

Trường ĐH CT TP. HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	BÀI 10. LẬP TRÌNH TƯƠNG TÁC CƠ SỞ DỮ LIỆU (tiếp theo)	
---	--	--

A. MỤC TIÊU:

- Sử dụng được đối tượng SqlDataAdapter, DataSet.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader hiển thị khóa ngoại cho combobox.
- Sử dụng SqlCommand thực hiện thêm, xóa, sửa cho một bảng dữ liệu.
- Sử dụng DataReader kiểm tra khóa chính/ khóa ngoại trước khi cập nhật dữ liệu.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

- **SqlDataAdapter:** là cầu nối của Database và Dataset. SqlDataAdapter chứa các lệnh SQL (dạng Select) và đối tượng Connection để đọc và ghi dữ liệu.
- **DataSet:** nhận dữ liệu từ SqlDataAdapter, và cung cấp các phương thức xử lý dữ liệu. DataSet có nhiều đối tượng đi kèm theo như sau:
 - DataTable (tương đương với 1 bảng trong database)
 - DataRow (tương đương với 1 dòng trong bảng)
 - DataColumn (tương đương với 1 cột trong bảng)
- **SqlCommandBuilder:** là đối tượng cập nhật dữ liệu từ SqlDataAdapter vào cơ sở dữ liệu.

1.2 Giới thiệu bài tập mẫu

Thiết kế form Quản lý Khoa hoàn chỉnh (nhập trên Textbox), các control giãn theo tỷ lệ tùy ý (Sử dụng cơ sở dữ liệu QLSinhVien)

GIAO DIỆN KHOA HOÀN CHỈNH

Thêm Xóa Sửa Lưu Xem In Đóng

Mã khoa

Tên khoa

	Mã khoa	Tên khoa
*		

Yêu cầu:

- FormLoad:
 - + Datagridview: hiển thị tất cả khoa trong bảng Khoa
 - + Tất cả textbox bị vô hiệu hóa
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa
- Khi chọn vào button lệnh Thêm:
 - + Các Textbox có hiệu lực
 - + Button Lưu có hiệu lực
 - + Dấu nháy xuất hiện ở textbox Mã khoa.
- Khi chọn vào Datagrid
 - + Hiển thị thông tin tương ứng lên các textbox
 - + Button Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Các textbox có hiệu lực trừ textbox Mã khoa
 - + Cho phép sửa các thông tin còn lại
- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa nhập hoặc sửa cho phù hợp
 - + Lưu vào Cơ sở dữ liệu (cần xác định là đang Lưu cho hành động Thêm hay Sửa).
 - + Thông báo thành công hoặc báo lỗi nếu có.
 - + Button Lưu bị vô hiệu hóa.

- Khi nhấn button Xóa.
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra mã khoa định xóa có tồn tại trong bảng lớp hay không. Nếu có thì thông báo “Dữ liệu đang sử dụng không thể xóa?”.
 - Nếu mã khoa không tồn tại bên bảng lớp thì xóa bên bảng khoa.
 - + Hiển thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

Công việc 1: Thiết kế giao diện hoàn chỉnh như hình trên

Công việc 2: Xử lý yêu cầu của đề bài

```

SqlConnection _Connsql = new SqlConnection("Data Source =
<Tên_server>; Initial Catalog = QLSINHVIEN; User ID =
<Tài_khoản_ĐN>; Password = <Mật_khẩu>");
DataSet ds_QLSV = new DataSet();
public frmQuanLyKhoa()
{
    InitializeComponent();
}
//Phương thức load dữ liệu cho Datagridview
void LoadDuLieuKhoa()
{
    string strsel = "select * from Khoa";
    SqlDataAdapter da_Khoa=new SqlDataAdapter(strsel, _Connsql);
    da_Khoa.Fill(ds_QLSV, "Khoa");
    dataGridView1.DataSource = ds_QLSV.Tables["Khoa"];
    DataColumn[] key = new DataColumn[1];
    key[0] = ds_QLSV.Tables["Khoa"].Columns[0];
    ds_QLSV.Tables["Khoa"].PrimaryKey = key;
}
//Sự kiện Load của form
private void frmQuanLyKhoa_Load(object sender, EventArgs e)
{
    LoadDuLieuKhoa();
    //Tất cả textbox bị vô hiệu hóa
    txtMaKhoa.Enabled = txtTenKhoa.Enabled = false;
    //Các button Sửa, xóa, Lưu bị vô hiệu hóa
    btnSua.Enabled = btnXoa.Enabled = btnLuu.Enabled = false;
}

