**Виконали:** Свобода Максим, Марусяк Олег

**Група:** КН-308

**Кафедра:** САП

**Дисципліна:** Проектування інформаційних систем

**Перевірив:** Василишин Б.С.

**ЗВІТ**

до лабораторної роботи №3

на тему: Робота з БД з використанням Connected ADO.NET.

Мета роботи

Ознайомитись зі структурою класів ADO.NET для роботи з БД. Здобути практичні навички роботи з класами ADO.NET, що забезпечують доступ і роботу з БД, зокрема DbConnection, DBCommand, DbDataAdapter та інші.

Індивідуальне завдання:

1. Побудувати стрічку зв’язку для під’єднання до існуючої бази даних, що відповідає структурі індивідуального завдання.
2. Забезпечити збереження і доступ до стрічки зв’язку через конфігураційний файл.
3. Для кожної з таблиць створити об’єкти DbCommand для читання, редагування, додавання та видалення даних.
4. Використавши створені в пункті 3 команди редагувати одну з існуючих таблиць, а саме: додати 3-5 рядків, модифікувати 2-3 рядки, видалити 2 рядки. Дані з таблиці після редагування вивести на екран.
5. Для кожної з таблиць створити відповідні об’єкти DbDataAdapter.
6. Сформувати DataSet, який відповідає структурі індивідуального завдання. Заповнити цей DataSet інформацією з БД використавши метод Fill.
7. Внести зміни в 2-3 таблиці з створеного в п.6 DataSet. Використавши метод Update внести ці зміни в БД і вивести інформацію з відредагованих таблиць на екран.

Завдання 1: побудувати стрічку зв’язку.

В ADO.net з’єднання з базою даних відбувається багатьма способами, проте для даної лабораторної роботи ми повинні були використати стрічку зв’язку. Вона виглядає наступним чигном:

connectionString="server=localhost; initial catalog=Hospital; integrated security=true" name="constr"

Server - назва користувача, що користується бд(localhost - для користувача, що має базу даних на своєму ПК)

Initial catalog - назва бази даних, з якою відбуватиметься робота.

Integrated security - дозвіл на доменне з’єднання з базою даних.

Name - назва змінної, що буде використовуватись в коді.

Завдання 2: Створити і записати в нього конфігураційний файл.

Для збереження стрічки ми створили конфігураційний файл App.config

Виглядає він наступним чином:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add connectionString="server=localhost; initial catalog=Hospital; integrated security=true" name="constr"/>

</connectionStrings>

</configuration>

Виклик стрічки через конфігураційний файл відбуватиметься через Configuration manager, з бібліотеки System.Configuration;

con = new SqlConnection(ConfigurationManager.ConnectionStrings["constr"].ToString());

Завдання 3: Для кожної з таблиць створити читання , редагування, додавання та видалення даних.

Отже, в ADO.net використовуються об’єкти класу DbCommand для взаємодії з таблицями, проте в нашому випадку доцільніше було б використати DataSet, або Query для комфортної роботи з даними. Так ми і зробили, функції, що реалізовують заданий функціонал використовують DbCommand та SqlQuery для роботи з даними:

public static void AlterDataTables(string tableName, string operation)

{

switch (tableName)

{

case "dbo.Equipment":

string name = "";

string manuf = "";

string command = "";

int rowid = 0;

int quant = 0;

if(operation == "add")

{

Console.WriteLine("Enter Name,Quantity and manufacturer");

name = Console.ReadLine();

quant = Convert.ToInt32(Console.ReadLine());

manuf = Console.ReadLine();

command = string.Format("insert into " + tableName + " values(N'{0}',{1},N'{2}');", name, quant, manuf);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "alter")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Name,Quantity and manufacturer");

name = Console.ReadLine();

quant = Convert.ToInt32(Console.ReadLine());

manuf = Console.ReadLine();

command = string.Format("Update Equipment SET Name = N'{0}', Quantity = {1}, Manufacturer = N'{2}' where IDeq = {3};", name, quant, manuf, rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "delete")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

command = string.Format("DELETE FROM Equipment WHERE IDeq = {0};",rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

break;

case "dbo.Medicine":

string crdate = "";

string expdate = "";

if(operation == "add")

{

Console.WriteLine("Введіть Назву, Виробника і Дати виготовлення/Термін придатності у форматі рік-місяць-число ");

name = Console.ReadLine();

manuf = Console.ReadLine();

crdate = Console.ReadLine();

expdate = Console.ReadLine();

command = string.Format("insert into " + tableName + " values(N'{0}',N'{1}','{2}','{3}');", name, manuf, crdate, expdate);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть Назву, Виробника і Дати виготовлення/Термін придатності у форматі рік-місяць-число ");

name = Console.ReadLine();

manuf = Console.ReadLine();

crdate = Console.ReadLine();

expdate = Console.ReadLine();

command = string.Format("update " + tableName + "SET Name = N'{0}', Producer = N'{1}', Created\_Date = '{2}', Expiration\_Date = '{3}' where IDmed = {4});", name, manuf, crdate, expdate,rowid);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "delete")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

command = string.Format("DELETE FROM Medicine WHERE IDmed = {0};", rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

break;

case "dbo.Patient":

string surname = "";

string middlename = "";

int age = 0;

string disease = "";

string ardate = "";

string disdate = "";

string ward = "";

if (operation == "add")

