LAB 5:

WINFORM KẾT NỐI CƠ SỞ DỮ LIỆU NÂNG CAO

Họ và tên: Đinh Thị Thảo An

Lớp: 11\_ĐH\_CNPM1

Mssv: 1150080083

Mục lục

[**Thực hành 1: Tạo và kết nối CSDL ngay bên trong Project Winform.** 1](#_Toc211271405)

[**Thực hành 2: Thêm dữ liệu không dùng Parameter.** 2](#_Toc211271406)

[**Áp dụng 1: Thêm dữ liệu có dùng Parameter.** 8](#_Toc211271407)

[**Thực hành 3: Sửa dữ liệu không dùng Parameter.** 14](#_Toc211271408)

[**Áp dụng 2: Sửa dữ liệu có dùng Parameter.** 21](#_Toc211271409)

[**Thực hành 4: Xóa dữ liệu không dùng Parameter.** 28](#_Toc211271410)

[**Áp dụng 3: Xóa dữ liệu có dùng Parameter.** 34](#_Toc211271411)

# **Thực hành 1: Tạo và kết nối CSDL ngay bên trong Project Winform.**

− Khởi tạo 1 project với template là Windowns Form Applications (.NET framework).

− Chuột phải vào project chọn Add > New Item.

− Trong danh sách các template, cuộn xuống và chọn Service-based Database.

− Sau đó, đặt tên cho CSDL, ví dụ như DBConnect.mdf và nhấn Add.

A screenshot of a computer

AI-generated content may be incorrect.

Visual Studio sẽ tự động hiển thị tab Server Explorer.

A screenshot of a computer

AI-generated content may be incorrect.

# **Thực hành 2: Thêm dữ liệu không dùng Parameter.**

− Sử dụng hàm ExecuteNonQuery để thực hiện thêm dữ liệu.

− Ví dụ: Thêm dữ liệu cho 1 sinh viên vào bảng SinhVien.

Code

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form1 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=E:\CNTT\Hoc\_ki\_7\_2025\Phuong\_phat\_trien\_phan\_mem\_huong\_doi\_tuong\BTtuan6\1150080083\_lab5\1150080083\_lab5\DBConnect.mdf;

Integrated Security=True;

Connect Timeout=30";

public Form1()

{

InitializeComponent();

cbGioiTinh.Items.Add("Nam");

cbGioiTinh.Items.Add("Nữ");

if (lsvDanhSachSV.Columns.Count == 0)

{

lsvDanhSachSV.View = View.Details;

lsvDanhSachSV.FullRowSelect = true;

lsvDanhSachSV.Columns.Add("Mã SV", 90);

lsvDanhSachSV.Columns.Add("Tên SV", 150);

lsvDanhSachSV.Columns.Add("Giới tính", 70);

lsvDanhSachSV.Columns.Add("Ngày sinh", 90);

lsvDanhSachSV.Columns.Add("Quê quán", 140);

lsvDanhSachSV.Columns.Add("Mã lớp", 80);

}

HienThiDanhSach();

}

private void MoKetNoi()

{

if (sqlCon == null)

sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open)

sqlCon.Close();

}

private void HienThiDanhSach()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand("SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop FROM SinhVien", sqlCon))

using (var reader = cmd.ExecuteReader())

{

lsvDanhSachSV.Items.Clear();

while (reader.Read())

{

string maSv = reader.GetString(0);

string tenSv = reader.GetString(1);

string gioiTinh = reader.GetString(2);

string ngaySinh = reader.GetDateTime(3).ToString("dd/MM/yyyy");

string queQuan = reader.IsDBNull(4) ? "" : reader.GetString(4);

string maLop = reader.IsDBNull(5) ? "" : reader.GetString(5);

var lvi = new ListViewItem(maSv);

lvi.SubItems.Add(tenSv);

lvi.SubItems.Add(gioiTinh);

lvi.SubItems.Add(ngaySinh);

lvi.SubItems.Add(queQuan);

lvi.SubItems.Add(maLop);

lsvDanhSachSV.Items.Add(lvi);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi hiển thị: " + ex.Message);

}

finally

{

DongKetNoi();