```

```

//Sự kiện click của button Thêm
private void btnThem_Click(object sender, EventArgs e)
{
    //Các textbox có hiệu lực
    txtMaKhoa.Enabled = txtTenKhoa.Enabled = true;
    //button Lưu có hiệu lực
    btnLuu.Enabled = true;
    //Dấu nháy xuất hiện ở textbox Mã khoa.
    txtMaKhoa.Focus();
    //Xóa dữ liệu hiện tại trên các textbox.
    txtMaKhoa.Clear();
    txtTenKhoa.Clear();
}
//Sự kiện SelectionChanged của datagridview
private void dgv_Khoa_SelectionChanged(object sender, EventArgs e)
{
    //Button Sửa và Xóa có hiệu lực
    btnSua.Enabled = btnXoa.Enabled = true;
}
//Sự kiện click của button Sửa
private void btnSua_Click(object sender, EventArgs e)
{
    //Button Lưu có hiệu lực
    btnLuu.Enabled = true;
    //Các textbox có hiệu lực trừ textbox Mã khoa
    txtTenKhoa.Enabled = true;
    txtMaKhoa.Enabled = false;
}
//Sự kiện click của button Lưu
private void btnLuu_Click(object sender, EventArgs e)
{
    //Kiểm tra thông tin vừa nhập hoặc sửa cho phù hợp
    if (txtMaKhoa.Text == string.Empty)
    {
        MessageBox.Show("Chưa nhập Mã Khoa");
        txtMaKhoa.Focus();
        return; }
}

```

```

if (txtTenKhoa.Text == string.Empty)
{
    MessageBox.Show("Chưa nhập Tên Khoa");
    txtTenKhoa.Focus();
    return;
}
if (txtMaKhoa.Enabled == true) // Thêm
{
    DataRow insert_New = ds_QLSV.Tables["Khoa"].NewRow();
    insert_New["MaKhoa"] = txtMaKhoa.Text;
    insert_New["TenKhoa"] = txtTenKhoa.Text;
    ds_QLSV.Tables["Khoa"].Rows.Add(insert_New);
}
else // Sửa
{
    DataRow update_New = ds_QLSV.Tables["Khoa"].Rows.Find
(txtMaKhoa.Text);
    if (update_New != null)
    {
        update_New["TenKhoa"] = txtTenKhoa.Text;
    }
}
SqlCommandBuilder cmb = new SqlCommandBuilder(da_Khoa);
da_Khoa.Update(ds_QLSV, "Khoa")
MessageBox.Show("Thành công");
btnLuu.Enabled = false;
}
//Sự kiện click của button Xóa
private void btnXoa_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Bạn muốn xóa?", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
        MessageBoxDefaultButton.Button2) ==
        System.Windows.Forms.DialogResult.Yes)
    {
        //Nếu đồng ý xóa
        DataTable dt_Lop=new DataTable();

```

```

//Kiểm tra khóa ngoại trong bảng lớp
SqlDataAdapter da_lop = new SqlDataAdapter("select * from
Lop where makhoa= '" + txtmakhoa.Text + "'", _Connsql);
da_lop.Fill(dt_Lop);
if (dt_Lop.Rows.Count > 0)
{
    MessageBox.Show("Dữ liệu đang sử dụng không thể xóa");
    return;
}
// Nếu cập nhật dữ liệu
DataRow upd_New=ds_QLSV.Tables["Khoa"].Rows.Find (txtMaKhoa.Text);
if (update_New != null)
{
    update_New.Delete();
}
SqlCommandBuilder cmb = new SqlCommandBuilder(da_Khoa);
da_Khoa.Update(ds_QLSV, "Khoa");
MessageBox.Show("Thành công");
}
}

```

2. Bài tập tại lớp

Thiết kế giao diện theo mẫu sau:

Yêu cầu:

