

DataAdapter

Čas 3.

Čemu služi?

- ▶ Obezbeđuje lak način za rad sa podacima.
- ▶ Objedinjuje
 - ▶ Insert
 - ▶ Delete
 - ▶ Update
 - ▶ Select
- ▶ Ažuriranje podataka u bazi izvodi korisniku na intuitivan način.
- ▶ Sadrži:
 - ▶ Konekciju
 - ▶ Komande



Kreiranje

- ▶ `string sql = "select * from Customers";`
- ▶ `SqlConnection sqlconn = new ....`
- ▶ `SqlDataAdapter da = new SqlDataAdapter(...);`
- ▶ `//Kuda sa podacima?`
- ▶ `//Kreiranje DataSet objekta za prihvatanje podataka`
- ▶ `DataSet ds = new DataSet();`
- ▶ `da.Fill(ds);`
- ▶ `//kako prikazati podatke iz ds?`

Deo koji definiše select komandu adaptera

Svojstva - 1

- ▶ **Komande**
- ▶ DeleteCommand
- ▶ InsertCommand
- ▶ SelectCommand
- ▶ UpdateCommand

Svojstva - 2

- ▶ **TableMappings** – kolekcija koja obezbeđuje relaciju između kolona iz objekta DataSet i izvora podataka. (Kako Fill ume da prebaci podatke iz baze u DS)
- ▶ **AcceptChangesDuringFill** – određuje da li se na objektu DataRow poziva AcceptChanges kada se doda u objekat DataTable
- ▶ **MissingMappingAction** – definiše akciju koja će se dogoditi kada se ne mogu upariti podaci sa nekom postojećom kolonom ili tabelom
- ▶ **MissingSchemaAction** - slično ali za šemu

Svojstva - 3

- ▶ U slučaju greške moguće akcije za `MissingMappingAction` su:
 - ▶ **Error**
 - ▶ **Ignore**
 - ▶ **Passthrough** – kolona ili tabela koja se ne pronađe dodaje se u `DataSet` korišćenjem njenog imena u izvoru podataka

Svojstva - 4

- ▶ Svojstva za **MissingSchemaAction** su:
 - ▶ **Add** dodaje potrebne kolone
 - ▶ **AddWithKey** dodaje potrebne kolone i ograničenja primarnog ključa
 - ▶ **Error** Ispaljuje izuzetak
 - ▶ **Ignore** Ignoriše dodate kolone

Svojstva - 5

TableMappings:

1. Kolekcija
2. Relacije između izvora podataka sa jedne strane i skladišta podataka (DataSet-a) sa druge strane.
3. Vezuje se ime kolone u tabeli u bazi sa imenom kolone u tabeli u DataSet-u

Metode

Fill metoda

- ▶ Popunjava DataSet podacima sa prethodim brisanjem postojećih ili ne.
 - ▶ To se podešava u određenim svojstvima – pronađite kako

Update

- ▶ Promene koje su učinjene na podacima se prebacuju u bazu.
- ▶ Kako?
 - ▶ Pogledajte kako glasi sql upit komande Update.

Događaji

- ▶ Događaji greške
- ▶ OnRowUpdating
 - ▶ Odmah posle trenutka kada metod Update postavi vrednosti parametara komande koja treba da se izvrši ali pre samog izvršavanja.
- ▶ Argument uz ovaj događaj
 - ▶ OleDbRowUpdatingEventArgs
 - ▶ SqlRowUpdatingEventArgs
 - ▶ Svojstva ovog objekta su:
 - Command - komanda za podatke koja treba da se izvrši
 - Errors – greške koje .NET generiše
 - Row – objekat DataReader koji treba da se ažurira
 - StatementType – *Select, Insert, Update, Delete*
 - TableMapping – Objekat DataTableMapping koji se koristi za ažuriranje

OnRowUpdated

- ▶ Kada metod Update izvrši odgovarajuću komandu na izvoru podataka
 - ▶ Svojstva argumenta koji ide uz ovaj događaj pogledaćemo na primeru.

DataSet

Metode

- ▶ *Clear* – briše sve tabele
- ▶ *Clone* – kopira strukturu DataSet-a
- ▶ *Copy* – kopira i strukturu i podatke
- ▶ *HasChanges* – da li u objektu DataSet postoje izmene koje čekaju.

