithValue("@Nam", nam);

// Tạo một DataAdapter để lấy dữ liệu trả về

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

DataTable dt = new DataTable();

// Điền dữ liệu vào DataTable

da.Fill(dt);

// Kiểm tra nếu có dữ liệu

if (dt.Rows.Count > 0)

{

// Gắn dữ liệu vào DataGridView

dgv2.DataSource = dt;

}

else

{

MessageBox.Show("Không có dữ liệu lương cho tháng và năm này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

}

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi khi truy vấn thông tin lương: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void btReset\_Click(object sender, EventArgs e)

{

tb\_MaLuong.Clear();

tb\_Thang.Clear();

tb\_Nam.Clear();

tb\_Thuong.Clear();

tb\_MaNhanVien.Clear();

tb\_MaLuong.Focus();

loaddata();

}

}

}