- Thiết kế:
 - + Sử dụng `tableLayoutPanel` để thiết kế giao diện như hình vẽ, sao cho nội dung luôn hiển thị ở giữa khi phóng to hay thu nhỏ cửa sổ.
- FormLoad:
 - + Combobox Khoa: hiển thị tất cả khoa trong Bảng Khoa
 - + Combobox Lớp: hiển thị tất cả lớp theo khoa
 - + Datagridview: Hiển thị tất cả sinh viên trong bảng sinh viên
 - + Tất cả textbox bị vô hiệu hóa
 - + Các button Sửa, Xóa, Lưu bị vô hiệu hóa
- Button Tìm kiếm:
 - + Hiển thị danh sách sinh viên theo lớp đã chọn trên combobox lớp
- Khi chọn vào button Thêm:
 - + Các textbox có hiệu lực
 - + Button Lưu có hiệu lực
 - + Textbox Mã sinh viên chờ nhập liệu
- Khi chọn vào Datagridview
 - + Hiển thị thông tin tương ứng lên các textbox
 - + Button lệnh Sửa và Xóa có hiệu lực
- Chọn button Sửa
 - + Button Lưu có hiệu lực
 - + Các textbox có hiệu lực trừ textbox Mã sinh viên
 - + Cho phép sửa các thông tin còn lại
- Khi chọn button Lưu
 - + Kiểm tra thông tin vừa thêm hoặc sửa cho phù hợp
 - + Lưu vào Cơ sở dữ liệu (Lưu ý đang Lưu Thêm hay Sửa)
 - + Thông báo thành công hoặc báo lỗi nếu có.
 - + Button Sửa, Lưu bị vô hiệu hóa
- Khi nhấn button Xóa
 - + Hiển thị thông báo xác nhận
 - + Nếu đồng ý
 - Kiểm tra mã sinh viên xóa có tồn tại trong bảng điểm hay không. Nếu có

thì thông báo “Dữ liệu đang sử dụng không thể xóa ?”.

- Nếu mã sinh viên không tồn tại bên bảng lớp thì xóa bên bảng sinh viên.
- + Hiện thị thông báo nếu xóa thành công hoặc báo lỗi (nếu có)

3. Bài tập về nhà

Thiết kế form Phiếu Nhập hàng như sau:

Mã phiếu nhập	Mã sản phẩm	Đơn giá	Số lượng
PN10112015001	SP001	5400000	1
PN10112015001	SP002	7600000	1
PN10112015002	SP001	5400000	2
PN10112015002	SP002	7600000	2
PN10112015003	SP001	5400000	1

Yêu cầu

– Formload:

Groupbox Phiếu nhập:

- + Textbox Mã phiếu nhập để trống, textbox ngày nhập là ngày hiện tại
- + Con trỏ nhập liệu tại textbox Mã phiếu nhập.
- + Combobox Nhà cung cấp: Chứa tất cả Mã nhà cung cấp của bảng NCC
- + Textbox thành tiền: Bị vô hiệu hóa, hiển thị số 0.

Groupbox Chi tiết phiếu nhập

- + Combobox mã phiếu nhập: Hiển thị tất cả Mã phiếu nhập trong bảng

Phieunhap.

- + Combobox mã sản phẩm: Hiển thị tất cả Mã sản phẩm trong bảng Sanpham.
- + Textbox đơn giá: bị vô hiệu hóa.
- + Số lượng: Hiển thị số 0.

- + DataGridView: Hiển thị thông tin trong bảng chitietphieunhap.
- Thực hiện các chức năng còn lại tương ứng.
- Chọn button “Tạo phiếu nhập”:
 - + Lưu thông tin trong groupbox Phiếu nhập vào table PhieuNhap Lưu ý kiểm tra các dữ liệu đầu vào.
 - + Load lại Combobox Mã phiếu nhập trong groupbox Chi tiết phiếu nhập.
 - + Lưu ý: Không cho thao tác trong textbox thành tiền, lúc này đang là số 0.
- Chọn “combobox Mã phiếu nhập” trong groupbox “Chi tiết Phiếu nhập”
 - + Thông tin về phiếu nhập vừa chọn sẽ được hiển thị lên groupbox “Phiếu nhập”.
 - + Thông tin về Chi tiết phiếu nhập tương ứng hiển thị lên datagridview ChiTietPhieuNhap.
- Chọn “Combobox Mã sản phẩm”:
 - + Tự động hiển thị đơn giá tương ứng với mã sản phẩm vừa chọn.
- Gõ số lượng vào “textbox số lượng”:
 - + Tự động cập nhật thành tiền trên “textbox thành tiền” ứng với Phiếu nhập đang chọn.
- Chọn “Thêm sản phẩm”
 - + Kiểm tra Mã sản phẩm trong combobox “Mã sản phẩm” đã có trong “ChiTietPhieuNhap” đang chọn hay chưa. Nếu chưa có thì Lưu thông tin trong groupbox “Chi tiết phiếu nhập” vào bảng Chitietphieunhap. Ngược lại cập nhật số lượng.
 - + Cập nhật Thành tiền vào trong bảng Phieunhap.
- Khi chọn dataGridView Chi tiết phiếu nhập:
 - + Hiển thị thông tin vừa chọn lên các control trong groupbox Chi tiết phiếu nhập.
 - + DataGridView Chi tiết phiếu nhập cho hiển thị ContextMenuStrip có các chức năng Sửa, Xóa Chi tiết phiếu nhập.

-----Hết-----