

# Застосування реляції даних на стороні клієнту з використанням технології ADO .NET на прикладах AdoRelation1 та AdoRelation2

## *Приклад AdoRelation1*

Працюючи з технологією ADO.NET об'єкт **DataSet** є сукупність таблиць моделі бази даних на боці клієнта. У цій моделі можна здійснювати реляційні зв'язки між таблицями за допомогою властивості **Relations** об'єкта **DataSet**, який повертає **DataRelation** об'єкт. Оскільки в елементі керування **DataGridView** можна розмістити лише одну таблицю, для демонстрації зв'язку кількох таблиць розмістимо два об'єкти класу **DataGridView**, відповідно **dataGridView1** та **dataGridView2**.

Як приклад реляційних зв'язків наведемо зв'язок «один-до-багатьох» на прикладі таблиць бази даних «nwind.mdb» **Categories** і **Products**. Так, одній категорії товарів відповідає багато різних продуктів.

Наведемо повний текст програмного коду **ADORelation1**:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Ado1
{
    public partial class Form1 : Form
    {
        //Об'єкт Connection
        OleDbConnection conn;
        DataSet ds;

        public Form1()
        {
            InitializeComponent();

            //Создание объекта Connection и Recordset
            try
            {
                string source = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=NWIND.MDB;Mode=ReadWrite";

                string select = "SELECT CategoryID,CategoryName FROM Categories";
                conn = new OleDbConnection(source);
                conn.Open();

                ds = new DataSet();
                OleDbDataAdapter da1 = new OleDbDataAdapter(select, conn);

                da1.Fill(ds, "Категория");
            }
            catch { }
        }
    }
}
```

```

da1.TableMappings.Add("Categories", "Категория");

select = "SELECT CategoryID,ProductName FROM Products";
OleDbDataAdapter da2 = new OleDbDataAdapter(select, conn);

da2.Fill(ds, "Продукты");
da1.TableMappings.Add("Products", "Продукты");

ds.Relations.Add("Продукты категории", ds.Tables[0].Columns[0],
                ds.Tables[1].Columns[0]);
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "Категория";

dataGridView2.DataSource = ds;
dataGridView2.DataMember = "Категория.Продукты категории";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
}
}
}

```

У цьому прикладі в одному об'єкті **DataSet** створюємо дві таблиці – об'єкти **DataTable**. За допомогою властивості **TableMappings** об'єкта **DataAdapter** створюємо псевдонім для таблиць. Для таблиці "Categories" виберемо псевдонім "Категорія", для таблиці "Products" - "Продукти". Далі при побудові запитів можна скористатися даними псевдонімами.

Далі створимо реляційні зв'язки між таблицями, використовуючи властивість **Relations** об'єкта **DataSet**. Зв'язок «один-до-багатьох» здійснюється за ключовим полем «CategoryID»:

```

.....
ds.Relations.Add("Продукты категории", ds.Tables[0].Columns[0],
                ds.Tables[1].Columns[0]);
.....

```

Ім'я реляційного зв'язку - "Продукти категорії".

На наступному етапі прив'яжемо таблицю "Категорія" з первинним ключем до елемента **dataGridView1**.

```

.....
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "Категория";
.....

```

А таблицю із зовнішнім ключем – до елемента **dataGridView2**. При цьому зазначимо реляційний зв'язок «Категорія. Продукти категорії».