{

Console.WriteLine("Введіть ПІБ, Вік, Хворобу, Дати прибуття/виписки у форматі рік-місяць-число, а також відділення ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

disease = Console.ReadLine();

ardate = Console.ReadLine();

disdate = Console.ReadLine();

if (disdate == "")

disdate = "NULL";

ward = Console.ReadLine();

command = string.Format("insert into " + tableName + " values(N'{0}',N'{1}',N'{2}',{3},N'{4}','{5}',{6},N'{7}');", surname, name, middlename, age, disease, ardate, disdate, ward);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ПІБ, Вік, Хворобу, Дати прибуття/виписки у форматі рік-місяць-число, а також відділення ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

disease = Console.ReadLine();

ardate = Console.ReadLine();

disdate = Console.ReadLine();

if (disdate == "")

disdate = "NULL";

ward = Console.ReadLine();

command = string.Format("update " + tableName + "SET Last\_name = N'{0}', First\_Name = N'{1}', Middle\_Name = N'{2}', Age = {3}, Disease = '{4}'" +

", Arrival\_Date = '{5}, Discharge\_Date = '{6}', Ward = '{7}' where IDpat = {8} ');",surname,name,middlename,age,disease,ardate,disdate,rowid);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "delete")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

command = string.Format("DELETE FROM Patient WHERE IDpat = {0};", rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

break;

case "dbo.Staff":

if(operation == "add")

{

Console.WriteLine("Введіть ПІБ, Вік і Посаду ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

string position = Console.ReadLine();

command = string.Format("insert into " + tableName + " values(N'{0}',N'{1}',N'{2}',{3},N'{4}');", surname, name, middlename, age, position);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ПІБ, Вік і Посаду ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

string position = Console.ReadLine();

command = string.Format("update " + tableName + "Set Last\_Name = N'{0}', First\_Name = N'{1}', Middle\_Name = N'{2}', Age = {3}, Position = N'{4}');", surname, name, middlename, age, position,rowid);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if( operation == "delete")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

command = string.Format("DELETE FROM Staff WHERE IDstaff = {0};", rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

break;

case "dbo.Proced":

int patid = 0;

int equipid = 0;

int staffid = 0;

int medid = 0;

int price = 0;

if (operation == "add")

{

Console.WriteLine("Введіть ID пацієнта, спорядження, персоналу та ліків, назву та ціну процедури: ");

patid = Convert.ToInt32(Console.ReadLine());

equipid = Convert.ToInt32(Console.ReadLine());

staffid = Convert.ToInt32(Console.ReadLine());

medid = Convert.ToInt32(Console.ReadLine());

name = Console.ReadLine();

price = Convert.ToInt32(Console.ReadLine());

command = string.Format("insert into " + tableName + " values({0},{1},{2},{3},N'{4}',{5});", patid, equipid, staffid, medid, name, price);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ID пацієнта, спорядження, персоналу та ліків, назву та ціну процедури: ");

patid = Convert.ToInt32(Console.ReadLine());

equipid = Convert.ToInt32(Console.ReadLine());

staffid = Convert.ToInt32(Console.ReadLine());

medid = Convert.ToInt32(Console.ReadLine());

name = Console.ReadLine();

price = Convert.ToInt32(Console.ReadLine());

command = string.Format("update " + tableName + "Patient\_ID = {0}, Equipment\_ID = {1}, Staff\_ID = {2}, Medicine\_ID = {3}," +

"Name = N'{4}', Price = {5} where IDproc = {6});", patid, equipid, staffid, medid, name, price,rowid);

cmd = new SqlCommand(command, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

if(operation == "delete")

{

con.Open();

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

command = string.Format("DELETE FROM Proced WHERE IDproc = {0};", rowid);

cmd = new SqlCommand(command, con);

cmd.ExecuteNonQuery();

con.Close();

}

break;

default: Console.WriteLine("Такої таблиці не існує!"); Console.Clear(); break;

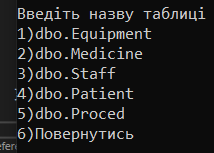
}

}

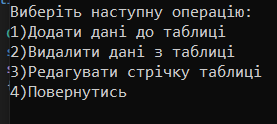
Завдання 4:

Давайте використаємо функціонал нашої програми. Додамо 3 стрічки в будь яку з таблиць, потім відредагуємо інші рядки, а потім видалимо 2.

