# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №4**

**«Розробка супутнього програмного забезпечення до**

**Бази даних коміркового складу»**

**з дисципліни «Конструювання програмного забезпечення»**

Виконав:

слухач групи ПЗС-21

Гринчук Тарас

Прийняв:

доц. Тушницький Р.Б.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

**Мета роботи:** Засвоїти основні технології, що дозволяють опрацьовувати дані СУБД.

## Хід роботи

Створимо в MS Visual Studio новий проект «Windows Forms». Додамо в проект модель ADO.NET Entity Framework (рис. 1):

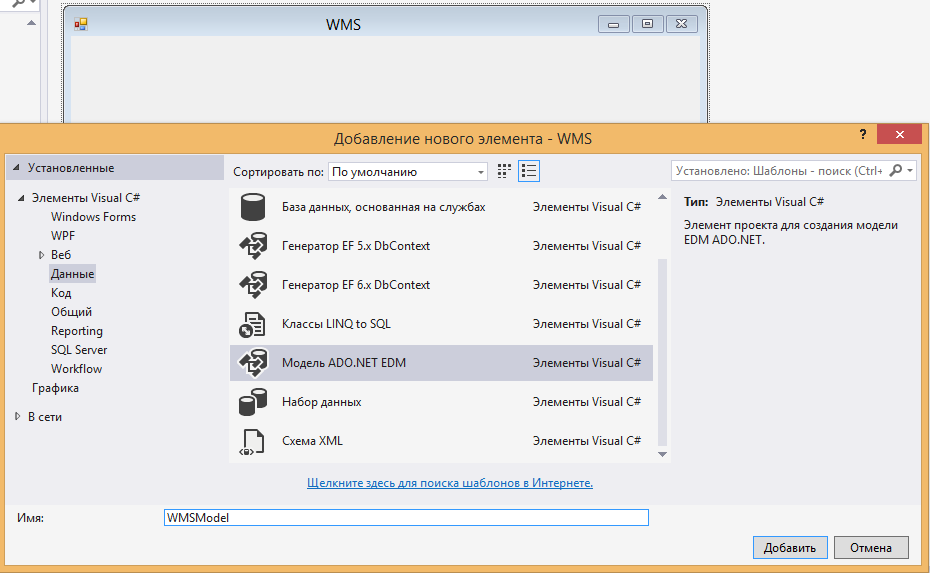


Рис. 1. Додавання моделі ADO.NET Entity Framework

За домогою майста EDM підключимось до нашої бази та виберемо потрібні таблиці (рис. 2).

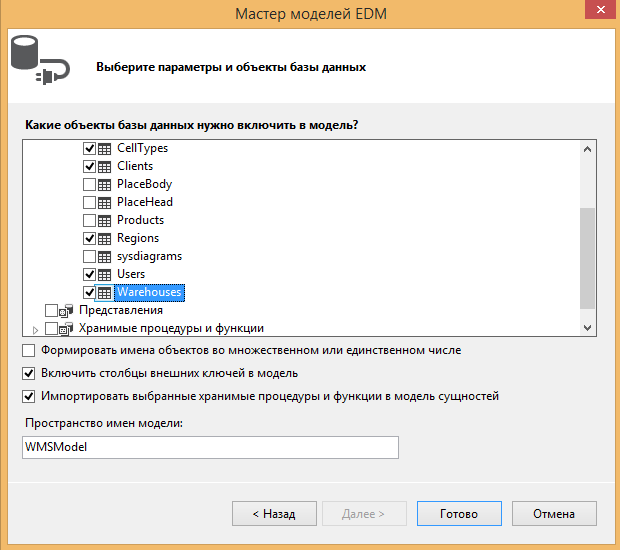


Рис. 2. Майстер EDM

Перебудуємо рішення та на панелі «Джерел даних» створимо нове джерело (рис. 3).