Filtriranje i uređivanje u okviru DataSeta

- ▶ 1. DataSet objekat napuniti pomoću DA podacima iz tabela *Customers* odnosno *Products*.
- ▶ `string SQL = "select * from customers; select * from products where UnitPrice < 10";`
- ▶ `da.SelectCommand = new SqlCommand(SQL, conn);`
- ▶ `da.Fill(ds);`
- ▶ `da.Fill(ds, "Customers");`
- ▶ `// proveriti kreirane tabele u ds i njihova imena u nekoliko varijanti popunjavanja`

Dodavanje reda

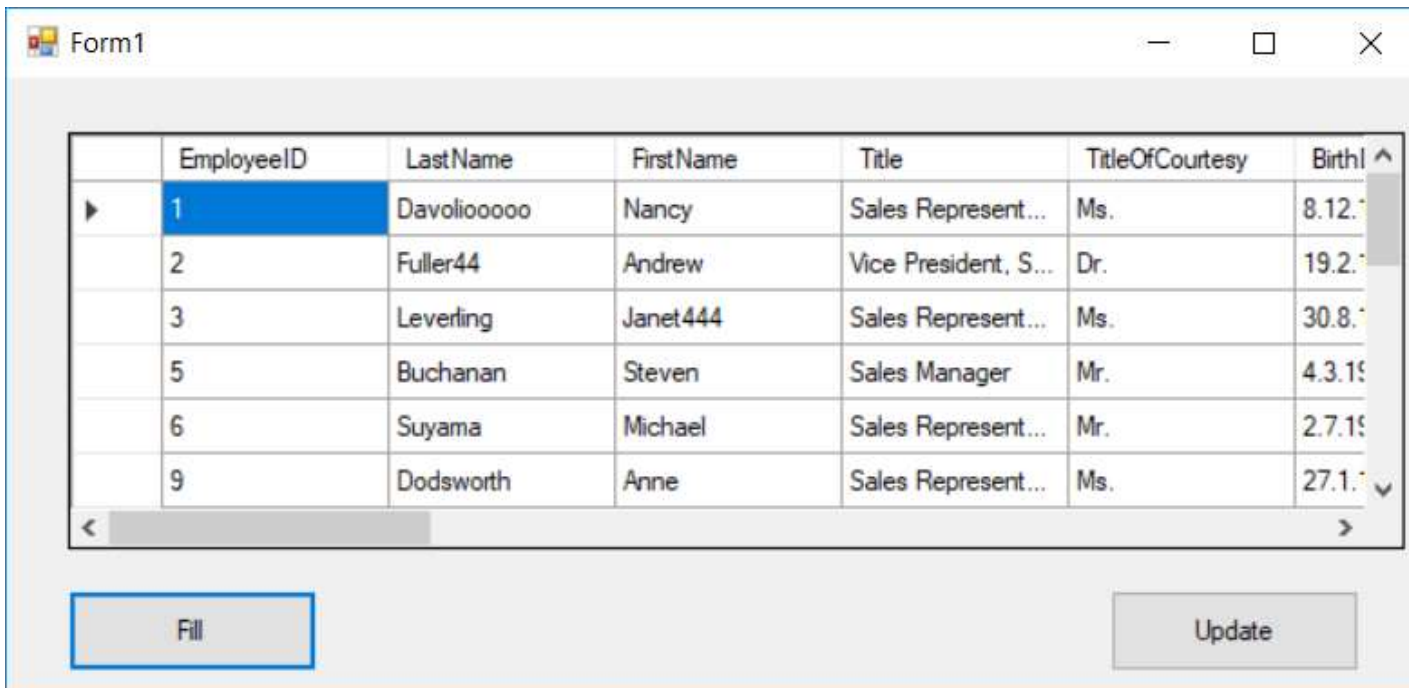
- ▶ `DataRow newrow= dt.NewRow();`
- ▶ `newrow[""] = ;`
- ▶
- ▶ `dt.Rows.Add(newrow);`

Filtriranje, sortiranje redova u DataSet-u

- ▶ `DataTableCollection dtc = ds.Tables;`
- ▶ `dtc["Customers"].Select("xxxx","yyyy");`
- ▶ `pr: xxxx : Country = 'Germany'`
- ▶ `pr: yyyy: CompanyName ASC`