}

}

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

{

string maSV = txtMaSV.Text.Trim();

string tenSV = txtTenSV.Text.Trim();

string gioiTinh = cbGioiTinh.SelectedItem?.ToString() ?? "";

string ngaySinhSql = dtpNgaySinh.Value.ToString("yyyy-MM-dd");

string queQuan = txtQueQuan.Text.Trim();

string maLop = txtMaLop.Text.Trim();

if (string.IsNullOrEmpty(maSV) || string.IsNullOrEmpty(tenSV) || string.IsNullOrEmpty(gioiTinh))

{

MessageBox.Show("Mã SV, Tên SV và Giới tính là bắt buộc.");

return;

}

string sql =

"INSERT INTO SinhVien(MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop) " +

"VALUES (N'" + maSV.Replace("'", "''") + "', " +

"N'" + tenSV.Replace("'", "''") + "', " +

"N'" + gioiTinh.Replace("'", "''") + "', " +

"'" + ngaySinhSql + "', " +

"N'" + queQuan.Replace("'", "''") + "', " +

"N'" + maLop.Replace("'", "''") + "')";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Thêm sinh viên thành công!");

HienThiDanhSach();

}

else

{

MessageBox.Show("Không có bản ghi nào được thêm.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Thêm dữ liệu bị lỗi: " + ex.Message);

}

finally

{

DongKetNoi();

}

}

}

}

Kết quả

A screenshot of a computer

AI-generated content may be incorrect.

# **Áp dụng 1: Thêm dữ liệu có dùng Parameter.**

**Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form2 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=|DataDirectory|\DBConnect.mdf;

Integrated Security=True;Connect Timeout=30";

public Form2()

{

InitializeComponent();

if (cbGioiTinh.Items.Count == 0)

{

cbGioiTinh.Items.Add("Nam");

cbGioiTinh.Items.Add("Nữ");

}

HienThiDanhSach();

}

private void MoKetNoi()

{

if (sqlCon == null) sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open)

sqlCon.Close();

}

private void HienThiDanhSach()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand(

"SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop FROM SinhVien", sqlCon))

using (var r = cmd.ExecuteReader())

{

lsvDanhSach.Items.Clear();

while (r.Read())

{

var it = new ListViewItem(r["MaSV"].ToString());

it.SubItems.Add(r["TenSV"].ToString());

it.SubItems.Add(r["GioiTinh"].ToString());

it.SubItems.Add(Convert.ToDateTime(r["NgaySinh"]).ToString("dd/MM/yyyy"));

it.SubItems.Add(r["QueQuan"].ToString());

it.SubItems.Add(r["MaLop"].ToString());

lsvDanhSach.Items.Add(it);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi hiển thị: " + ex.Message);

}

finally

{

DongKetNoi();

}

}

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

{

string maSV = txtMaSV.Text.Trim();

string tenSV = txtTenSV.Text.Trim();

string gioiTinh = cbGioiTinh.SelectedItem?.ToString() ?? "";

DateTime ngaySinh = dtpNgaySinh.Value.Date;

string queQuan = txtQueQuan.Text.Trim();

string maLop = txtMaLop.Text.Trim();

if (string.IsNullOrWhiteSpace(maSV) ||

string.IsNullOrWhiteSpace(tenSV) ||

string.IsNullOrWhiteSpace(gioiTinh))

{

MessageBox.Show("Mã SV, Tên SV, Giới tính là bắt buộc.");

return;

}

const string sql = @"

INSERT INTO SinhVien (MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop)

VALUES (@MaSV, @TenSV, @GioiTinh, @NgaySinh, @QueQuan, @MaLop);";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

cmd.Parameters.Add("@MaSV", SqlDbType.NVarChar, 20).Value = maSV;

cmd.Parameters.Add("@TenSV", SqlDbType.NVarChar, 100).Value = tenSV;

cmd.Parameters.Add("@GioiTinh", SqlDbType.NVarChar, 5).Value = gioiTinh;

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = ngaySinh;

cmd.Parameters.Add("@QueQuan", SqlDbType.NVarChar, 100).Value =

string.IsNullOrWhiteSpace(queQuan) ? (object)DBNull.Value : queQuan;

cmd.Parameters.Add("@MaLop", SqlDbType.NVarChar, 20).Value =

string.IsNullOrWhiteSpace(maLop) ? (object)DBNull.Value : maLop;

