# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №3**

**«Доступ до баз даних за допомогою технологій ADO.NET,**

**Linq to SQL та ADO.NET Entity Framework»**

**з дисципліни «Конструювання програмного забезпечення»**

Виконав:

слухач групи ПЗС-21

Гринчук Тарас

Прийняв:

ст. викл. Тушницький Р.Б.

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

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

ЛЬВІВ – 2014

**Тема роботи**: Доступ до баз даних за допомогою технологій ADO.NET,

Linq to SQL та ADO.NET Entity Framework.

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

## Хід роботи

**Завдання:**

Відповідно до варіанту проекту з баз даних

1) вивести на консоль або UI дані із таблиць. Дані вибирати відповідно до списку сценаріїв та планованого графічного інтерфейсу.

2) Реалізувати додавання, оновлення та видалення даних

Реалізувати це використовуючи різні технології доступу:

* ADO.NET.
* LinqToSql
* ADO.NET Entity Framework.

Зробити порівняльний аналіз цих технологій

В середовищі MS Visual Studio створимо новий консольний проект наберемо текст програми:

using System;

using System.Data.SqlClient;

using System.Data;

namespace Students

{

class Program

{

static void Main(string[] args)

{

//Запит для виводу таблиці студентів

string RequestText = "Select \* From Students";

//Запит для виводу таблиці оцінок студента з кодом 10

string RequestText2 =

"Select " +

"st.ID as ID, " +

"st.StudName as StudName, " +

//"st.GroupNum as GroupNum, " +

"mar.Mark as Mark " +

"FROM Marks mar " +

"Join Students st on st.ID = mar.ID " +

"WHERE st.ID = 10";

//стрічка підключення до бази

string connectionString = "server=TarasHP;uid=sa;pwd=pass45678;database=Test";

SqlConnection conn = new SqlConnection(connectionString);

try

{

//спроба підключитися

conn.Open();

}

catch (SqlException se)

{

Console.WriteLine("Помилка пiдключення:{0}", se.Message);

Console.ReadKey();

return;

}

Console.WriteLine("Успiшно пiдключено до бази");

/\* Створюємо екземпляр класу SqlCommand по імені cmdCreateTable

              і передаємо конструктору цього класу, запит на створення таблиці

              і об'єкт типу SqlConnection

             \*/

//створення таблиць

CreateNewTable(conn, "CREATE TABLE " +

" Students (ID int not null" +

", StudName char(60) not null," +

" GroupNum char(20) not null)");

CreateNewTable(conn, "CREATE TABLE " +

" Marks (ID int not null, Mark int not null)");

Console.WriteLine("Таблицi створенi успiшно");

//додавання нових записiв

Console.WriteLine("Вставляємо записи:");

InsertToTable(conn, 10, "Гринчук Т.", "ПЗС-11");

InsertToTable(conn, 20, "Непомнящий Iгор", "ПЗС-11");

InsertToTable(conn, 30, "Середа Iгор", "ПЗС-11");

//Оцiнки

InsertMarks(conn, 10, 5);

InsertMarks(conn, 10, 4);

InsertMarks(conn, 10, 5);

InsertMarks(conn, 20, 5);

InsertMarks(conn, 20, 3);

InsertMarks(conn, 30, 4);

Console.ReadKey();

//виводимо на екран

ShowFields(conn, RequestText);

//змiна значення рядка

UpdateFromTable(conn);

ShowFields(conn, RequestText);

Console.WriteLine("Оцiнки студента з ID = 10 :");

ShowFields(conn, RequestText2);

Console.ReadKey();

//видалення данних

DeleteFromTable(conn);

ShowFields(conn, RequestText);

//видалення таблиць

DeleteTable(conn, "Students");

DeleteTable(conn, "Marks");

Console.ReadKey();

//закриття пiдключення до бази

conn.Close();

conn.Dispose();

}

private static void DeleteTable(SqlConnection conn, string table)