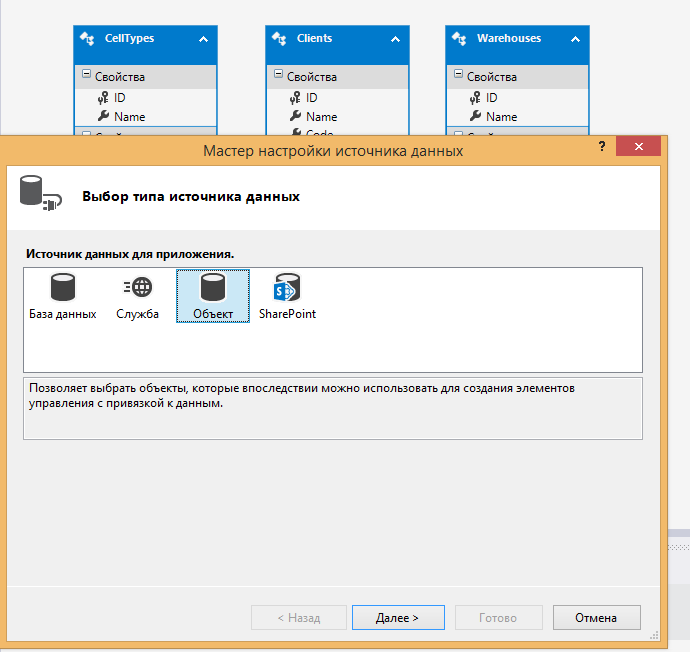


Рис. 3. Майстер джерел даних

Додамо таблицю Clients на форму (рис. 4):

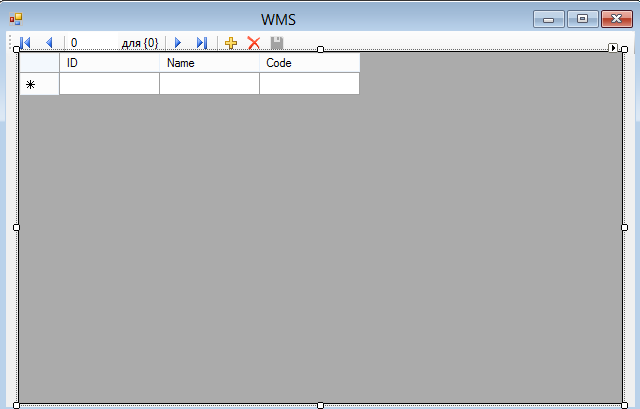


Рис. 4. Таблиця Clients

Додамо в модуль форми наступний код:

namespace WMS

{

public partial class Form1 : Form

{

private WMSEntities context;

public Form1()

{

InitializeComponent();

}

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

{

context = new WMSEntities();

var query = context.Clients.OrderBy(p => p.ID);

clientsBindingSource.DataSource = query.ToList();

}

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

{

context.SaveChanges();

}

}

}

Запустимо програму на виконання (рис. 5):

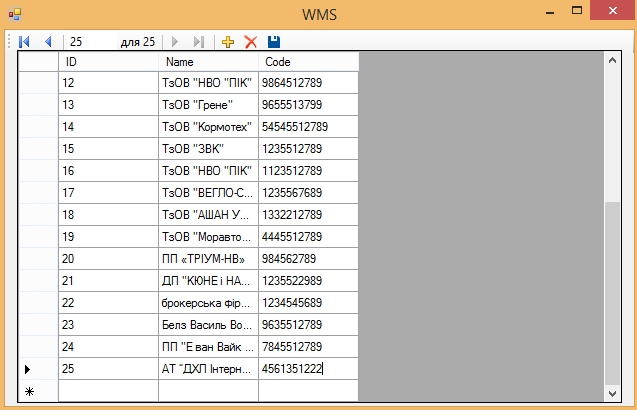


Рис. 5. Головне вікно програми

Створено головне меню, та додамо виклики нових форм для інших таблиць БД. Програма набуде вигляду (рис. 6):