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Thêm sinh viên thành công (Parameter)!");

HienThiDanhSach();

ClearInputs();

}

}

}

catch (SqlException ex) when (ex.Number == 2627 || ex.Number == 2601)

{

MessageBox.Show("Mã SV đã tồn tại. Vui lòng chọn mã khác.");

}

catch (Exception ex)

{

MessageBox.Show("Lỗi thêm: " + ex.Message);

}

finally

{

DongKetNoi();

}

}

private void ClearInputs()

{

txtMaSV.Clear();

txtTenSV.Clear();

cbGioiTinh.SelectedIndex = -1;

dtpNgaySinh.Value = DateTime.Today;

txtQueQuan.Clear();

txtMaLop.Clear();

txtMaSV.Focus();

}

}

}

**Kết quả**

**A screenshot of a computer

AI-generated content may be incorrect.**

# **Thực hành 3: Sửa dữ liệu không dùng Parameter.**

**Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form3 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=|DataDirectory|\DBConnect.mdf;

Integrated Security=True;Connect Timeout=30";

public Form3()

{

InitializeComponent();

}

// ===== Helpers mở/đóng kết nối =====

private void MoKetNoi()

{

if (sqlCon == null) sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open) sqlCon.Close();

}

// ===== Load danh sách mã lớp vào combobox (Lop: MaLop, TenLop) =====

private void HienThiDSMaLop()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand("SELECT MaLop, TenLop FROM Lop", sqlCon))

using (var r = cmd.ExecuteReader())

{

cbMaLop.Items.Clear();

while (r.Read())

{

string ma = r.GetString(0);

string ten = r.IsDBNull(1) ? "" : r.GetString(1);

cbMaLop.Items.Add($"{ma} - {ten}");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải danh sách lớp: " + ex.Message);

}

finally { DongKetNoi(); }

}

// ===== Hiển thị SV theo mã lớp đã chọn =====

private void HienThiDSSinhVienTheoLop(string maLop)

{

try

{

MoKetNoi();

string sql = "SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop " +

"FROM SinhVien WHERE MaLop='" + maLop.Replace("'", "''") + "'";

using (var cmd = new SqlCommand(sql, sqlCon))

using (var r = cmd.ExecuteReader())

{

lsvDanhSach.Items.Clear();

while (r.Read())

{

var item = new ListViewItem(r["MaSV"].ToString());

item.SubItems.Add(r["TenSV"].ToString());

item.SubItems.Add(r["GioiTinh"].ToString());

item.SubItems.Add(Convert.ToDateTime(r["NgaySinh"]).ToString("dd/MM/yyyy"));

item.SubItems.Add(r["QueQuan"].ToString());

item.SubItems.Add(r["MaLop"].ToString());

lsvDanhSach.Items.Add(item);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi hiển thị SV theo lớp: " + ex.Message);

}

finally { DongKetNoi(); }

}

// ===== Form Load =====

private void Form3\_Load(object sender, EventArgs e)

{

HienThiDSMaLop();

}

// ===== Chọn lớp -> load DS SV =====

private void cbMaLop\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbMaLop.SelectedIndex == -1) return;

var parts = cbMaLop.SelectedItem.ToString().Split('-');

string maLop = parts[0].Trim();

HienThiDSSinhVienTheoLop(maLop);

btnSuaThongTin.Enabled = false;

}

// ===== Chọn 1 dòng SV -> đổ vào control bên phải =====

private void lsvDanhSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvDanhSach.SelectedItems.Count == 0)

{

btnSuaThongTin.Enabled = false;

return;

}

var lvi = lsvDanhSach.SelectedItems[0];

txtMaSV.Text = lvi.SubItems[0].Text;

txtTenSV.Text = lvi.SubItems[1].Text;

cbGioiTinh.SelectedItem = lvi.SubItems[2].Text;

// dd/MM/yyyy

if (DateTime.TryParse(lvi.SubItems[3].Text, out DateTime d))

dtpNgaySinh.Value = d;

txtQueQuan.Text = lvi.SubItems[4].Text;

txtMaLop.Text = lvi.SubItems[5].Text;

btnSuaThongTin.Enabled = true;