Zadatak

- ▶ Napraviti aplikaciju za ažuriranje tabele Employees, kao na slici.
 - ▶ Fill – popunjavanje kontrolu podacima
 - ▶ Update – snimanje promena u tabeli

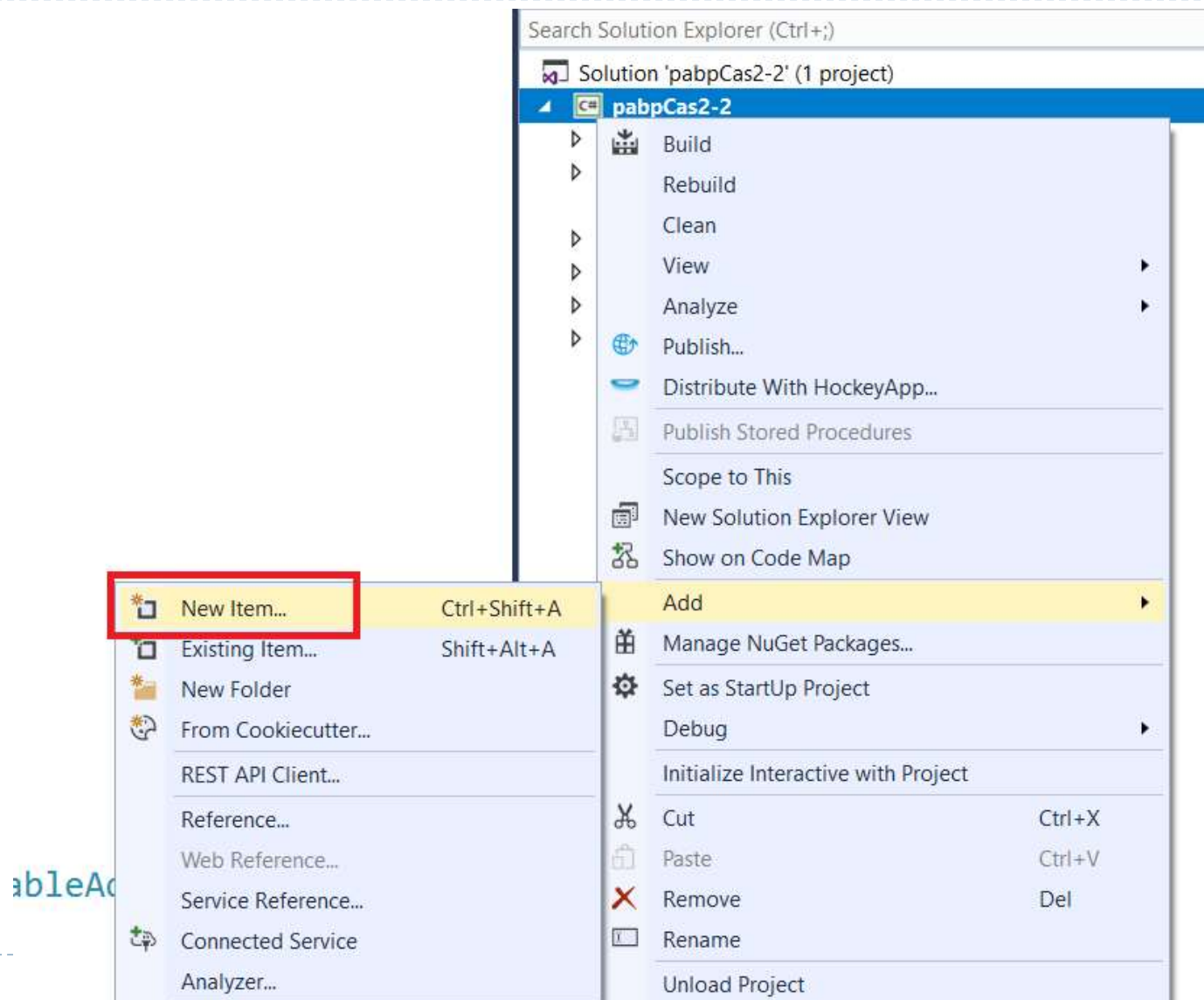


The screenshot shows a Windows application window titled "Form1". Inside the window is a table with 7 columns: EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, and BirthDate. The table contains 6 rows of data. The first row is highlighted in blue. Below the table are two buttons: "Fill" and "Update".