{

using (SqlCommand cmdDeleteTable = new SqlCommand("DROP TABLE " + table, conn))

{

//вiдправка запиту

try

{

cmdDeleteTable.ExecuteNonQuery();

}

catch

{

Console.WriteLine("Помилка видалення таблицi");

Console.ReadKey();

return;

}

}

Console.WriteLine("Таблиця " + table + " успiшно видалена");

}

private static void DeleteFromTable(SqlConnection conn)

{

using (SqlCommand cmd = new SqlCommand("Delete From Students" +

" where ID = @ID", conn))

{

//Створення параметрiв

SqlParameter param = new SqlParameter();

//задаємо назву параметра

param.ParameterName = "@ID";

//задаємо значення параметра

param.Value = 10;

//задаємо тип параметра

param.SqlDbType = SqlDbType.Int;

//передаємо параметр екземпляру класу SqlCommand

cmd.Parameters.Add(param);

param = new SqlParameter();

param.ParameterName = "@StudName";

param.Value = "Иванов Иван";

param.SqlDbType = SqlDbType.Text;

cmd.Parameters.Add(param);

Console.WriteLine("Видалення запису");

try

{

cmd.ExecuteNonQuery();

}

catch

{

Console.WriteLine("Помилка видалення запису");

Console.WriteLine("Можливе запис вже був видалений");

Console.ReadKey();

return;

}

Console.ReadKey();

}

}

private static void UpdateFromTable(SqlConnection conn)

{

using (SqlCommand cmd = new SqlCommand("Update Students" +

" Set StudName = @StudName where ID = @ID", conn))

{

SqlParameter param = new SqlParameter();

param.ParameterName = "@ID";

param.Value = 10;

param.SqlDbType = SqlDbType.Int;

cmd.Parameters.Add(param);

param = new SqlParameter();

param.ParameterName = "@StudName";

param.Value = "Гринчук Тарас";

param.SqlDbType = SqlDbType.Text;

cmd.Parameters.Add(param);

Console.WriteLine("Змiнюємо записи");

{

try

{

cmd.ExecuteNonQuery();

}

catch (SqlException se)

{

Console.WriteLine("Помилка, при змiнi запису: {0}",se.Message);

Console.ReadKey();

return;

}

Console.ReadKey();

}

}

}

private static void ShowFields(SqlConnection conn, string text)

{

//Виводимо значення на екран

Console.WriteLine("================================= База даних =================================");

using (SqlCommand cmd = new SqlCommand(text, conn))

{

/\* Метод ExecuteReader() класу SqlCommand повертає

                  об'єкт типу SqlDataReader, з допомогою якого ми можемо

                  прочитати всі рядки, поверненні в результаті виконання запиту

                  CommandBehavior.CloseConnection - закриваємо з'єднання після запиту

                  \*/

using (SqlDataReader dr = cmd.ExecuteReader())

{

//цикл по всіх стовпцях отриманої в результаті запиту таблиці

for (int i = 0; i < dr.FieldCount; i++)

/\* метод GetName() класу SqlDataReader дозволяє отримати ім'я стовпця

                          за номером, який передається як параметр, даному методу

                          і позначає номер стовпчика в таблиці (починається з 0)

                          \*/

Console.Write("{0}\t", dr.GetName(i).ToString().Trim());

/\* читаємо дані з таблиці

                      читання відбувається тільки в прямому напрямку

                      всі прочитання рядка відкидаються \*/

Console.WriteLine();

while (dr.Read())

{

/\* метод GetValue() класу SqlDataReader дозволяє отримати значення стовпця

                                                 за номером, який передається як параметр, даному методу

                                                 і позначає номер стовпчика в таблиці (починається з 0)

                                                 \*/

Console.WriteLine("{0}\t{1}\t{2}", dr.GetValue(0).ToString().Trim(),

dr.GetValue(1).ToString().Trim(),

dr.GetValue(2).ToString().Trim());