}

// ===== Nút Sửa (KHÔNG Parameter) =====

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

{

string maSV = txtMaSV.Text.Trim();

string tenSV = txtTenSV.Text.Trim();

string gioiTinh = cbGioiTinh.SelectedItem?.ToString() ?? "";

string ngaySinh = dtpNgaySinh.Value.ToString("yyyy/MM/dd");

string queQuan = txtQueQuan.Text.Trim();

string maLop = txtMaLop.Text.Trim();

if (string.IsNullOrWhiteSpace(maSV))

{

MessageBox.Show("Vui lòng chọn sinh viên cần sửa (Mã SV).");

return;

}

string sql = "UPDATE SinhVien SET " +

"TenSV = N'" + tenSV.Replace("'", "''") + "', " +

"GioiTinh = N'" + gioiTinh.Replace("'", "''") + "', " +

"NgaySinh = CAST('" + ngaySinh + "' AS DATETIME), " +

"QueQuan = N'" + queQuan.Replace("'", "''") + "', " +

"MaLop = N'" + maLop.Replace("'", "''") + "' " +

"WHERE MaSV = N'" + maSV.Replace("'", "''") + "'";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Cập nhật thành công!");

HienThiDSSinhVienTheoLop(maLop);

}

else MessageBox.Show("Không có bản ghi nào được cập nhật.");

}

}

catch (Exception ex)

{

MessageBox.Show("Cập nhật bị lỗi: " + ex.Message);

}

finally { DongKetNoi(); }

}

}

}

**Kết quả**

**A screenshot of a computer

AI-generated content may be incorrect.**

# **Áp dụng 2: Sửa dữ liệu có dùng Parameter.**

**Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form4 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=|DataDirectory|\DBConnect.mdf;

Integrated Security=True;Connect Timeout=30";

public Form4()

{

InitializeComponent();

}

// ===== Helpers =====

private void MoKetNoi()

{

if (sqlCon == null) sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open) sqlCon.Close();

}

// ===== Load combobox lớp =====

private void HienThiDSMaLop()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand("SELECT MaLop, TenLop FROM Lop", sqlCon))

using (var r = cmd.ExecuteReader())

{

cbMaLop.Items.Clear();

while (r.Read())

{

string ma = r.GetString(0);

string ten = r.IsDBNull(1) ? "" : r.GetString(1);

cbMaLop.Items.Add($"{ma} - {ten}");

}

}

}

catch (Exception ex) { MessageBox.Show("Lỗi tải lớp: " + ex.Message); }

finally { DongKetNoi(); }

}

// ===== Load SV theo lớp (cũng dùng parameter) =====

private void HienThiDSSVTheoLop(string maLop)

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand(

"SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop FROM SinhVien WHERE MaLop=@MaLop", sqlCon))

{

cmd.Parameters.Add("@MaLop", SqlDbType.NVarChar, 20).Value = maLop;

using (var r = cmd.ExecuteReader())

{

lsvDanhSach.Items.Clear();

while (r.Read())

{

var item = new ListViewItem(r["MaSV"].ToString());

item.SubItems.Add(r["TenSV"].ToString());

item.SubItems.Add(r["GioiTinh"].ToString());

item.SubItems.Add(Convert.ToDateTime(r["NgaySinh"]).ToString("dd/MM/yyyy"));

item.SubItems.Add(r["QueQuan"].ToString());

item.SubItems.Add(r["MaLop"].ToString());

lsvDanhSach.Items.Add(item);

}

}

}

}

catch (Exception ex) { MessageBox.Show("Lỗi hiển thị SV: " + ex.Message); }

finally { DongKetNoi(); }

}

// ===== Events =====

private void Form4\_Load(object sender, EventArgs e)

{

cbGioiTinh.Items.Clear();

cbGioiTinh.Items.Add("Nam");

cbGioiTinh.Items.Add("Nữ");

HienThiDSMaLop();

}

private void cbMaLop\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbMaLop.SelectedIndex == -1) return;

string maLop = cbMaLop.SelectedItem.ToString().Split('-')[0].Trim();

HienThiDSSVTheoLop(maLop);

btnSua.Enabled = false;

}

private void lsvDanhSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvDanhSach.SelectedItems.Count == 0)

{

btnSua.Enabled = false;

return;

