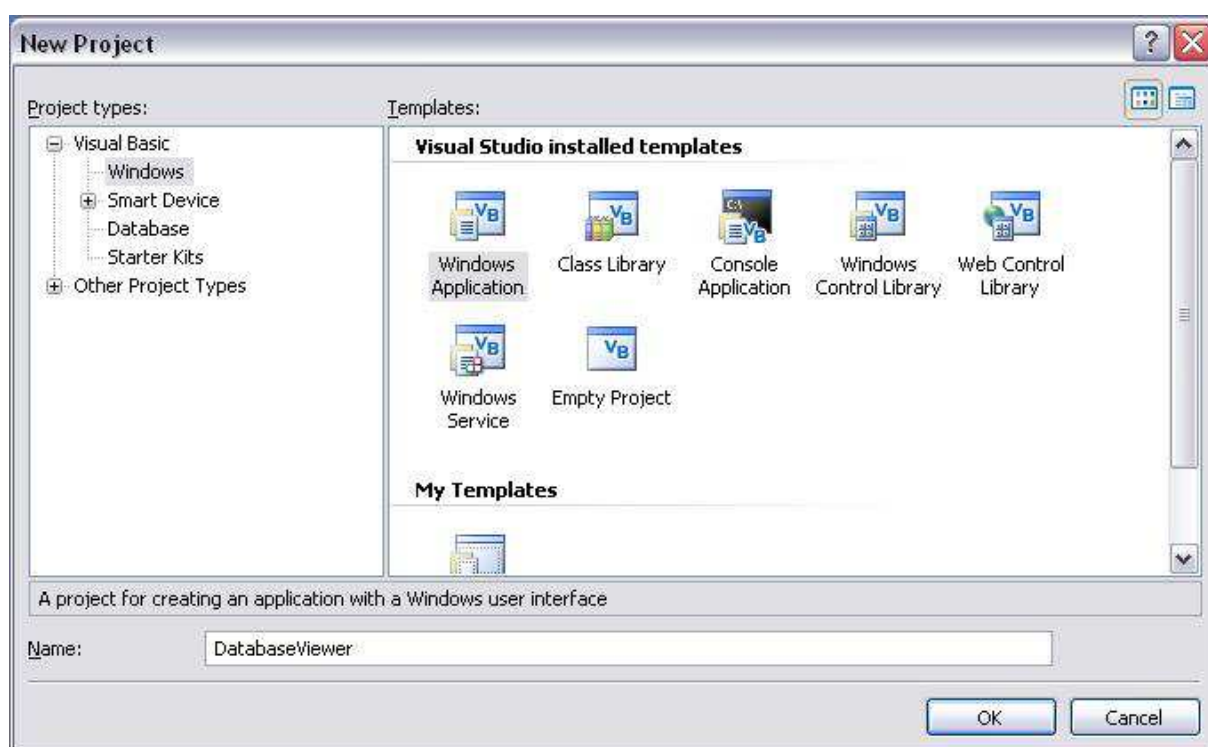


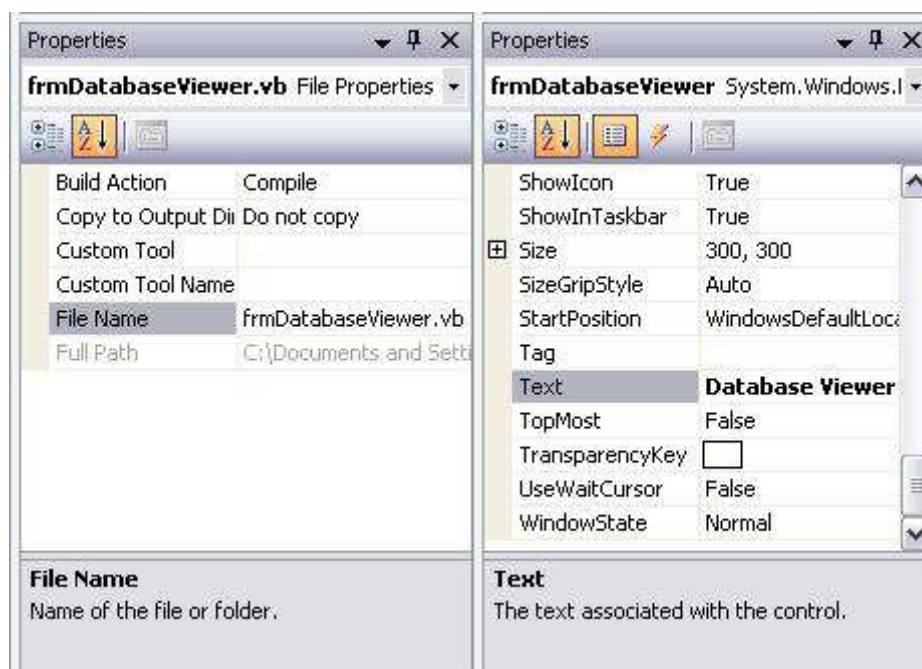
# Przykład pokazujący jak wykorzystując tryb design wygenerować aplikację współpracującą z konkretną bazą danych

## 1. Tworzenie Przeglądarki bazy danej

1. Stwórz nowy projekt typu Windows Application.



2. Ustal nowe właściwości projektu zmieniając nazwę pliku typu formularz (**File Name**) oraz właściwość typu Name i Text dla samego formularza.