}

}

}

Console.WriteLine("==============================================================================");

Console.ReadKey();

}

private static void CreateNewTable(SqlConnection conn, string text)

{

using (SqlCommand cmdCreateTable = new SqlCommand(text, conn))

{

//вiдправка запиту

try

{

cmdCreateTable.ExecuteNonQuery();

}

catch (SqlException se)

{

Console.WriteLine("Помилка створення таблицi: {0}",se.Message);

Console.ReadKey();

return;

}

}

}

private static void InsertToTable(SqlConnection conn, int ID, string StudName, string GroupNum)

{

using (SqlCommand cmd = new SqlCommand("Insert into Students" +

"(ID,StudName,GroupNum) Values (@ID,@StudName,@GroupNum)", conn))

{

SqlParameter param = new SqlParameter();

param.ParameterName = "@ID";

param.Value = ID;

param.SqlDbType = SqlDbType.Int;

cmd.Parameters.Add(param);

param = new SqlParameter();

param.ParameterName = "@StudName";

param.Value = StudName;

param.SqlDbType = SqlDbType.Text;

cmd.Parameters.Add(param);

param = new SqlParameter();

param.ParameterName = "@GroupNum";

param.Value = GroupNum;

param.SqlDbType = SqlDbType.Text;

cmd.Parameters.Add(param);

try

{

cmd.ExecuteNonQuery();

}

catch

{

Console.WriteLine("Помилка вставки запису");

Console.ReadKey();

return;

}

}

}

private static void InsertMarks(SqlConnection conn, int ID, int Mark)

{

using (SqlCommand cmd = new SqlCommand("Insert into Marks" +

"(ID,Mark) Values (@ID,@Mark)", conn))

{

SqlParameter param = new SqlParameter();

param.ParameterName = "@ID";

param.Value = ID;

param.SqlDbType = SqlDbType.Int;

cmd.Parameters.Add(param);

param = new SqlParameter();

param.ParameterName = "@Mark";

param.Value = Mark;

param.SqlDbType = SqlDbType.Int;

cmd.Parameters.Add(param);

try

{

cmd.ExecuteNonQuery();

}

catch

{

Console.WriteLine("Помилка вставки запису");

Console.ReadKey();

return;

}

}

}

}

}

Запустимо дану програму на виконання (рис. 1):

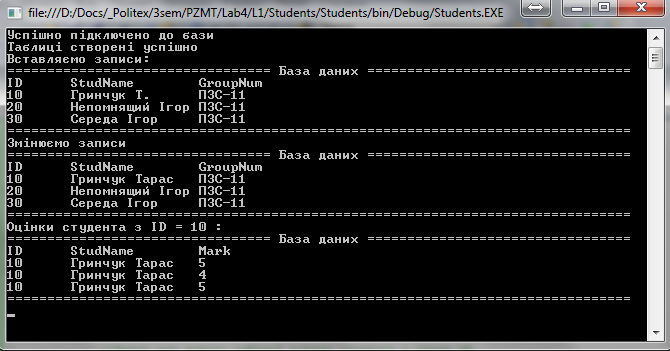


Рис. 1. Вікно програми

Як бачимо програма вміє створювати таблиці бази даних в SQL Server. Змінювати їх вміст, додавати стрічки, виводити результати запитів до декількох таблиць бази даних. Що пересвідчитись у правильності виконання програми відкриємо створену БД в SQL Server (рис. 2):

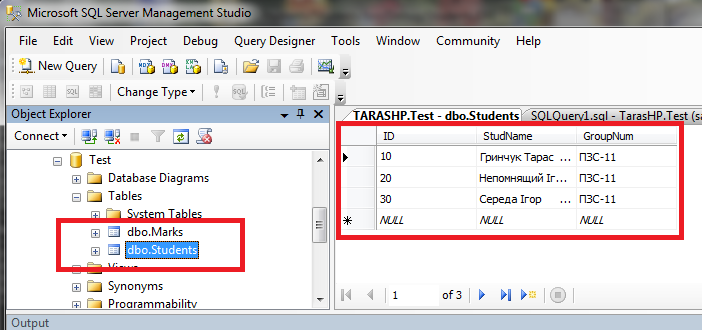


Рис. 2. Вигляд БД у SQL Server

Також програмно можна видалити певну стрічку і таблиці взагалі (рис. 3 - 4):