}

var lvi = lsvDanhSach.SelectedItems[0];

txtMaSV.Text = lvi.SubItems[0].Text;

txtTenSV.Text = lvi.SubItems[1].Text;

cbGioiTinh.SelectedItem = lvi.SubItems[2].Text;

if (DateTime.TryParse(lvi.SubItems[3].Text, out var d)) dtpNgaySinh.Value = d;

txtQueQuan.Text = lvi.SubItems[4].Text;

txtMaLop.Text = lvi.SubItems[5].Text;

btnSua.Enabled = true;

}

// ===== Update với Parameter =====

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

{

string maSV = txtMaSV.Text.Trim();

string tenSV = txtTenSV.Text.Trim();

string gioiTinh = cbGioiTinh.SelectedItem?.ToString() ?? "";

DateTime ngaySinh = dtpNgaySinh.Value.Date;

string queQuan = txtQueQuan.Text.Trim();

string maLop = txtMaLop.Text.Trim();

if (string.IsNullOrWhiteSpace(maSV))

{

MessageBox.Show("Vui lòng chọn sinh viên cần sửa.");

return;

}

const string sql = @"

UPDATE SinhVien

SET TenSV=@TenSV, GioiTinh=@GioiTinh, NgaySinh=@NgaySinh,

QueQuan=@QueQuan, MaLop=@MaLop

WHERE MaSV=@MaSV;";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

cmd.Parameters.Add("@TenSV", SqlDbType.NVarChar, 100).Value = tenSV;

cmd.Parameters.Add("@GioiTinh", SqlDbType.NVarChar, 5).Value = gioiTinh;

cmd.Parameters.Add("@NgaySinh", SqlDbType.Date).Value = ngaySinh;

cmd.Parameters.Add("@QueQuan", SqlDbType.NVarChar, 100).Value =

string.IsNullOrWhiteSpace(queQuan) ? (object)DBNull.Value : queQuan;

cmd.Parameters.Add("@MaLop", SqlDbType.NVarChar, 20).Value =

string.IsNullOrWhiteSpace(maLop) ? (object)DBNull.Value : maLop;

cmd.Parameters.Add("@MaSV", SqlDbType.NVarChar, 20).Value = maSV;

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Cập nhật thành công (Parameter)!");

HienThiDSSVTheoLop(maLop);

}

else MessageBox.Show("Không có bản ghi nào được cập nhật.");