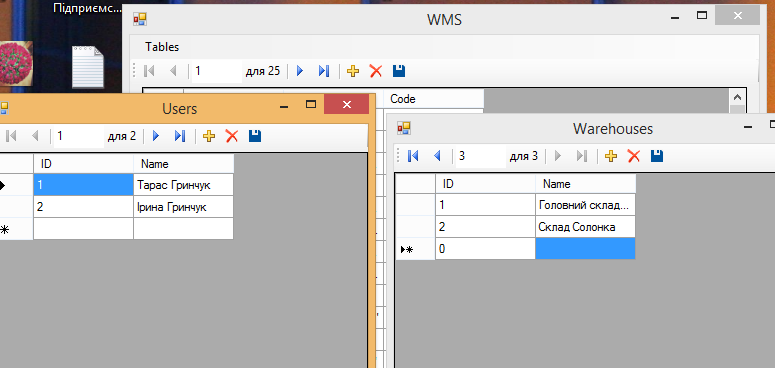


Рис. 6. Форми для таблиць Users та Warehouses

Додамо в таблицю Clients можливість додавати та видадяти рядки з БД:

//збереження змін при виході з комірки

private void clientsDataGridView\_CellEndEdit(object sender, DataGridViewCellEventArgs e)

{

context.SaveChanges();

}

//отримання останнього ID таблмці

private int LastID()

{

int res = 0;

using (WMSEntities cont = new WMSEntities())

{

var query = cont.Clients.OrderBy(p => p.ID);

if (query.Count() > 0) res = query.ToList().Last().ID;

}

return res;

}

//додавання нового рядка

private void button1\_Click(object sender, EventArgs e)

{

using (WMSEntities cont = new WMSEntities())

{

Clients t = new Clients() { ID = 1 + LastID(), Name = textBox2.Text, Code = textBox1.Text};

cont.Clients.Add(t);

cont.SaveChanges();

}

Form1\_Load(null, null);

}

//видалення рядка

private void button2\_Click(object sender, EventArgs e)

{

HashSet<int> rows = new HashSet<int>();

foreach (var row in clientsDataGridView.SelectedCells)

{

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

rows.Add((int)clientsDataGridView[0, RowIndex].Value);

}

foreach (int row in rows)

{

using (WMSEntities cont = new WMSEntities())

{

var query = from t in cont.Clients

where t.ID == row

select t;

if (query.Count() > 0)

{

var t = query.ToList().First();

cont.Clients.Remove(t);

cont.SaveChanges();

}

}

}

Form1\_Load(null, null);

## }

Спробуємо внести зміни в цю таблицю (рис. 7):

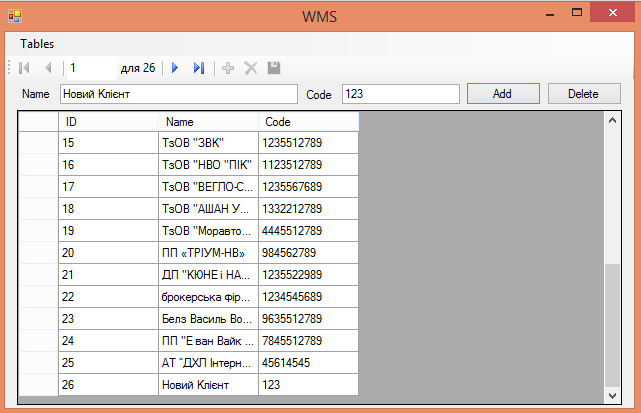


Рис. 7. Додавання рядка в таблицю Clients

Додамо в проект таблицю Products, яка має посилання на таблицю Clients (рис. 8):

## 

Рис. 8. Таблиця Products

Натиснемо на кнопку Add та додамо новий товар в базу даних (рис. 9):

## 

Рис. 10. Форма для створення нового товару

Заповнимо дану форму та виберемо, для цього товару клієнта-власника(Client) та Область розміщення(Region) (рис. 11):

## 

Рис. 11. Створення нового товару

Натиснемо на «ОК» та переглянемо новий товар в таблиці товарів (рис. 12):