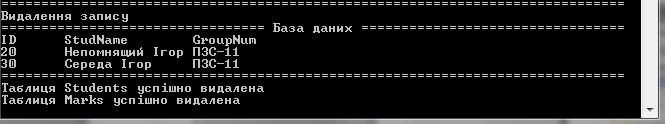


Рис. 3. Вікно програми

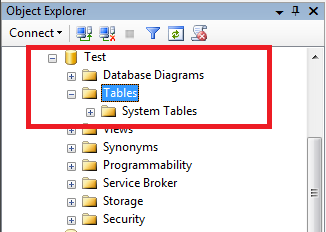


Рис. 4. Вигляд пустої бази даних після завершення роботи програми

## Для демонстрації роботи технології Linq to sql створимо пустий ASP.NET проект, з наступним кодом основної Web-форми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Windows.Forms;

namespace WebApplication5

{

public partial class WebForm1 : System.Web.UI.Page

{

//всі працівники та відділи

protected void Page\_Load(object sender, EventArgs e)

{

SampleDataContext dbContext = new SampleDataContext();

GridView1.DataSource =

dbContext.Employees;

GridView1.DataBind();

GridView2.DataSource =

dbContext.Departments;

GridView2.DataBind();

}

//тільки чоловіки

protected void Button1\_Click(object sender, EventArgs e)

{

SampleDataContext dbContext = new SampleDataContext();

GridView1.DataSource =

from employee in dbContext.Employees

where employee.Gender == "Male"

orderby employee.Salary descending

select employee;

GridView1.DataBind();

}

//Працівнити та професії

protected void Button2\_Click(object sender, EventArgs e)

{

SampleDataContext dbContext = new SampleDataContext();

GridView1.DataSource =

from employee in dbContext.Employees

join department in dbContext.Departments on employee.DepartmentId equals department.ID

select new { employee.FirstName, employee.LastName,

department.Name,

department.Location

};

GridView1.DataBind();

}

//Жінки в IT

protected void Button3\_Click(object sender, EventArgs e)

{

SampleDataContext dbContext = new SampleDataContext();

GridView1.DataSource =

from employee in dbContext.Employees

join department in dbContext.Departments on employee.DepartmentId equals department.ID

where department.Name == "IT" && employee.Gender == "Female"

select new

{

employee.FirstName,

employee.LastName,

department.Name,

department.Location

};

GridView1.DataBind();

}

//Додати запис

protected void Button4\_Click(object sender, EventArgs e)

{

using (SampleDataContext dbContext = new SampleDataContext())

{

Departments dep = new Departments();

dep.Location = "Ukraine";

dep.Name = "Support";

dbContext.Departments.InsertOnSubmit(dep);

try

{

dbContext.SubmitChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

GridView2.DataBind();

}

//Видалення

protected void Button5\_Click(object sender, EventArgs e)

{

SampleDataContext dbContext = new SampleDataContext();

var deps = (

from department in dbContext.Departments

where department.Name == "Support"

select department);

if(deps.Count() == 0) return;

var dep = deps.First();

dbContext.Departments.DeleteOnSubmit(dep);

try

{

dbContext.SubmitChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

GridView2.DataBind();

}

}

## }

## Запустимо проект на виконання (рис. 5):

## 

## Рис. 5. Проект з використанням Linq to SQL

## Натиснувши кнопку «Жінки в ІТ» отримаємо відповідну вибірку з БД (рис.6):

## 

## Рис 6. Вибірка з БД за допомогою Linq to SQL

## ВИСНОВКИ

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