```

.....
dataGridView2.DataSource = ds;

```

```
dataGridView2.DataMember = "Категория.Продукты категории";
```

.....

Такий запис означає, що при переміщенні на певний запис у головній таблиці «Категорія» – у другій таблиці ми бачитимемо лише ті товари, які відповідають обраній категорії.

На рис. 1. наведемо результат виконання програми **ADORelation1**.

Так, у другому елементі **dataGridView2** ми бачимо лише продукти обраної 6-ї категорії (поле **CategoryID**).

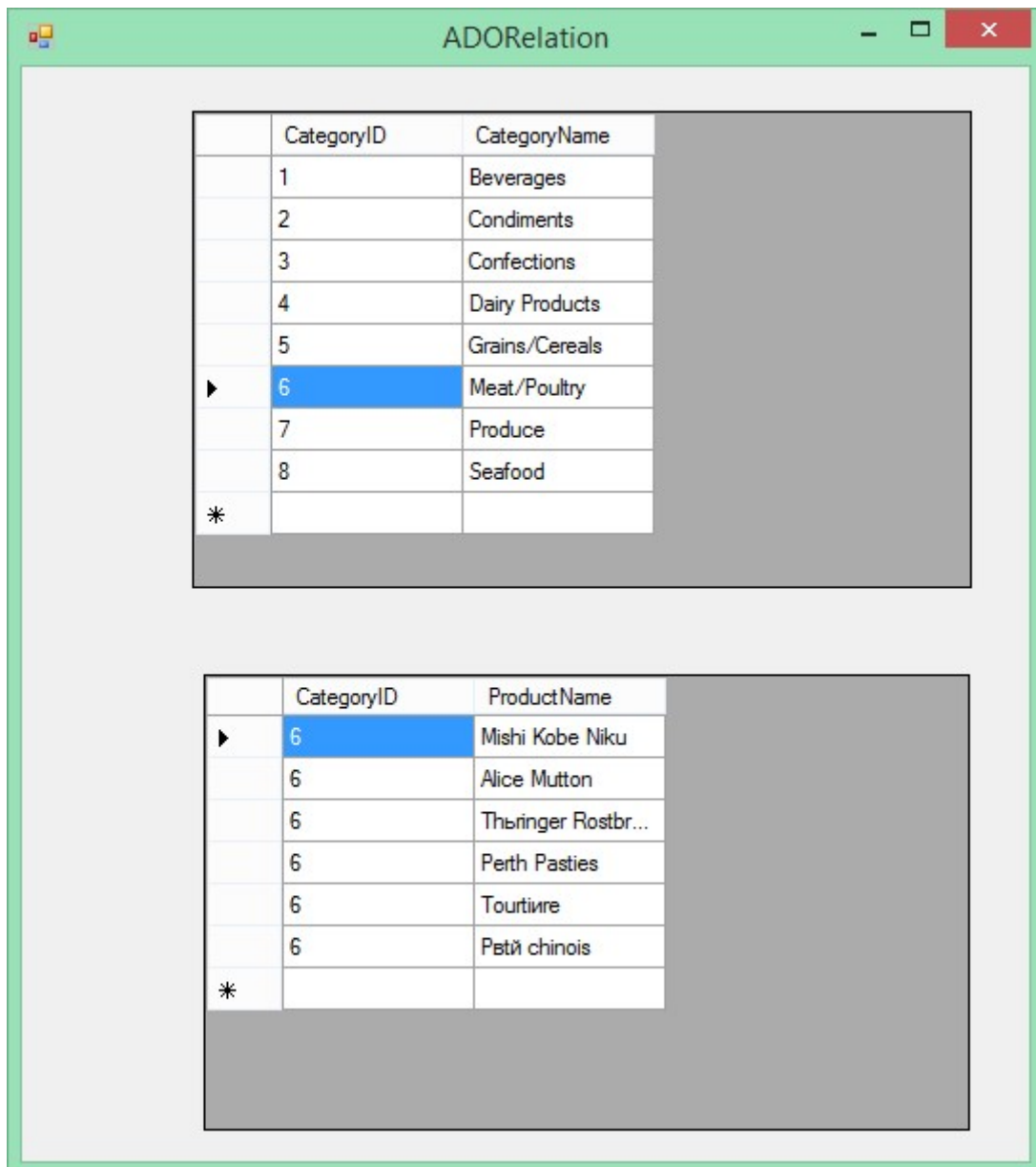


Рис. 1. Результат роботи програми ADORelation1

## *Приклад AdoRelation2*

У прикладі **ADORelation2** також здійснюється реляційний зв'язок «один-до-багатьох». Відмінністю є застосування старішого елемента керування відображення таблиць **DataGrid**. Його перевагою є відображення підлеглих таблиць при зв'язку одним так, як це реалізовано в **MS Access**.

Наведемо повний текст програмного коду **ADORelation2**:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Ado1
{
    public partial class Form1 : Form
    {
        //Объект Connection
        OleDbConnection conn;
        DataSet ds;

        public Form1()
        {
            InitializeComponent();

            //Создание объекта Connection и Recordset
            try
            {
                string source = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=NWIND.MDB;Mode=ReadWrite";

                string select = "SELECT CategoryID,CategoryName FROM Categories";
                conn = new OleDbConnection(source);
                conn.Open();
                ds = new DataSet();

                OleDbDataAdapter da1 = new OleDbDataAdapter(select, conn);
                da1.Fill(ds,"Категория");
                da1.TableMappings.Add("Categories", "Категория");

                select = "SELECT CategoryID,ProductName FROM Products";
                OleDbDataAdapter da2 = new OleDbDataAdapter(select, conn);

                da2.Fill(ds,"Продукты");
                da1.TableMappings.Add("Products", "Продукты");
                ds.Relations.Add("Продукты категории", ds.Tables[0].Columns[0],
                    ds.Tables[1].Columns[0]);
                dataGridView1.DataSource = ds.Tables[0];
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}
```