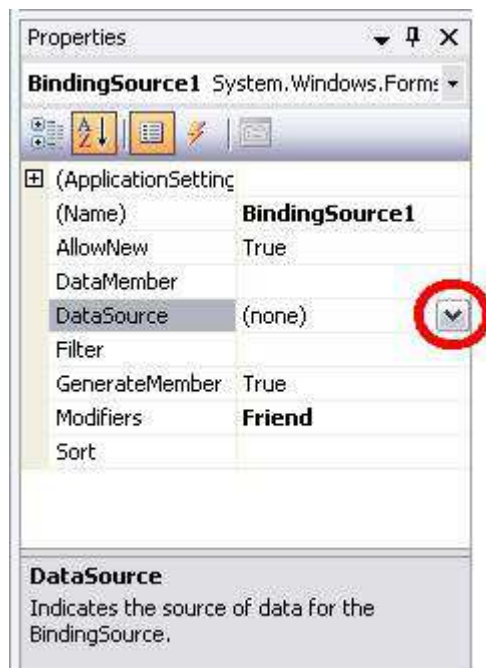


### 3. Stworzenie połączenia z bazą danych

Dodaj do projektu kontrolkę **BindingSource** z sekcji **Data** w **Control Toolbox**:



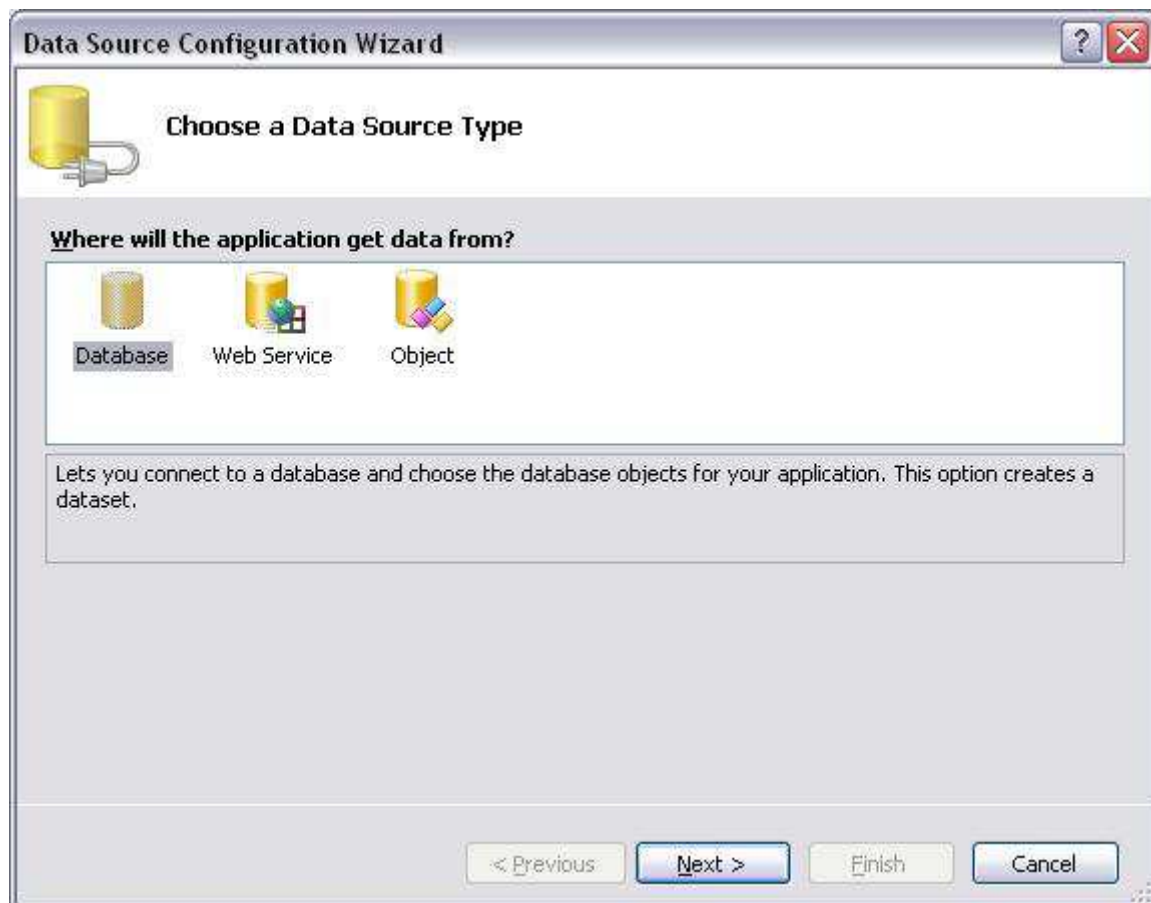
Ustaw właściwość DataSource w kontrolce **BindingSource**