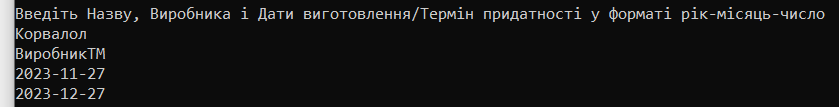
Для початку, оберемо таблицю:



Нехай це буде таблиця Ліки,



Обираємо “Додати дані до таблиці”,

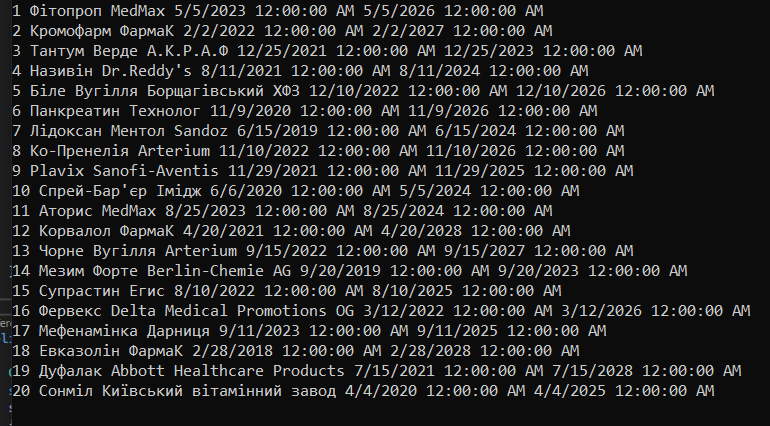


Після введення даний і зчитування їх системою, відбувається перевірка на коректність. Після успішної перевірки отримуємо

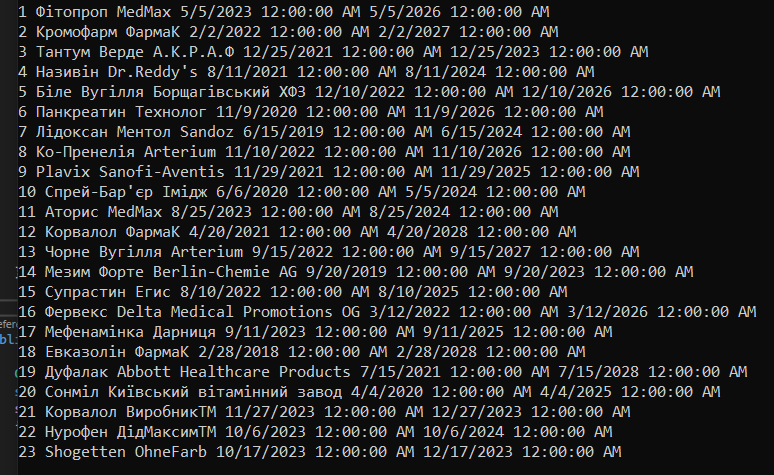


Аналогічно для інших двох записів.

Початкова таблиця Ліки:

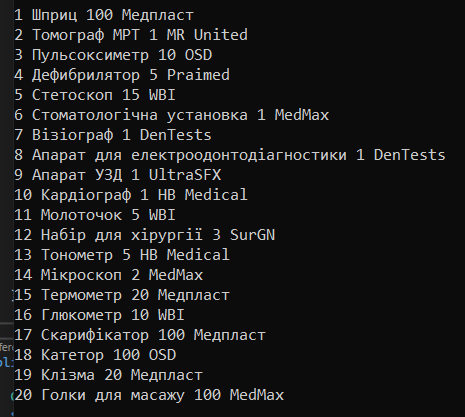


Змінена таблиця:



Чудово, тепер, змінимо кілька рядків даних, скажімо в таблиці Спорядження:

Таблиця спорядження до змін:



Змінимо к-сть молоточків WBI,



Бачимо, що молоточки у нас під номером 11.

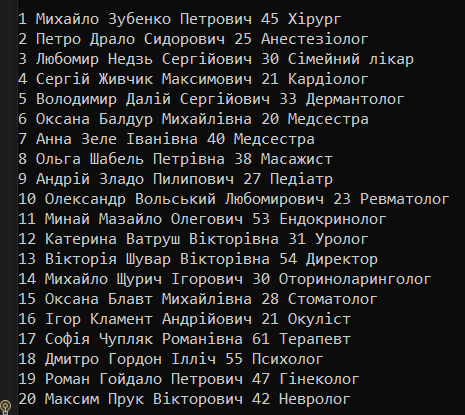


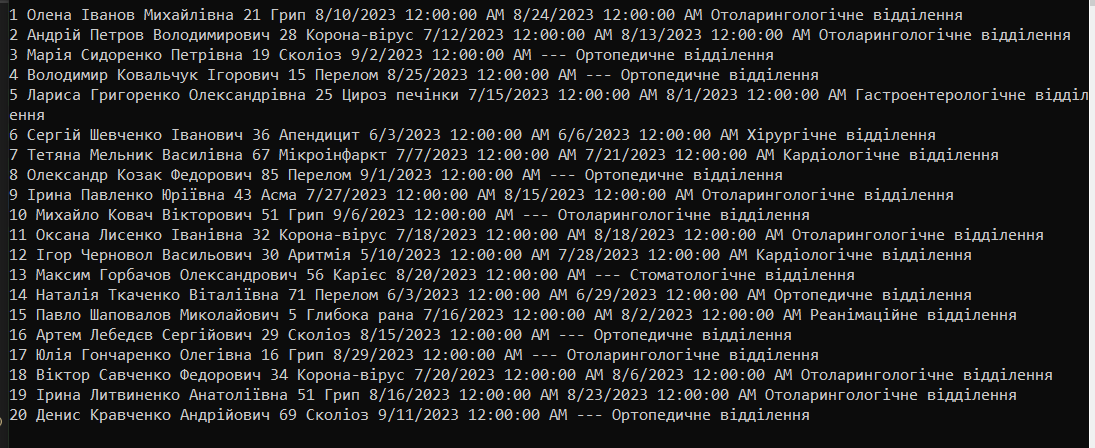
Залишаємо все як було і змінюємо лише к-сть.