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate
▶	1	Davoliooooo	Nancy	Sales Represent...	Ms.	8.12.19...
	2	Fuller44	Andrew	Vice President, S...	Dr.	19.2.19...
	3	Leverling	Janet444	Sales Represent...	Ms.	30.8.19...
	5	Buchanan	Steven	Sales Manager	Mr.	4.3.19...
	6	Suyama	Michael	Sales Represent...	Mr.	2.7.19...
	9	Dodsworth	Anne	Sales Represent...	Ms.	27.1.19...

Fill Update

► Dodavanje imenovanog DataSet objekta.



Add New Item - pabpCas2-2



Installed

Sort by: Default

Search (Ctrl+E)

Visual C# Items

Code

Data

General

Web

Windows Forms

WPF

ASP.NET Core

SQL Server

Storm Items

Graphics

Online



ADO.NET Entity Data Model

Visual C# Items



DataSet

Visual C# Items



EF 5.x DbContext Generator

Visual C# Items



EF 6.x DbContext Generator

Visual C# Items



Service-based Database

Visual C# Items



XML File

Visual C# Items



XML Schema

Visual C# Items



XSLT File

Visual C# Items

Type: Visual C# Items

A DataSet for using data in your application

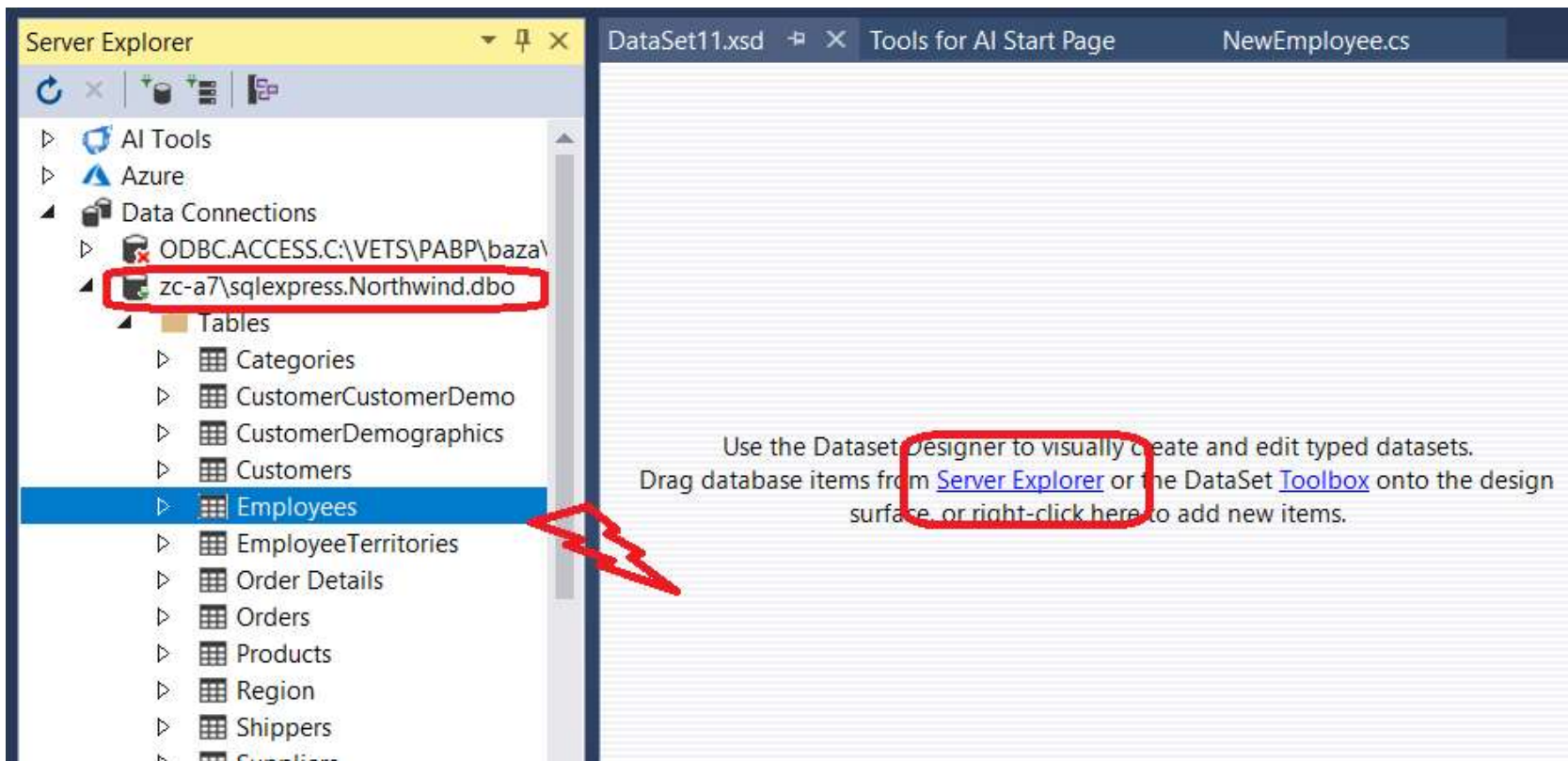
Name:

DataSet1.xsd

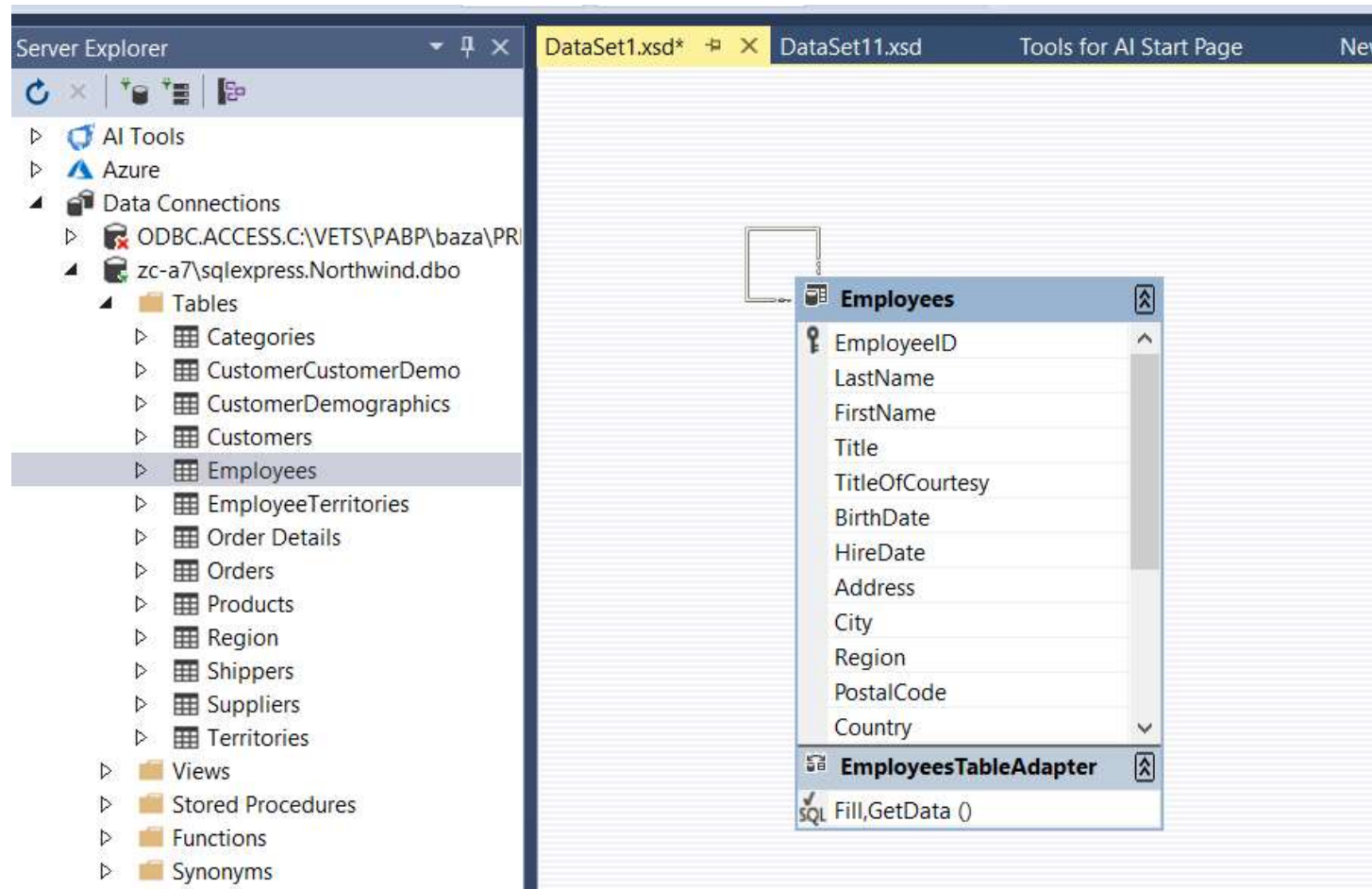
Add

Cancel

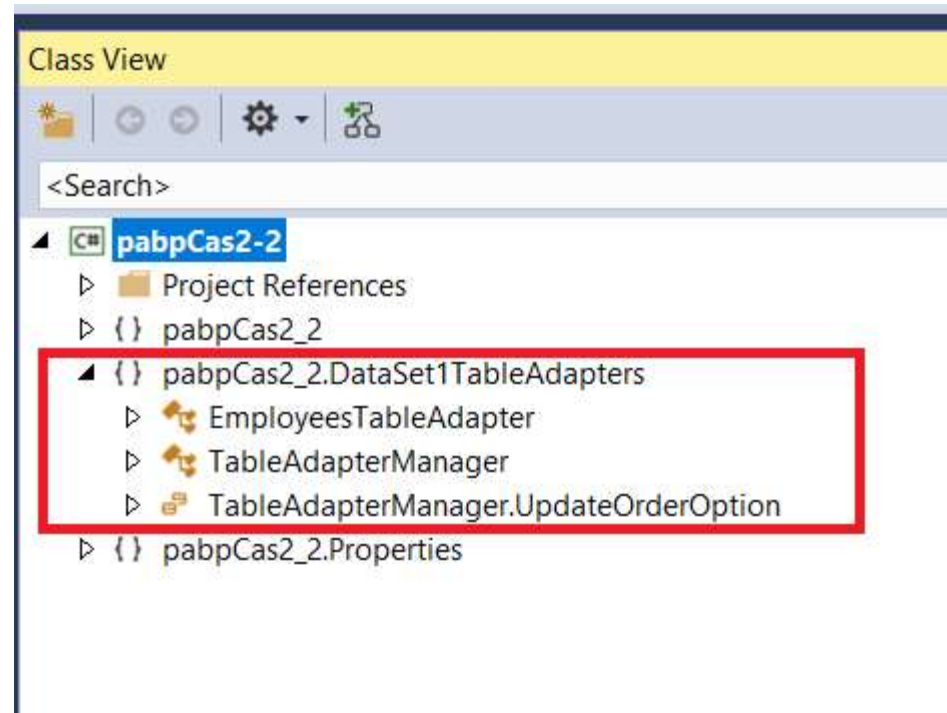
- ▶ Prevlačenjem tabela iz prozora Server Explorer formirate tipizirane tabele u DataSet objektu



► Konačan izgled kreiranog DataSetI objekta



-
- ▶ Pogled na kreirane klase tipiziranog DataSetI objekta.



► Organizacija koda

```
public partial class Form1 : Form
{
    DataSet1 ds;
    pabpCas2_2.DataSet1TableAdapters.EmployeesTableAdapter da;
    public Form1()
    {
        InitializeComponent();

        ds = new DataSet1();
        da = new DataSet1TableAdapters.EmployeesTableAdapter();
    }

    private void btnFill_Click(object sender, EventArgs e)
    {
        da.Fill(ds.Employees);
    }

    private void btnUpdate_Click(object sender, EventArgs e)
    {
        int cnt = da.Update(ds.Employees);
        MessageBox.Show("cnt=" + cnt);
    }
}
```



- ▶ Novi zadatak: Izdvajanje redova iz DataSet objekta

Form1

First Name

Filter result:

	EmployeeID	LastName	FirstName	
▶	1	Davoliooooo	Nancy	S
	2	Fuller44	Andrew	V
	3	Leverling	Janet444	S
	5	Buchanan	Steven	S
	6	Suyama	Michael	S
	9	Dodsworth	Anne	S
	10	Peric	Pera	
	11	bbb1	bbb1	a
	12	a	b233	a
	15	hhh44	aaa1	a