## 

Рис. 12. Новий товар в таблиці товарів

Спробуємо тепер відредагувати існуючий товар, натиснувши – Edit. З’явиться те саме вікно, що й при сторенні нового, але на цей раз, буде заповнене поточними даними товару (рис. 13):

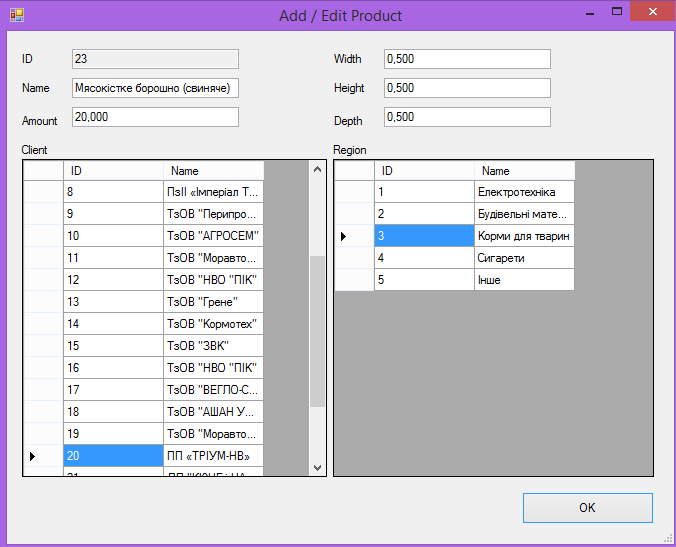


Рис. 12. Форма редагування товару

Код даної форми:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Entity.Migrations;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WMS

{

public partial class AddEditProduct : Form

{

private WMSEntities context;

private bool editMode;

//конструктор - для нового товару

public AddEditProduct()

{

editMode = false;

InitializeComponent();

IDText.Text = (1 + LastID()).ToString();

}

//конструктор - для редагування існуючого товару

public AddEditProduct(String ID)

{

editMode = true;

InitializeComponent();

IDText.Text = ID;

}

//кнопка "ОК"

private void button1\_Click(object sender, EventArgs e)

{

//перевірка кокектності заповнення

if (NameText.Text.Trim().Length == 0)

{

MessageBox.Show("Name is empty !");

return;

}

if (AmountText.Text.Trim().Length == 0)

{

MessageBox.Show("Amount is empty !");

return;

}

if (WidthText.Text.Trim().Length == 0)

{

MessageBox.Show("Width is empty !");

return;

}

if (HeightText.Text.Trim().Length == 0)

{

MessageBox.Show("Height is empty !");

return;

}

if (DepthText.Text.Trim().Length == 0)

{

MessageBox.Show("Depth is empty !");

return;

}

decimal nAmount;

if (!decimal.TryParse(AmountText.Text.Trim(), out nAmount))

{

MessageBox.Show("Amount: bad format of number !");

return;

}

decimal nWidth;

if (!decimal.TryParse(WidthText.Text.Trim(), out nWidth))

{

MessageBox.Show("Width: bad format of number !");

return;

}

decimal nHeight;

if (!decimal.TryParse(HeightText.Text.Trim(), out nHeight))

{

MessageBox.Show("Height: bad format of number !");

return;

}

decimal nDepth;

if (!decimal.TryParse(DepthText.Text.Trim(), out nDepth))

{

MessageBox.Show("Depth: bad format of number !");

return;

}

if (clientsDataGridView.SelectedCells.Count == 0)

{

MessageBox.Show("Choose the Client !");

return;

}

if (regionDataGridView.SelectedCells.Count == 0)

{

MessageBox.Show("Choose the Region !");

return;

}

var row = clientsDataGridView.SelectedCells[0];

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int nClientID = (int)clientsDataGridView[0, RowIndex].Value;

row = regionDataGridView.SelectedCells[0];

RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int nRegionID = (int)regionDataGridView[0, RowIndex].Value;

using (WMSEntities cont = new WMSEntities())