Аналогічно робимо з пацієнтом та лікарем





В таблиці пацієнт змінюємо першого пацієнта, а в таблиці лікарі змінюємо останнього





Завдання 5.

Тепер розглянемо спосіб отримання/зміни інформації в БД за допомогою DataSet. Спочатку створимо об’єкти DbDataAdapter до кожної з таблиць за допомогою функції settingAdapters(), а також DataSet, в якому буде зберігатись інформація:

public static Dictionary<string, SqlDataAdapter> dataAdapters = new Dictionary<string, SqlDataAdapter>();

public static DataSet hSet = new DataSet("Hospital");

public static void settingAdapters()

{

SqlCommand cmd1 = con.CreateCommand();

cmd1.CommandType = CommandType.Text; cmd1.CommandText = "SELECT \* FROM Equipment";

dataAdapters.Add("Equipment", new SqlDataAdapter(cmd1));

SqlCommandBuilder bldr1 = new SqlCommandBuilder(dataAdapters["Equipment"]);

bldr1.GetUpdateCommand();

SqlCommand cmd2 = con.CreateCommand();

cmd2.CommandType = CommandType.Text; cmd2.CommandText = "SELECT \* FROM Medicine";

dataAdapters.Add("Medicine", new SqlDataAdapter(cmd2));

SqlCommandBuilder bldr2 = new SqlCommandBuilder(dataAdapters["Medicine"]);

bldr2.GetUpdateCommand();

SqlCommand cmd3 = con.CreateCommand();

cmd3.CommandType = CommandType.Text; cmd3.CommandText = "SELECT \* FROM Patient";

dataAdapters.Add("Patient", new SqlDataAdapter(cmd3));

SqlCommandBuilder bldr3 = new SqlCommandBuilder(dataAdapters["Patient"]);

bldr3.GetUpdateCommand();

SqlCommand cmd4 = con.CreateCommand();

cmd4.CommandType = CommandType.Text; cmd4.CommandText = "SELECT \* FROM Staff";

dataAdapters.Add("Staff", new SqlDataAdapter(cmd4));

SqlCommandBuilder bldr4 = new SqlCommandBuilder(dataAdapters["Staff"]);

bldr4.GetUpdateCommand();

SqlCommand cmd5 = con.CreateCommand();

cmd5.CommandType = CommandType.Text; cmd5.CommandText = "SELECT \* FROM Proced";

dataAdapters.Add("Proced", new SqlDataAdapter(cmd5));

SqlCommandBuilder bldr5 = new SqlCommandBuilder(dataAdapters["Proced"]);

bldr5.GetUpdateCommand();

}

Завдання 6&7.

Перетворимо раніше створені методи для оперування над базою даних через DataSet:

public static void AlterDataTablesDS(string tableName, string operation) // Функція редагування таблиць, оперується через DataSet'и

{

string tableNameAdapter = tableName.Remove(0, 4);

switch (tableName)

{

case "dbo.Equipment":

string name = ""; // Оголошуємо змінні, щоб уникнути помилки

string manuf = "";

string command = "";

int rowid = 0;

int quant = 0;

if (operation == "add") // Визначаємо операцію

{

Console.WriteLine("Введіть Назву, К-сть та виробника");

name = Console.ReadLine(); // Записуємо ввід користувача

quant = Convert.ToInt32(Console.ReadLine());

manuf = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter); // Заповнюємо нову таблицю датасету інформацією із адаптера цієї таблиці

var equipmentTable = hSet.Tables[tableNameAdapter];

equipmentTable.Rows.Add(0, name, quant, manuf); // Створюємо новий рядок у цій таблиці

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter); // Синхронізуємо інформацію датасету із БД

con.Close();

hSet.Clear(); // Очищаємо датасет для подальших операцій з ним

}

if (operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть Назву, К-сть та виробника");

name = Console.ReadLine();

string quantString = Console.ReadLine();

manuf = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var updRow = equipmentTable.Select($"IDeq={rowid}")[0]; // Витягуємо рядок, який потрібно модифікувати

if (!string.IsNullOrWhiteSpace(name)) // Записуємо зміни в рядок, якщо ввід не є пустим або пробілом

updRow["Name"] = name;

if (!string.IsNullOrWhiteSpace(quantString))

updRow["Quantity"] = Convert.ToInt32(quantString);

if (!string.IsNullOrWhiteSpace(manuf))

updRow["Manufacturer"] = manuf;

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "delete")

{

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var delRow = equipmentTable.Select($"IDeq={rowid}")[0]; // Витягуємо рядок, який потрібно видалити

delRow.Delete(); // Видаляємо рядок з датасету

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

break;

case "dbo.Medicine":

string crdate = "";

string expdate = "";

if (operation == "add")

{

Console.WriteLine("Введіть Назву, Виробника і Дати виготовлення/Термін придатності у форматі рік-місяць-число ");

name = Console.ReadLine();

manuf = Console.ReadLine();

crdate = Console.ReadLine();

expdate = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

equipmentTable.Rows.Add(0, name, manuf, crdate, expdate);