На рис. 2. наведемо результат виконання програми **ADORelation2**.

На відміну від прикладу **AdoRelation1** дані батьківської та підлеглої таблиці можна побачити в елементі керування **DataGrid**. Для перегляду підлеглих даних необхідно натиснути "+".

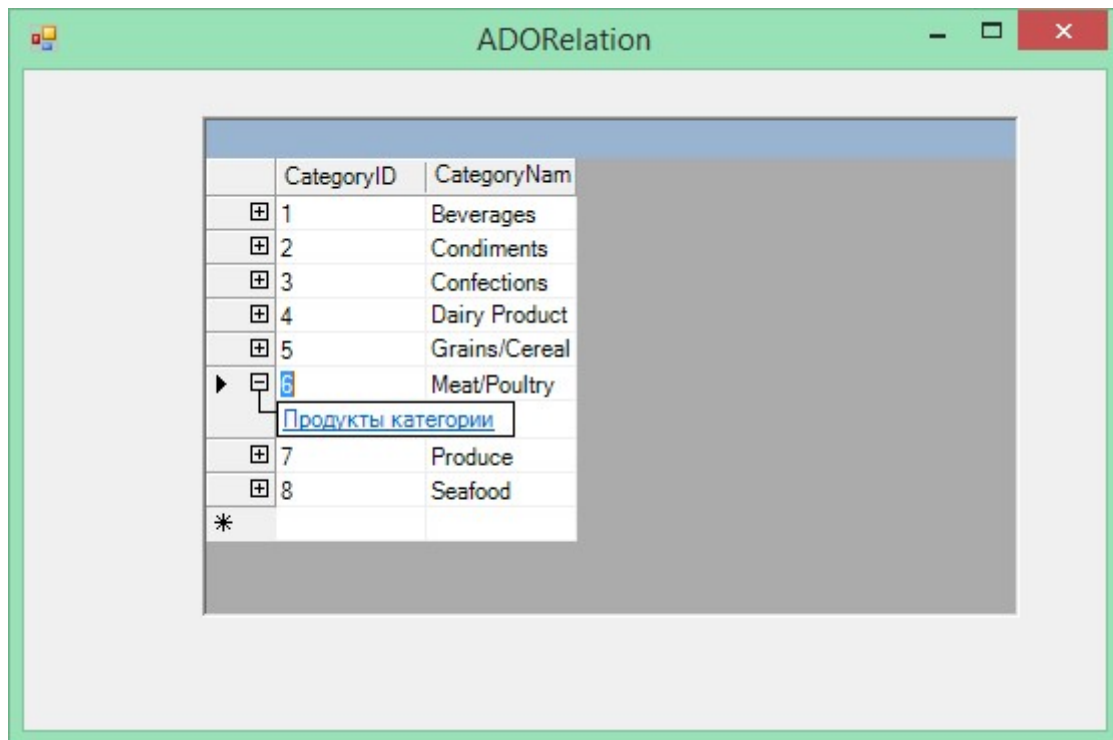


Рис. 2. Результат роботи програми ADORelation2