}

}

catch (Exception ex) { MessageBox.Show("Lỗi cập nhật: " + ex.Message); }

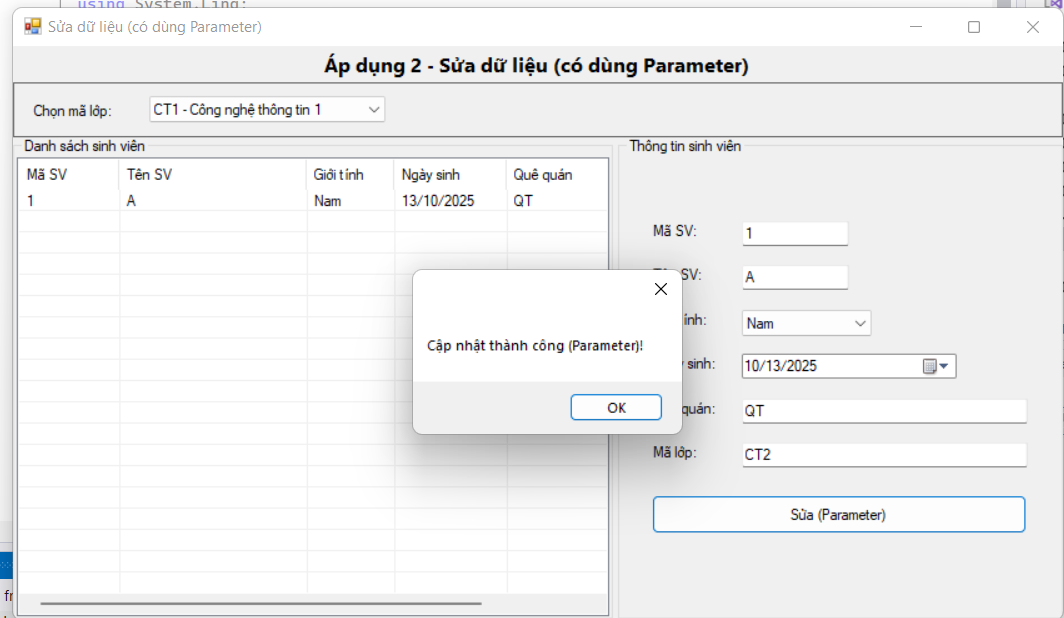
finally { DongKetNoi(); }

}

}

}

**Kết quả**

****

# **Thực hành 4: Xóa dữ liệu không dùng Parameter.**

**Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form5 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=|DataDirectory|\DBConnect.mdf;

Integrated Security=True;Connect Timeout=30";

private string \_selectedMaSV = string.Empty;

public Form5()

{

InitializeComponent();

}

// ========= Kết nối =========

private void MoKetNoi()

{

if (sqlCon == null) sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open) sqlCon.Close();

}

// ========= Load danh sách =========

private void HienThiDSSinhVien()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand(

"SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop FROM SinhVien", sqlCon))

using (var r = cmd.ExecuteReader())

{

lsvDanhSach.Items.Clear();

while (r.Read())

{

var it = new ListViewItem(r["MaSV"].ToString());

it.SubItems.Add(r["TenSV"].ToString());

it.SubItems.Add(r["GioiTinh"].ToString());

it.SubItems.Add(Convert.ToDateTime(r["NgaySinh"]).ToString("dd/MM/yyyy"));

it.SubItems.Add(r["QueQuan"].ToString());

it.SubItems.Add(r["MaLop"].ToString());

lsvDanhSach.Items.Add(it);

}

}

\_selectedMaSV = string.Empty;

btnXoa.Enabled = false;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi hiển thị: " + ex.Message);

}

finally { DongKetNoi(); }

}

// ========= Events =========

private void Form5\_Load(object sender, EventArgs e)

{

HienThiDSSinhVien();

}

private void lsvDanhSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvDanhSach.SelectedItems.Count == 0)

{

\_selectedMaSV = string.Empty;

btnXoa.Enabled = false;

return;

}

\_selectedMaSV = lsvDanhSach.SelectedItems[0].SubItems[0].Text.Trim();

btnXoa.Enabled = true;

}

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

{

if (string.IsNullOrEmpty(\_selectedMaSV))

{

MessageBox.Show("Bạn chưa chọn sinh viên để xóa.");

return;

}

var confirm = MessageBox.Show(

$"Bạn có chắc muốn xóa sinh viên mã '{\_selectedMaSV}'?",

"Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (confirm != DialogResult.Yes) return;

XoaSinhVien(\_selectedMaSV);

}

private void XoaSinhVien(string maSV)

{

string sql = "DELETE FROM SinhVien WHERE MaSV = N'" + maSV.Replace("'", "''") + "'";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Xóa dữ liệu thành công!");

HienThiDSSinhVien();

}

else

{

MessageBox.Show("Không có bản ghi nào bị xóa.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa dữ liệu: " + ex.Message);

}

finally { DongKetNoi(); }

}

}

}

**Kết quả**

A screenshot of a computer

AI-generated content may be incorrect.

# **Áp dụng 3: Xóa dữ liệu có dùng Parameter.**

**Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace \_1150080083\_lab5

{

public partial class Form6 : Form

{

private SqlConnection sqlCon = null;

private readonly string strCon =

@"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=|DataDirectory|\DBConnect.mdf;

Integrated Security=True;Connect Timeout=30";

private string \_selectedMaSV = string.Empty;

public Form6()

{

InitializeComponent();

}

// ===== helpers =====

private void MoKetNoi()

{

if (sqlCon == null) sqlCon = new SqlConnection(strCon);

if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();

}

private void DongKetNoi()

{

if (sqlCon != null && sqlCon.State == ConnectionState.Open) sqlCon.Close();

}

// ===== load list =====

private void HienThiDSSinhVien()

{

try

{

MoKetNoi();

using (var cmd = new SqlCommand(

"SELECT MaSV, TenSV, GioiTinh, NgaySinh, QueQuan, MaLop FROM SinhVien", sqlCon))

using (var r = cmd.ExecuteReader())