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть Назву, Виробника і Дати виготовлення/Термін придатності у форматі рік-місяць-число ");

name = Console.ReadLine();

manuf = Console.ReadLine();

crdate = Console.ReadLine();

expdate = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var updRow = equipmentTable.Select($"IDmed={rowid}")[0];

if (!string.IsNullOrWhiteSpace(name))

updRow["Name"] = name;

if (!string.IsNullOrWhiteSpace(manuf))

updRow["Producer"] = manuf;

if (!string.IsNullOrWhiteSpace(crdate))

updRow["Created\_Date"] = crdate;

if (!string.IsNullOrWhiteSpace(expdate))

updRow["Expiration\_Date"] = expdate;

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "delete")

{

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var delRow = equipmentTable.Select($"IDmed={rowid}")[0];

delRow.Delete();

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

break;

case "dbo.Patient":

string surname = "";

string middlename = "";

int age = 0;

string disease = "";

string ardate = "";

string disdate = "";

string ward = "";

if (operation == "add")

{

Console.WriteLine("Введіть ПІБ, Вік, Хворобу, Дати прибуття/виписки у форматі рік-місяць-число, а також відділення ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

disease = Console.ReadLine();

ardate = Console.ReadLine();

disdate = Console.ReadLine();

ward = Console.ReadLine();

string[] temp1 = ardate.Split("-");

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

if (disdate == "")

{

equipmentTable.Rows.Add(0, surname, name, middlename, age, disease,

ardate, DBNull.Value, ward);

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

break;

}

string[] temp2 = disdate.Split("-");

equipmentTable.Rows.Add(0, surname, name, middlename, age, disease, ardate, disdate, ward);

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ПІБ, Вік, Хворобу, Дати прибуття/виписки у форматі рік-місяць-число, а також відділення ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

string ageString = Console.ReadLine();

disease = Console.ReadLine();

ardate = Console.ReadLine();

disdate = Console.ReadLine();

ward = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var updRow = equipmentTable.Select($"IDpat={rowid}")[0];

if (!string.IsNullOrWhiteSpace(name))

updRow["First\_name"] = name;

if (!string.IsNullOrWhiteSpace(surname))

updRow["Last\_name"] = surname;

if (!string.IsNullOrWhiteSpace(middlename))

updRow["Middle\_name"] = middlename;

if (!string.IsNullOrWhiteSpace(ageString))

updRow["Age"] = ageString;

if (!string.IsNullOrWhiteSpace(disease))

updRow["Disease"] = disease;

if (!string.IsNullOrWhiteSpace(ardate))

updRow["Arrival\_date"] = ardate;

if (!string.IsNullOrWhiteSpace(disdate))

updRow["Discharge\_date"] = disdate;

if (!string.IsNullOrWhiteSpace(ward))

updRow["Ward"] = ward;

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "delete")

{

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var delRow = equipmentTable.Select($"IDpat={rowid}")[0];

delRow.Delete();

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

break;

case "dbo.Staff":

if (operation == "add")

{

Console.WriteLine("Введіть ПІБ, Вік і Посаду ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

age = Convert.ToInt32(Console.ReadLine());

string position = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

equipmentTable.Rows.Add(0, name, surname, middlename, age, position);

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ПІБ, Вік і Посаду ");

surname = Console.ReadLine();

name = Console.ReadLine();

middlename = Console.ReadLine();

string ageString = Console.ReadLine();

string position = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var updRow = equipmentTable.Select($"IDstaff={rowid}")[0];

if (!string.IsNullOrWhiteSpace(name))

updRow["Name"] = name;

if (!string.IsNullOrWhiteSpace(surname))

updRow["Last\_name"] = surname;

if (!string.IsNullOrWhiteSpace(middlename))

updRow["Middle\_name"] = middlename;

if (!string.IsNullOrWhiteSpace(ageString))

updRow["Age"] = ageString;

if (!string.IsNullOrWhiteSpace(position))

updRow["Position"] = position;

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "delete")

{

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var delRow = equipmentTable.Select($"IDstaff={rowid}")[0];

delRow.Delete();

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

break;

case "dbo.Proced":

int patid = 0;

int equipid = 0;

int staffid = 0;

int medid = 0;

int price = 0;

if (operation == "add")

{

Console.WriteLine("Введіть ID пацієнта, спорядження, персоналу та ліків, назву та ціну процедури: ");

patid = Convert.ToInt32(Console.ReadLine());

equipid = Convert.ToInt32(Console.ReadLine());

staffid = Convert.ToInt32(Console.ReadLine());

medid = Convert.ToInt32(Console.ReadLine());

name = Console.ReadLine();

price = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

equipmentTable.Rows.Add(0, patid, equipid, staffid, medid, name, price);

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "alter")

{

Console.WriteLine("Введіть id рядка, що буде змінений");

rowid = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введіть ID пацієнта, спорядження, персоналу та ліків, назву та ціну процедури: ");

string patidString = Console.ReadLine();

string equipidString = Console.ReadLine();

string staffidString = Console.ReadLine();

string medidString = Console.ReadLine();

name = Console.ReadLine();

string priceString = Console.ReadLine();