Nie ma aktualnie żadnej bazy danych podpiętej do projektu dlatego DataSource jest pusty. Wybierz AddProject Data Source



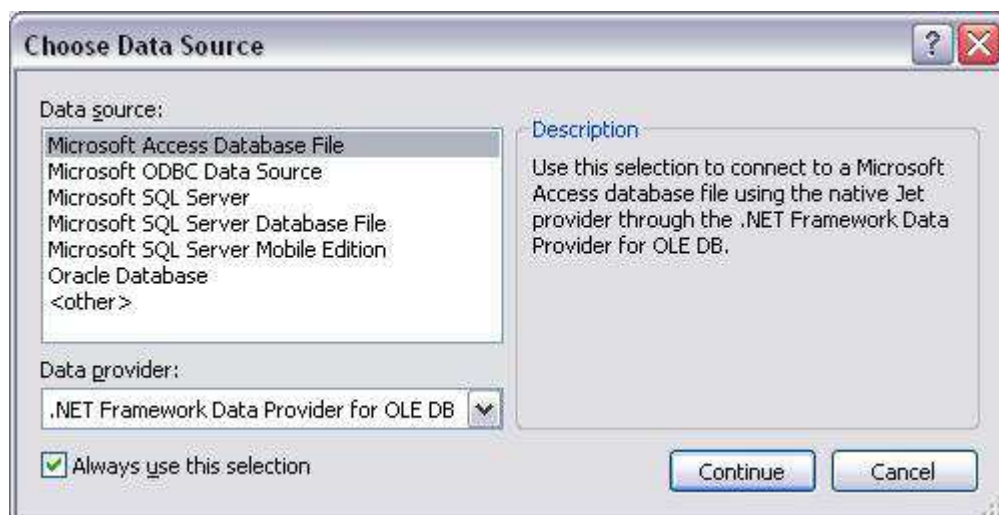
Pozwoli to na uruchomienie kreatora tworzącego połączenie z bazą danych



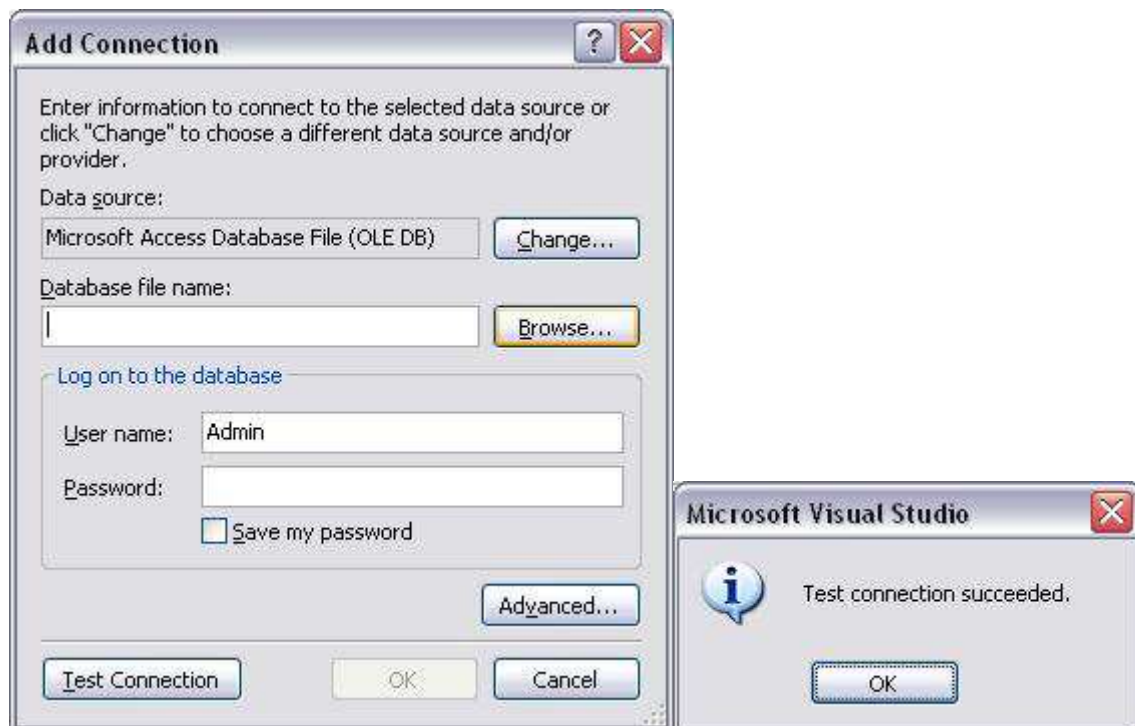
Wybieramy **Database**, która umożliwia połączenie aplikacji z bazą danych taka jak *Microsoft SQL Server*, *Access*, *Oracle*:



Wybieramy rodzaj połączenia - **New Connection...** i wyświetla się dialog **Choose Data Source**:

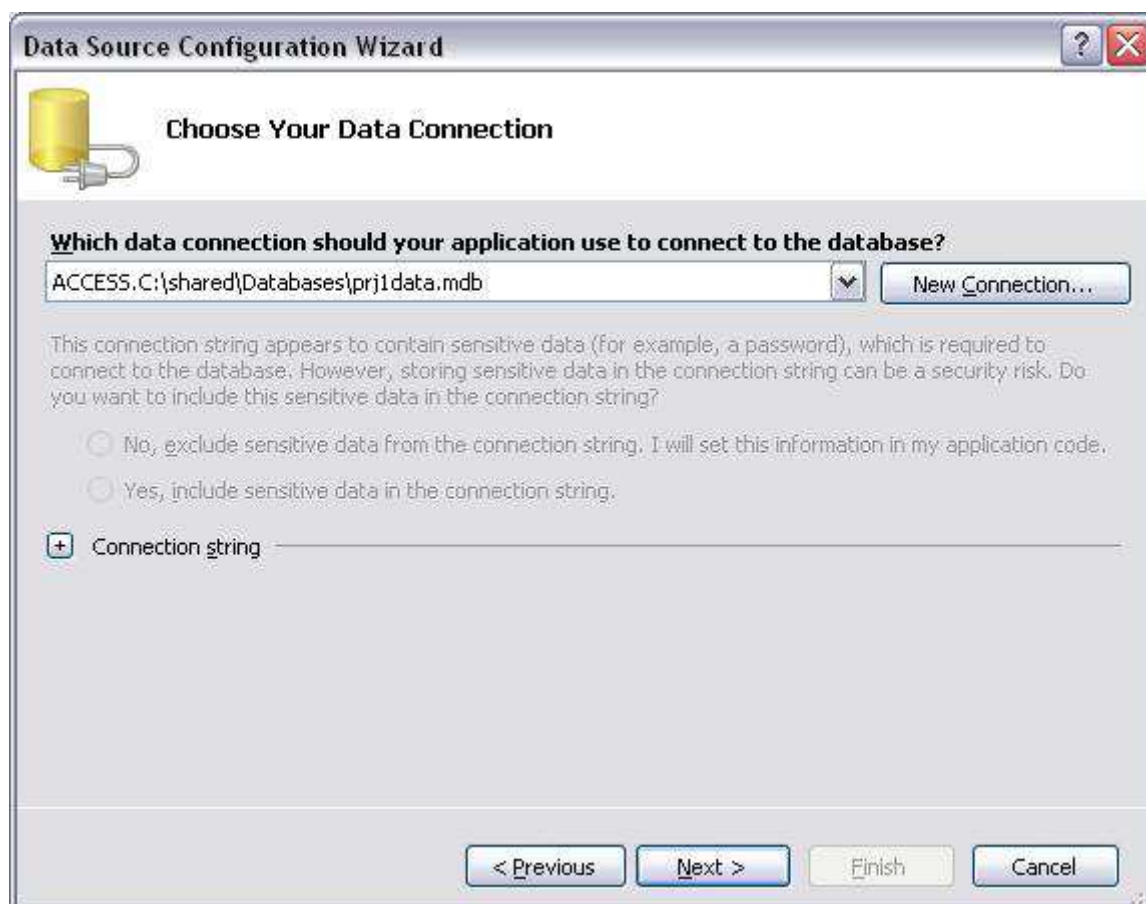


Tutaj wybieramy typ providera oraz rodzaj bazy danych **Microsoft Access Database File** i klikamy na **Continue** a celu wygenerowania końcowego połączenia



Tutaj specyfikujemy z jaką bazą chcemy się połączyć (wybierz **Browse...** a następnie znajdź baze danych) Na końcu przetestuj połączenie **Test Connection**

**Choose Your Data Connection** powinien wyglądać tak jak niżej na rysunku. Następnie wybieramy Next



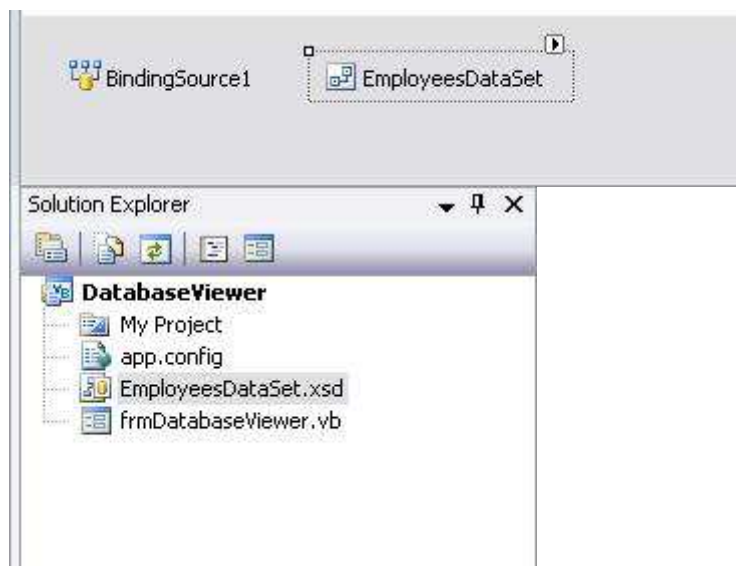
Kolejnym krokiem jest wybór źródła danych (tabel lub kwerend SQL)