{

lsvDanhSach.Items.Clear();

while (r.Read())

{

var it = new ListViewItem(r["MaSV"].ToString());

it.SubItems.Add(r["TenSV"].ToString());

it.SubItems.Add(r["GioiTinh"].ToString());

it.SubItems.Add(Convert.ToDateTime(r["NgaySinh"]).ToString("dd/MM/yyyy"));

it.SubItems.Add(r["QueQuan"].ToString());

it.SubItems.Add(r["MaLop"].ToString());

lsvDanhSach.Items.Add(it);

}

}

\_selectedMaSV = string.Empty;

btnXoa.Enabled = false;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi hiển thị: " + ex.Message);

}

finally { DongKetNoi(); }

}

// ===== events =====

private void Form6\_Load(object sender, EventArgs e)

{

HienThiDSSinhVien();

}

private void lsvDanhSach\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lsvDanhSach.SelectedItems.Count == 0)

{

\_selectedMaSV = string.Empty;

btnXoa.Enabled = false;

return;

}

\_selectedMaSV = lsvDanhSach.SelectedItems[0].SubItems[0].Text.Trim();

btnXoa.Enabled = true;

}

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

{

if (string.IsNullOrEmpty(\_selectedMaSV))

{

MessageBox.Show("Bạn chưa chọn sinh viên để xóa.");

return;

}

var confirm = MessageBox.Show(

$"Bạn có chắc muốn xóa sinh viên mã '{\_selectedMaSV}'?",

"Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (confirm != DialogResult.Yes) return;

XoaSinhVienTheoMa(\_selectedMaSV);

}

// ===== delete with parameter =====

private void XoaSinhVienTheoMa(string maSV)

{

const string sql = "DELETE FROM SinhVien WHERE MaSV = @MaSV";

try

{

MoKetNoi();

using (var cmd = new SqlCommand(sql, sqlCon))

{

cmd.Parameters.Add("@MaSV", SqlDbType.NVarChar, 20).Value = maSV;

int kq = cmd.ExecuteNonQuery();

if (kq > 0)

{

MessageBox.Show("Xóa dữ liệu thành công (Parameter)!");

HienThiDSSinhVien();

}

else

{

MessageBox.Show("Không có bản ghi nào bị xóa.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa dữ liệu: " + ex.Message);

}

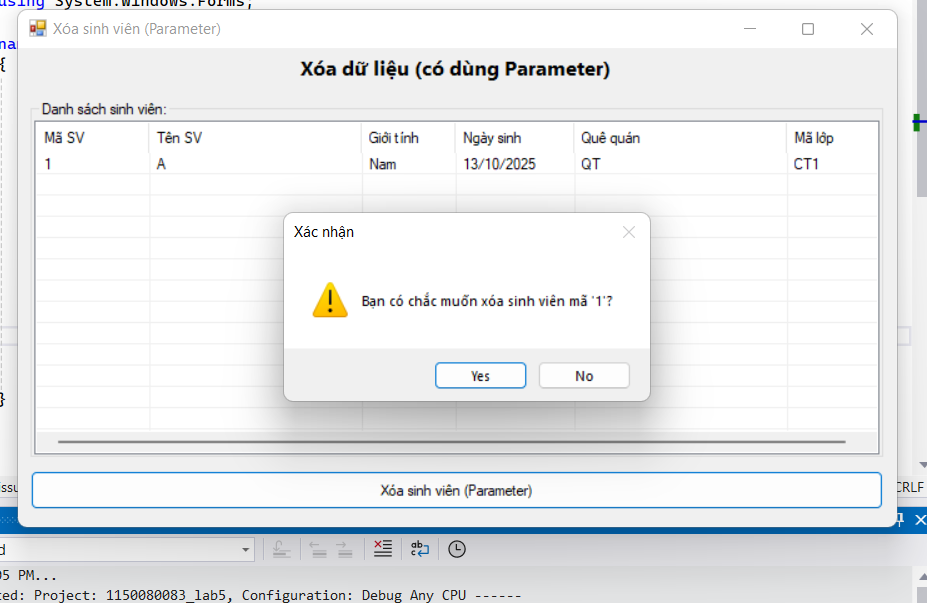
finally { DongKetNoi(); }

}

}

}

**Kết quả**

****