< >

- ▶ Dodavanje tabela za rezultat filtriranja I povezivanje iste sa drugom kontrolom.
-

```
public partial class Form1 : Form
{
    . . .
    DataSet1.EmployeesDataTable dt;
    . . .
    public Form1()
    {
        . . .
        dt = new DataSet1.EmployeesDataTable();
        dataGridView2.DataSource = dt;
        . . .
    }
}
```



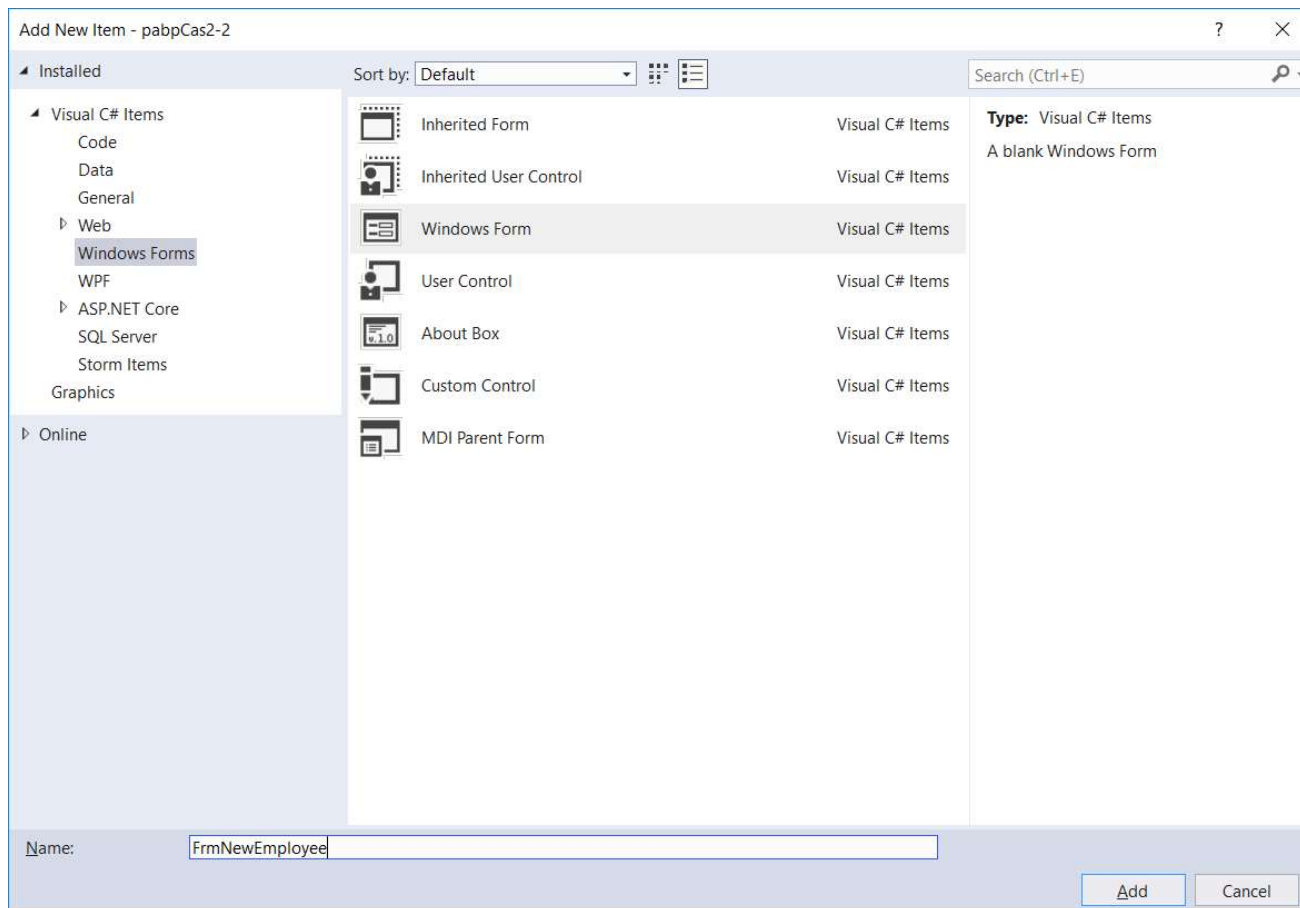
► Izdvajanje filtriranih redova:

```
private void btnFilter_Click(object sender, EventArgs e)
{
    DataSet1.EmployeesRow[] rows = (DataSet1.EmployeesRow[])
ds.Employees.Select("FirstName like '%" + txtFirstName.Text +
"%')");
    DataRow[] rows2 = ds.Employees.Select("FirstName like '%"
+ txtFirstName.Text + "%')");

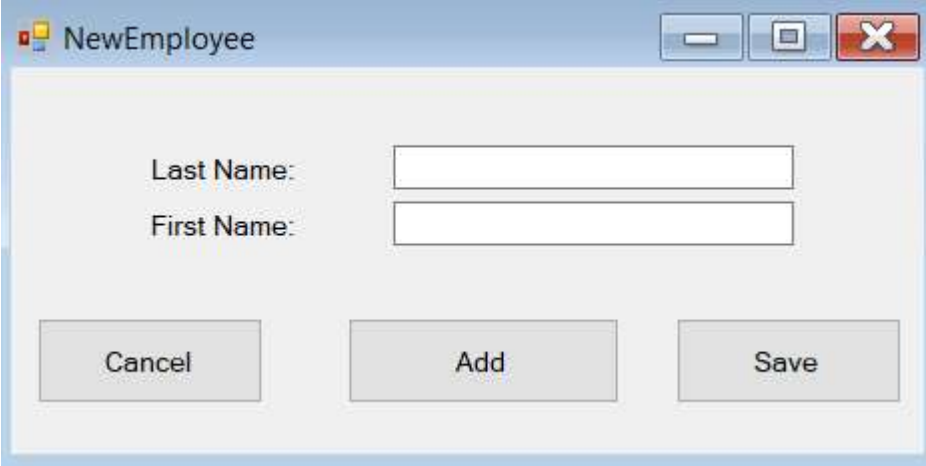
    foreach (DataSet1.EmployeesRow row in rows)
    {
        DataSet1.EmployeesRow newrow = dt.NewEmployeesRow();
        newrow.FirstName = row.FirstName;
        newrow.LastName = row.LastName;
        newrow.EmployeeID = row.EmployeeID;
        dt.AddEmployeesRow(newrow);
    }
}
```