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var updRow = equipmentTable.Select($"IDproc={rowid}")[0];

if (!string.IsNullOrWhiteSpace(patidString))

updRow["Patient\_ID"] = patidString;

if (!string.IsNullOrWhiteSpace(equipidString))

updRow["Equipment\_ID"] = equipidString;

if (!string.IsNullOrWhiteSpace(staffidString))

updRow["Staff\_ID"] = staffidString;

if (!string.IsNullOrWhiteSpace(medidString))

updRow["Medicine\_ID"] = medidString;

if (!string.IsNullOrWhiteSpace(name))

updRow["Name"] = name;

if (!string.IsNullOrWhiteSpace(priceString))

updRow["Price"] = priceString;

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

if (operation == "delete")

{

Console.WriteLine("Введіть id рядка, що буде видалений");

rowid = Convert.ToInt32(Console.ReadLine());

con.Open();

dataAdapters[tableNameAdapter].Fill(hSet, tableNameAdapter);

var equipmentTable = hSet.Tables[tableNameAdapter];

var delRow = equipmentTable.Select($"IDproc={rowid}")[0];

delRow.Delete();

dataAdapters[tableNameAdapter].Update(hSet, tableNameAdapter);

con.Close();

hSet.Clear();

}

break;

default: Console.Clear(); Console.WriteLine("Такої таблиці не існує!"); break;

}

}

public static void Menu\_operationsDS(string tableName) // Меню операцій, оперується через DataSet'и

{

Console.WriteLine("Виберіть наступну операцію:\n1)Додати дані до таблиці\n2)Видалити дані з таблиці\n3)Редагувати стрічку таблиці\n4)Повернутись");

string a = Console.ReadLine();

switch (a)

{

case "1":

AlterDataTablesDS(tableName, "add");

Console.WriteLine("Дані додані успішно");

break;

case "2":

AlterDataTablesDS(tableName, "delete");

Console.WriteLine("Дані видалені успішно");

break;

case "3":

AlterDataTablesDS(tableName, "alter");

Console.WriteLine("Дані змінені успішно");

break;

case "4":

Console.Clear();

Menu\_pickTablesDS();

break;

default:

Console.Clear();

Console.WriteLine("Немає такої операції!");

break;

}

}

public static void Menu\_pickTablesDS() // Меню вибору таблиць, оперується через DataSet'и

{

Console.Clear();

Console.WriteLine("Введіть назву таблиці\n1)dbo.Equipment \n2)dbo.Medicine \n3)dbo.Staff \n4)dbo.Patient \n5)dbo.Proced\n6)Повернутись");

string tableName = Console.ReadLine();

Console.Clear();

switch (tableName)

{

case "1":

tableName = "dbo.Equipment";

Display("Equipment");

Menu\_operationsDS(tableName);

break;

case "2":

tableName = "dbo.Medicine";

Display("Medicine");

Menu\_operationsDS(tableName);

break;

case "3":

tableName = "dbo.Staff";

Display("Staff");

Menu\_operationsDS(tableName);

break;

case "4":

tableName = "dbo.Patient";

Display("Patient");

Menu\_operationsDS(tableName);

break;

case "5":

tableName = "dbo.Proced";

Display("Proced");

Menu\_operationsDS(tableName);

break;

case "6":

Console.Clear();

Menu\_main();

break;

default:

Console.Clear();

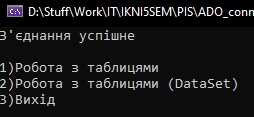
Console.WriteLine("No such choice");

break;