Poniżej widok dostępnych tabel i widoków z wybranej bazy danych - wybrane wygenerują DataSet





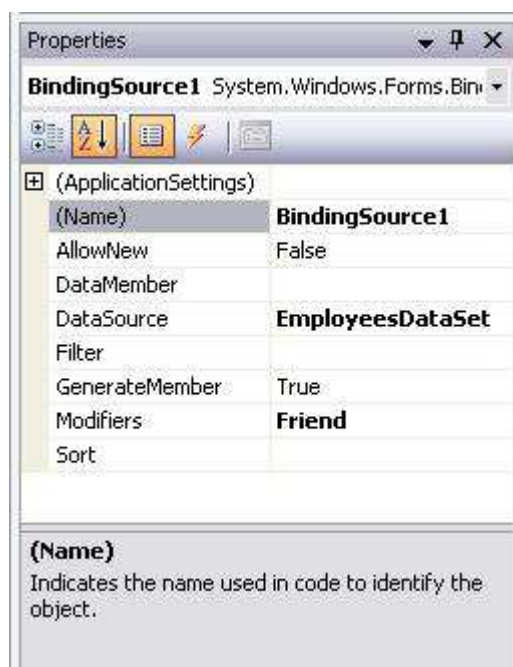
Aktualnie nasza aplikacja wygląda tak::



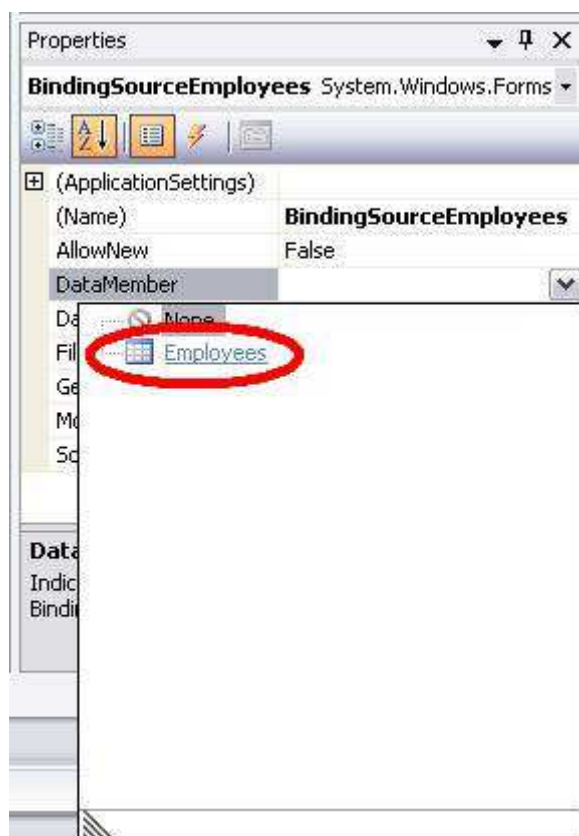
#### 4. Generowanie DataSet w projekcie.

Kliknij na **BindingSource1** w celu wyświetlenia okienka właściwości.