{

Products t = new Products

{

ID = int.Parse(IDText.Text),

Name = NameText.Text.Trim(),

Amount = nAmount,

Width = nWidth,

Height = nHeight,

Depth = nDepth,

ClientID = nClientID,

RegionID = nRegionID,

};

cont.Products.AddOrUpdate(t);

cont.SaveChanges();

}

DialogResult = DialogResult.OK;

Close();

}

//при запуску форми - наповнення вмістом з БД

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

{

context = new WMSEntities();

context.Configuration.LazyLoadingEnabled = false;

//var query = context.Clients.OrderBy(p => p.ID);

var query =

from client in context.Clients

select new

{

client.ID,

client.Name,

};

clientsDataGridView.DataSource = query.ToList();

query =

from region in context.Regions

select new

{

region.ID,

region.Name,

};

regionDataGridView.DataSource = query.ToList();

if (editMode)

{

int nID = int.Parse(IDText.Text);

using (WMSEntities cont = new WMSEntities())

{

var q =

from products in cont.Products

where products.ID == nID

select products;

Products first = q.ToList().First();

NameText.Text = first.Name;

AmountText.Text = first.Amount.ToString();

WidthText.Text = first.Width.ToString();

HeightText.Text = first.Height.ToString();

DepthText.Text = first.Depth.ToString();

for (int i = 0; i < clientsDataGridView.RowCount; i++)

{

int nClientID = (int)clientsDataGridView[0, i].Value;

if (nClientID == first.ClientID)

{

clientsDataGridView[0, i].Selected = true;

break;

}

}

for (int i = 0; i < regionDataGridView.RowCount; i++)

{

int nRegionID = (int)regionDataGridView[0, i].Value;

if (nRegionID == first.RegionID)

{

regionDataGridView[0, i].Selected = true;

break;

}

}

}

}

}

private int LastID()

{

int res = 0;

using (WMSEntities cont = new WMSEntities())

{

var query = cont.Products.OrderBy(p => p.ID);

if (query.Count() > 0) res = query.ToList().Last().ID;

}

return res;

}

}

}

Створимо форму для списку комірок (Cells) (рис. 14):

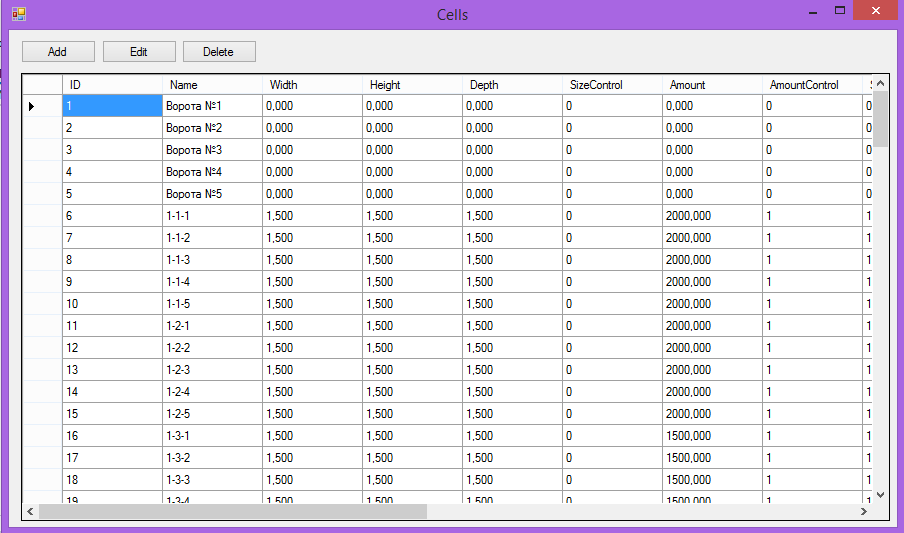


Рис. 14. Таблиця Cells (комірки)

Натиснемо на кнопку «Edit» та відредагуємо вибрану комірку (рис. 15):

## 

Рис. 15. Редагування комірки

## ВИСНОВКИ

На даній лабораторній роботі засвоїв основні технології, що дозволяють опрацьовувати дані СУБД.