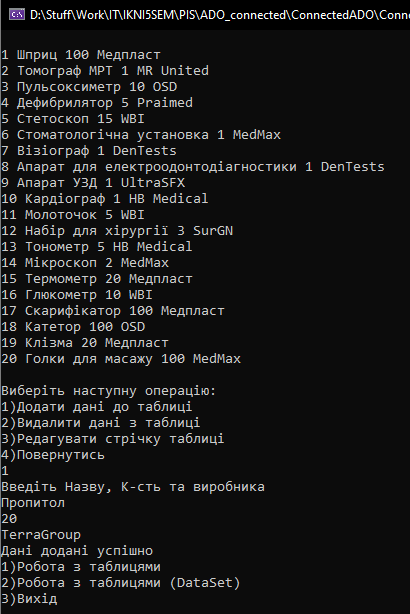
}

}

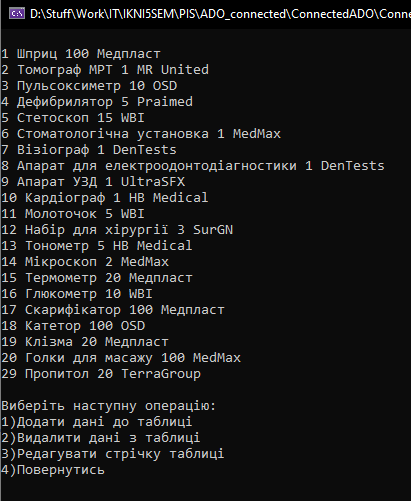
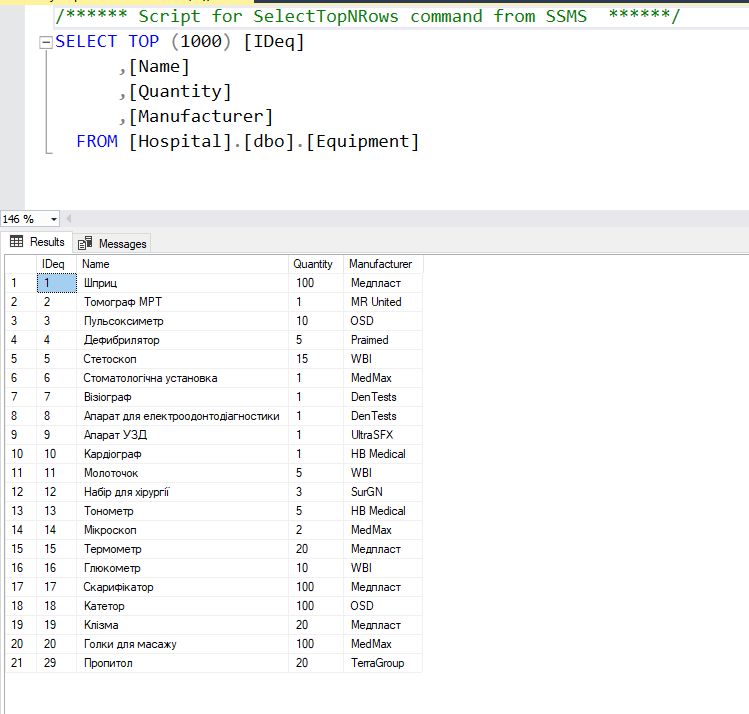
Випробуємо нашу програму тепер:



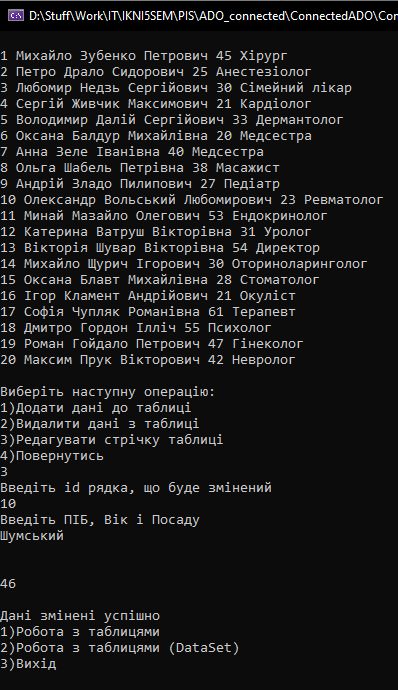
Обираємо другу опцію. Добавимо в таблицю Equipment новий елемент:



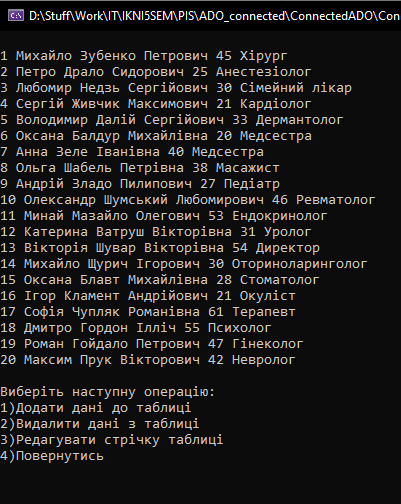
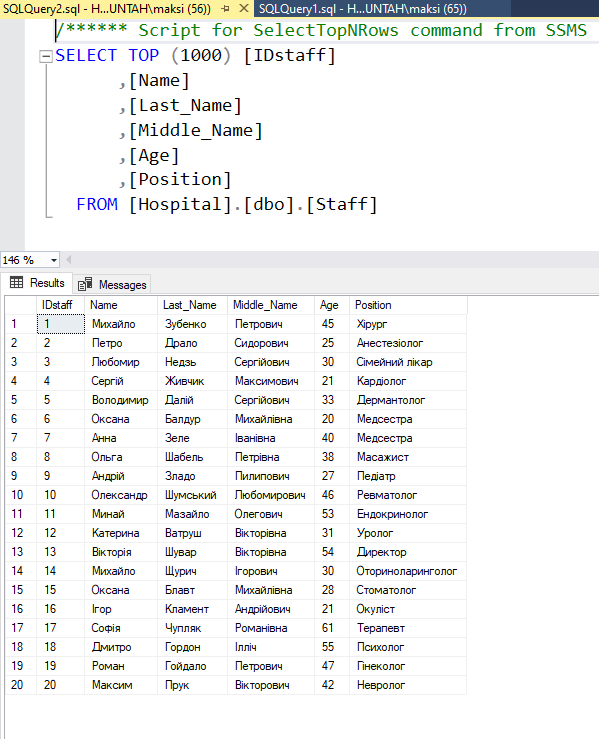
Перевіримо правильність виконання операції в SSMS та програмі:



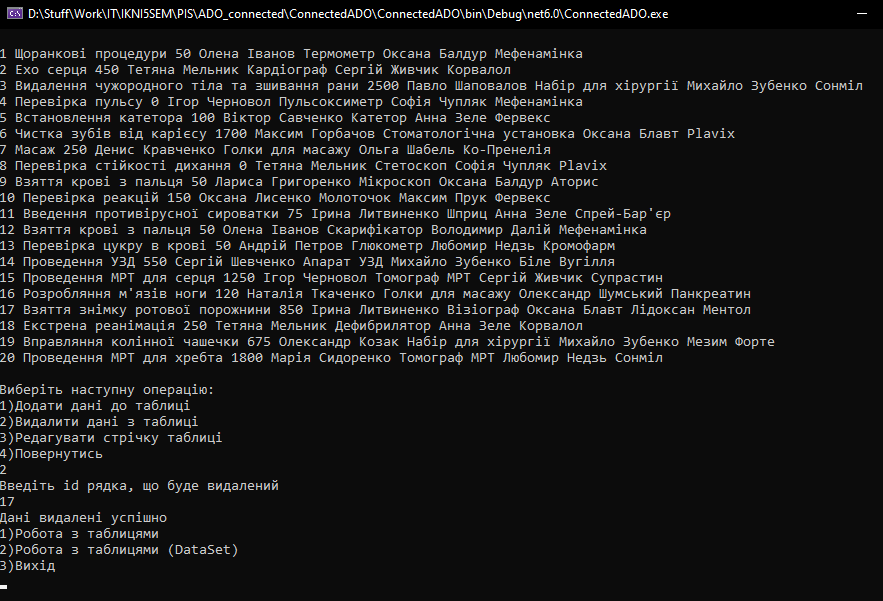
Тепер змінимо дані в іншій таблиці. Перейдемо до таблиці Staff та змінимо інформацію в рядку 10:



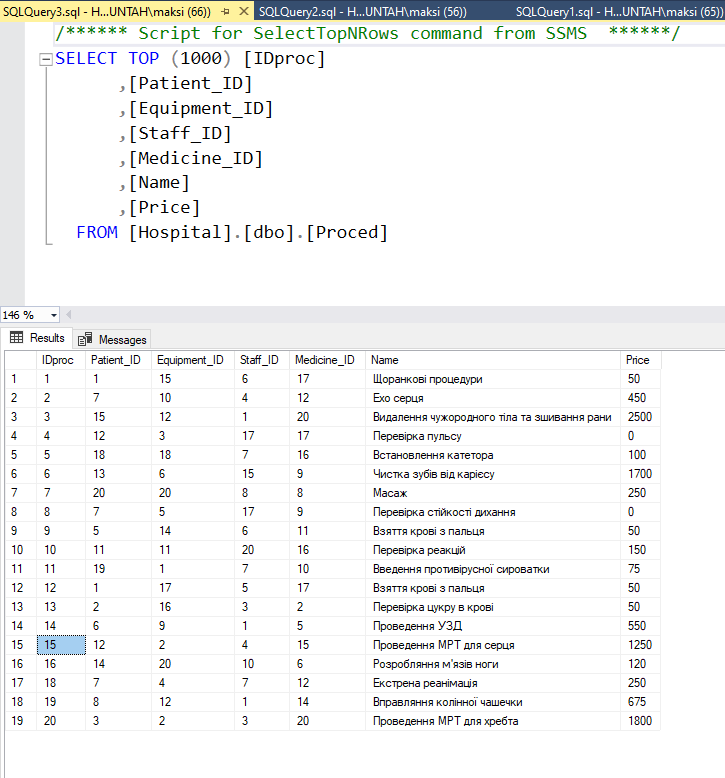
Перевіримо коректність виконання операції:

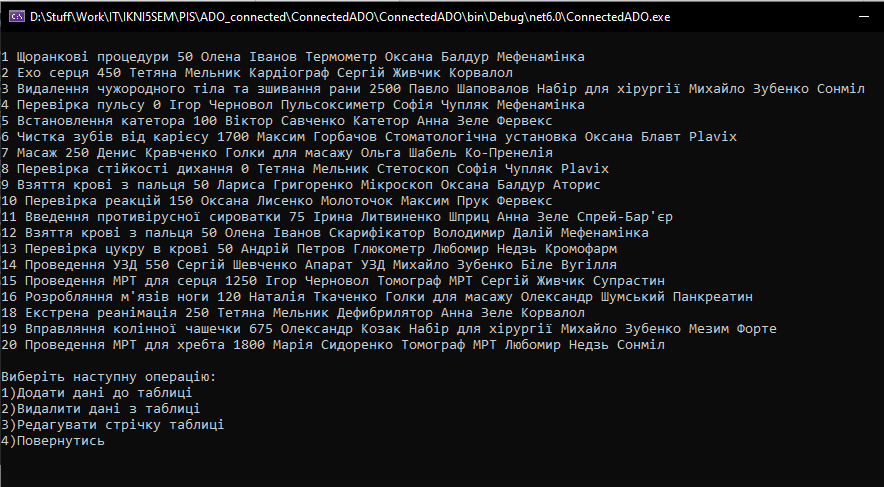


Тепер видалимо якийсь рядок.



Перевіримо коректність виконання операції:





Як бачимо всі операції були виконані успішно.

**Висновок:** На цій лабораторній роботі ми ознайомились зі структурою класів ADO.NET для роботи з БД, здобули практичні навички роботи з класами ADO.NET, що забезпечують доступ і роботу з БД, зокрема DbConnection, DBCommand, DbDataAdapter та інші. Написали програму, що з’єднюється з нашою БД Hospital та дозволяє оперувати над нею через прості запити та за допомогою DataSet’у.