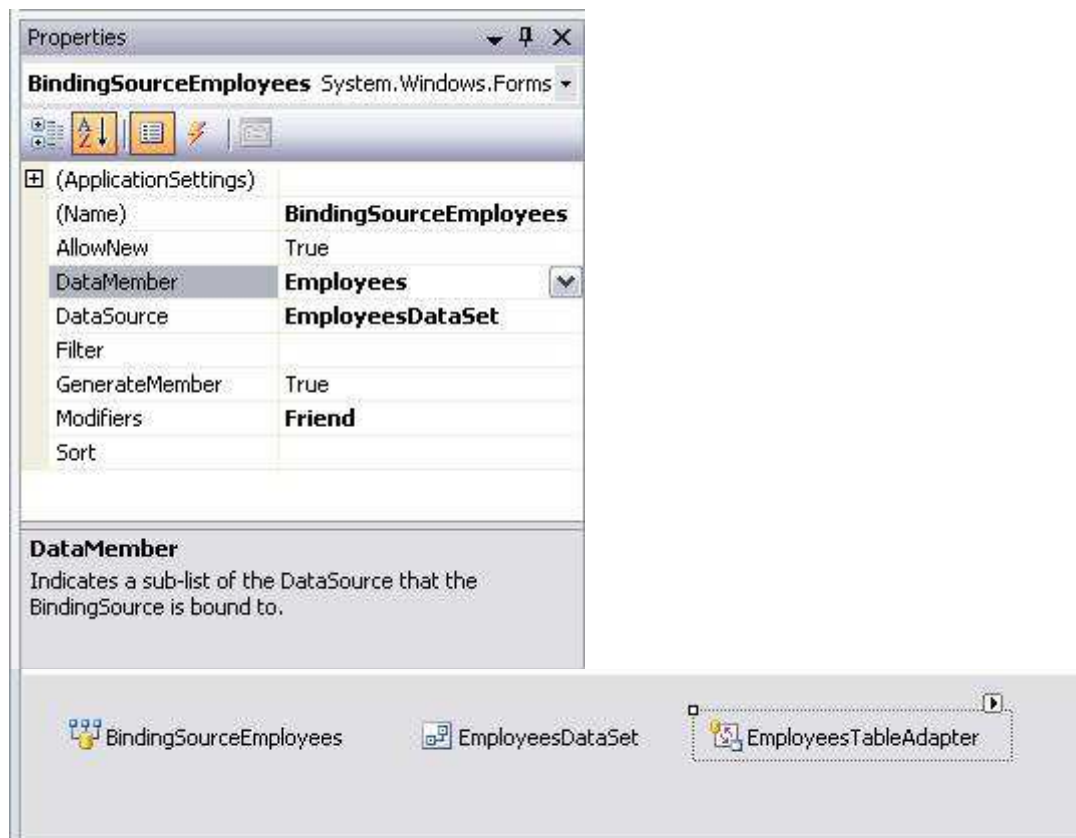




Zmień **Name** property a następnie wybierz właściwość **DataMember** w której należy wybrać dowolną tabelę:



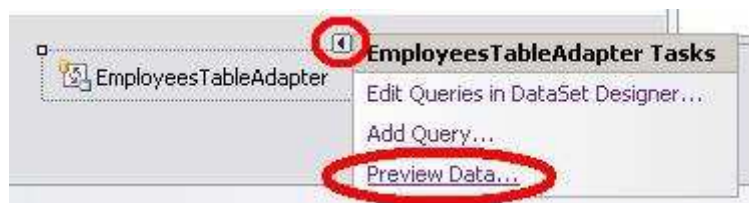
Teraz projekt będzie wyglądał mniej więcej tak::



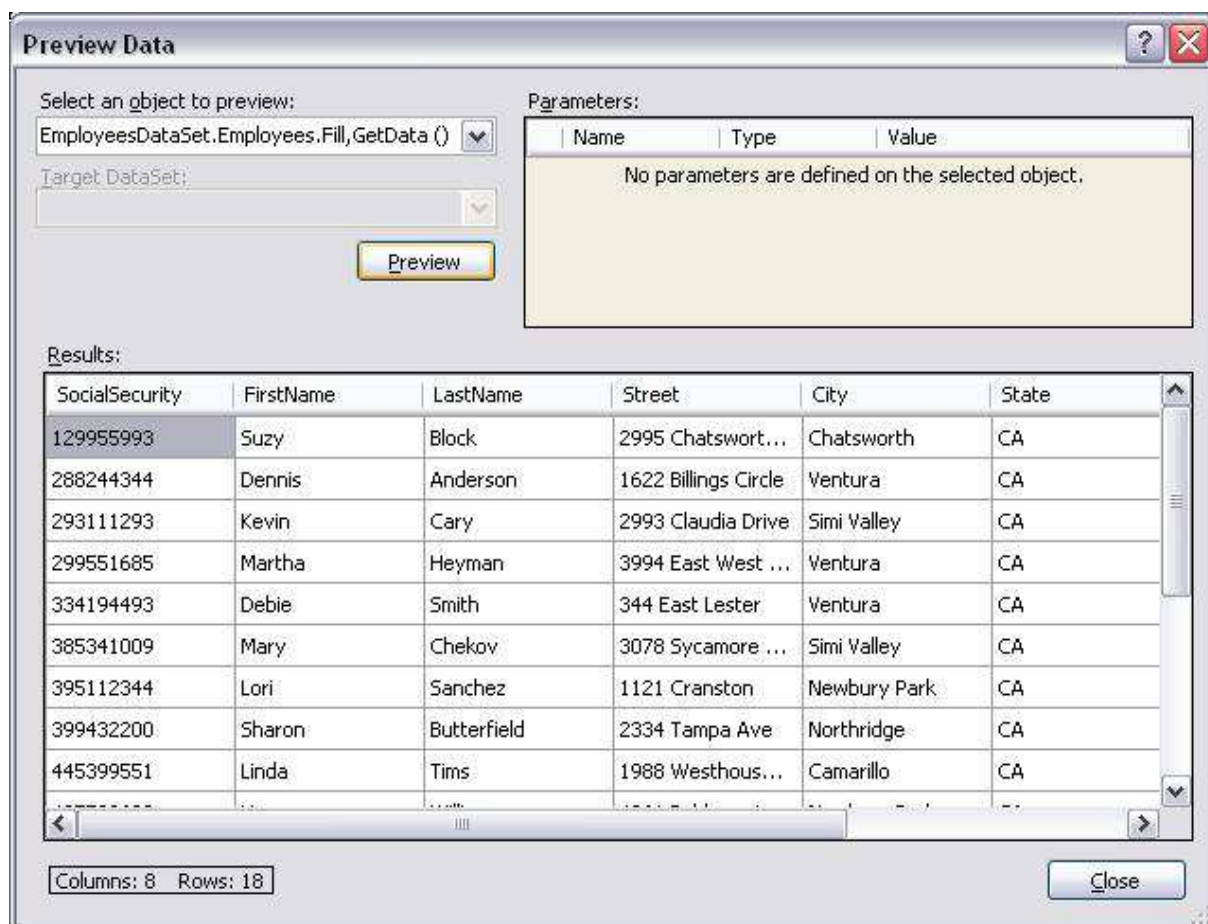
**TableAdapters** jest pośrednikiem pomiędzy projektem a bazą danych. Kontrolka ostatnio pokazana odpowiada za połączenie z wybraną tabelą oraz wypełnia kontrolkę typu DataSet danymi