- ▶ Dodavanje posebne forme za unos novog reda podataka:
 - ▶ Desni klik na Projekat u prozoru Solution Explorer
 - ▶ Add -> New Item



- ▶ Nova forma treba da:
 - ▶ Ima izgled kao na slici
- ▶ Obezbedi unos vise redova pritiskom na dugme Add bez snimanja u bazu
- ▶ Pritiskom na dugme Save snimi sve dodate redove



The image shows a screenshot of a Windows application window titled "NewEmployee". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first field is labeled "Last Name:" and the second field is labeled "First Name:". Below these fields, there are three buttons: "Cancel", "Add", and "Save". The "Add" button is positioned between the "Cancel" and "Save" buttons.

- ▶ Na novoj formi korišćićemo isti DataSet objekat kao i na prvoj formi aplikacije.

- ▶ Zbog toga ćemo preneti referencu iz prve forme preko novog konstruktora nove forme

```
public partial class FrmNewEmployee : Form
{
    DataSet1 ds;
    pabpCas2_2.DataSet1TableAdapters.EmployeesTableAdapter da;
    public FrmNewEmployee()
    {
        InitializeComponent();
    }
    public FrmNewEmployee(DataSet1 _ds,
                           pabpCas2_2.DataSet1TableAdapters.EmployeesTableAdapter _da)
    {
        ds = _ds;
        da = _da;

        InitializeComponent();
    }
}
```



► Dodavanje novog reda

```
private void btnAdd_Click(object sender, EventArgs e)
{
    DataSet1.EmployeesRow newrow = ds.Employees.NewEmployeesRow();
    newrow.FirstName = firstNameTextBox.Text;
    newrow.LastName = lastNameTextBox.Text;
    ds.Employees.AddEmployeesRow(newrow);
}
```

► Snimanje promena

```
private void btnSave_Click(object sender, EventArgs e)
{
    da.Update(ds.Employees);
    this.Close();
}
```

► Zatvaranje forme bez snimanja

```
private void btnCancel_Click(object sender, EventArgs e)
{
    // Ostao neresen problem poništavanja promena u DataSet-u
    this.Close();
}
```