## 5. Podgląd danych

Kliknij na **EmployeesTableAdapter** i wybierz **Preview Data**.



Wyświetli się okno typu podgląd danych



## Dostęp do bazy danych – podsumowanie

1. Dodanie *BindingSource* do projektu
2. Ustawienie **właściwości DataSource** kontrolki **BindingSource** poprzez odpalenie kreatora konfiguracji:
  1. Wybranie typu źródła danych
  2. Wybór bazy danych .
  3. Przetestowanie połączenia .
  4. Wybranie elementów z bazy danych w celu włączenia do **Dataset**.
  5. Stworzenie kontrolki **Dataset** (EmployeesDataSet).
3. **Dataset** (*EmployeesDataSet*)  
 Została stworzona za pomocą konfiguratora i zawiera rekordy z tabeli wybranych w procesie konfiguracji).
4. Stworzenie **TableAdapter** (*EmployeesTableAdapter*)  
 Ustawienie właściwości DataMember w BindingSource na wybraną tabelę pozwoliło na stworzenie *TableAdapter*. Kontroluje ono połączenie do wybranej tabeli w bazie danych , wykonuje zapytania i zapełnia danymi Data.

## 6. Zaprojektowanie interfejsu aplikacji

Database Viewer

Social Security:

Last Name:

First Name:

Street:

City:

State:

Zip Code:

Age:

Record:  of

<< < > >>

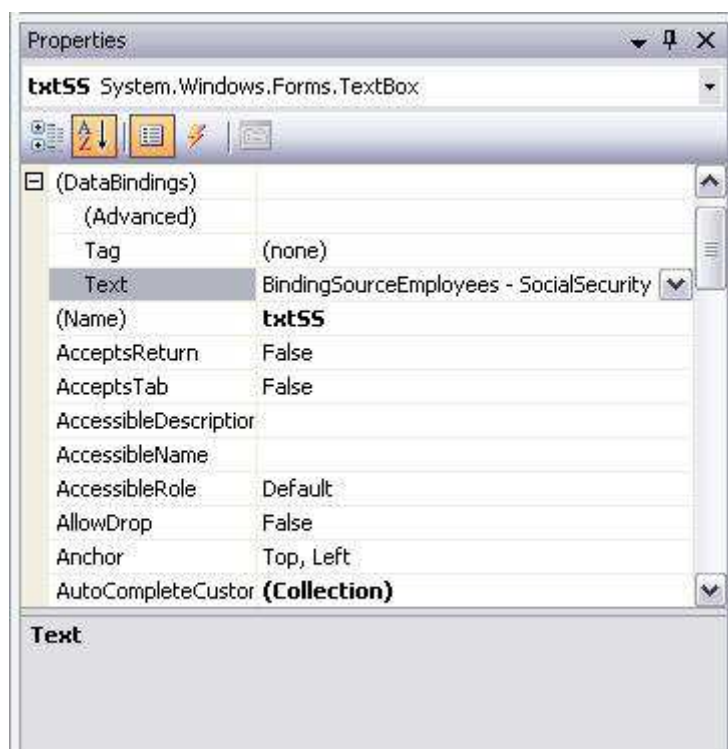
Add Edit Save Cancel Delete

Każde pole testowe powinno mieć unikalna nazwę na przykład takie:

Property	Value
Name	txtSS
Name	txtLast
Name	txtFirst
Name	txtStreet
Name	txtCity
Name	txtState
Name	txtZip
Name	txtAge

## 7. Powiązanie pól tekstowych z DataSet

Dal wybranego pola tekstowego w oknie właściwości wybierz właściwość **(DataBindings)** a następnie podwłaściwość **Text**



Dla każdego pola tekstowego wybierz źródło danych

## 8. Wyświetlenie danych w formularzu

Aby wyświetlić aktualne wartości w formularzu należy zapełnić DataSet

W tym celu dla zdarzenia formy **Load** należy napisać przykładowy kod:

```
EmployeesTableAdapter.Fill( _  
    EmployeesDataSet.Employees)
```

## 9. Test programu

## 10 Navigacja po bazie

### Next button

*Clik procedure*

```
'zwiększenie właściwości Position  
' BindingSourceEmployees aby przesunąć się do przodu  
' Jeśli Position =0 to jesteśmy na pierwszej pozycji  
BindingSourceEmployees.Position += 1
```

## 11 Dodanie First, Previous, i Last buttons

zdarzenie Click dla Previous:

**BindingSourceEmployees.Position -= 1**

zdarzenie Click dla First:

**BindingSourceEmployees.Position = 0**

zdarzenie Click dla Last:

**BindingSourceEmployees.Position = \_  
BindingSourceEmployees.Count - 1**

## 12. Modyfikacje interfejsu- licznik pozycji i ilości wierszy



**lblCurrRow.Text = \_  
BindingSourceEmployees.Position + 1**

**lblTotRows.Text = \_  
BindingSourceEmployees.Count**

**Edycja rekordów**

Database Viewer

Social Security:	385341009
Last Name:	Chekov
First Name:	Mary
Street:	3078 Sycamore Road
City:	Simi Valley
State:	CA
Zip Code:	93065
Age:	32

Record: 6 of 18

<< < > >> Edit Save Cancel

**btnEdit:**

**btnFirst.Enabled = False**  
**btnPrevious.Enabled = False**  
**btnNext.Enabled = False**  
**btnLast.Enabled = False**  
**btnEdit.Enabled = False**  
**txtSS.ReadOnly = False**  
**txtLast.ReadOnly = False**  
**txtFirst.ReadOnly = False**  
**txtStreet.ReadOnly = False**  
**txtCity.ReadOnly = False**  
**txtState.ReadOnly = False**  
**txtZip.ReadOnly = False**  
**txtAge.ReadOnly = False**  
**txtSS.Focus()**

**btnSave**

*' iCurrentRecord zmienna przechowująca aktualną pozycję*

**Dim iCurrentRecord As Integer**  
*' EndEdit dla BindingSourceEmployees*  
*' potwierdza zmiany w EmployeesDataSet,*  
*' ale nie zatwierdza ich w bazie .*

**BindingSourceEmployees.EndEdit()**

**iCurrentRecord = BindingSourceEmployees.Position**  
*' zaszła miana od ostatniego Update.*

**If EmployeesDataSet.HasChanges() = True Then**



*'w celu zatwierdzenia zmian .*

**EmployeesTableAdapter.Update(EmployeesDataSet)**

*'po zatwierdzeniu należy przeładować dataSet.*

**EmployeesTableAdapter.Fill( \_  
EmployeesDataSet.Employees)**

**End If**

**txtSS.ReadOnly = True**

**txtLast.ReadOnly = True**

**txtFirst.ReadOnly = True**

**txtStreet.ReadOnly = True**

**txtCity.ReadOnly = True**

**txtState.ReadOnly = True**

**txtZip.ReadOnly = True**

**txtAge.ReadOnly = True**

**btnEdit.Enabled = True**

**BindingSourceEmployees.Position \_  
= iCurrentRecord**

1. **EndEdit Binding Source** w celu uaktualnienia **Dataset** o modyfikacje.
2. **HasChanges Dataset** - sprawdzenie czy zaszły modyfikacje w DataSet .
3. **Update Table Adapter** potwierdzenie zmian z **Dataset** w **Data Source** (w pliku bazy danych).
4. **Fill Data Table** ponowne przeładowanie **Dataset** danymi z źródła **Data Source**.

**btnCancel:**

*'odwołuje wszystkie zmiany od ostatniego Update.*

**BindingSourceEmployees.CancelEdit()**

**txtSS.ReadOnly = True**

**txtLast.ReadOnly = True**

**txtFirst.ReadOnly = True**

**txtStreet.ReadOnly = True**

**txtCity.ReadOnly = True**

**txtState.ReadOnly = True**

**txtZip.ReadOnly = True**

**txtAge.ReadOnly = True**

**BindingSourceEmployees\_PositionChanged( \_  
Nothing, Nothing)**

.

**btnAdd**

```
.  
txtSS.ReadOnly = False  
txtLast.ReadOnly = False  
txtFirst.ReadOnly = False  
txtStreet.ReadOnly = False  
txtCity.ReadOnly = False  
txtState.ReadOnly = False  
txtZip.ReadOnly = False  
txtAge.ReadOnly = False  
txtSS.Focus()  
iStartRec = BindingSourceEmployees.Position  
BindingSourceEmployees.AddNew()
```

***btnDelete:***

```
Dim iResponse As DialogResult  
  
iResponse = MessageBox.Show( _  
    "Czy chcesz na pewno usunąć wiersz ?", _  
    "potwierdz", MessageBoxButtons.YesNo, _  
    MessageBoxIcon.Question)  
If iResponse = Windows.Forms.DialogResult.Yes Then  
    BindingSourceEmployees.RemoveAt( _  
        BindingSourceEmployees.Position)  
End If
```

**Przykład interfejsu do zadania 1**

Database Viewer

Social Security:

Last Name:

First Name:

Street:

City:

State:

Zip Code:

Age:

☐

☒ Find - LastName:

☐ Find First ☐ Find Next

☐

Record:  of

<< < > >>

Add Edit Save Cancel Delete