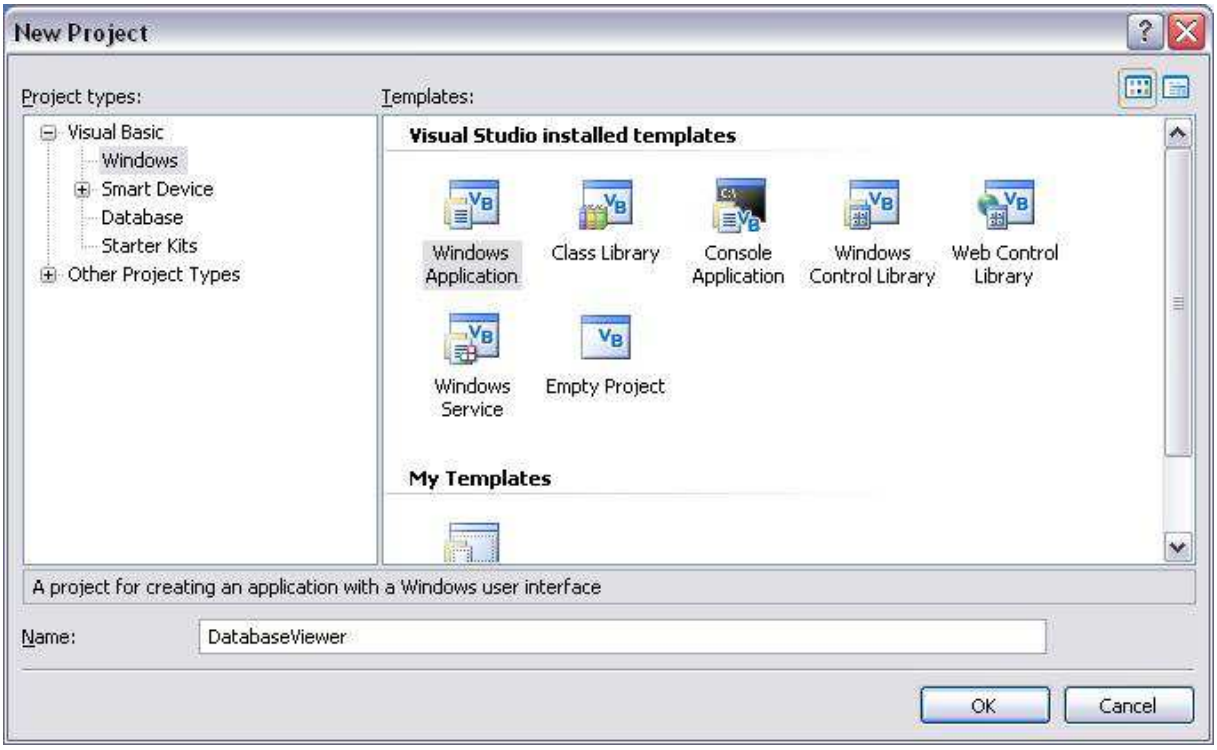
Przykład pokazujący jak stworzyć aplikacje pozwalającą na przeglądanie struktury bazy danych

1. Przeglądarka struktury bazy danej

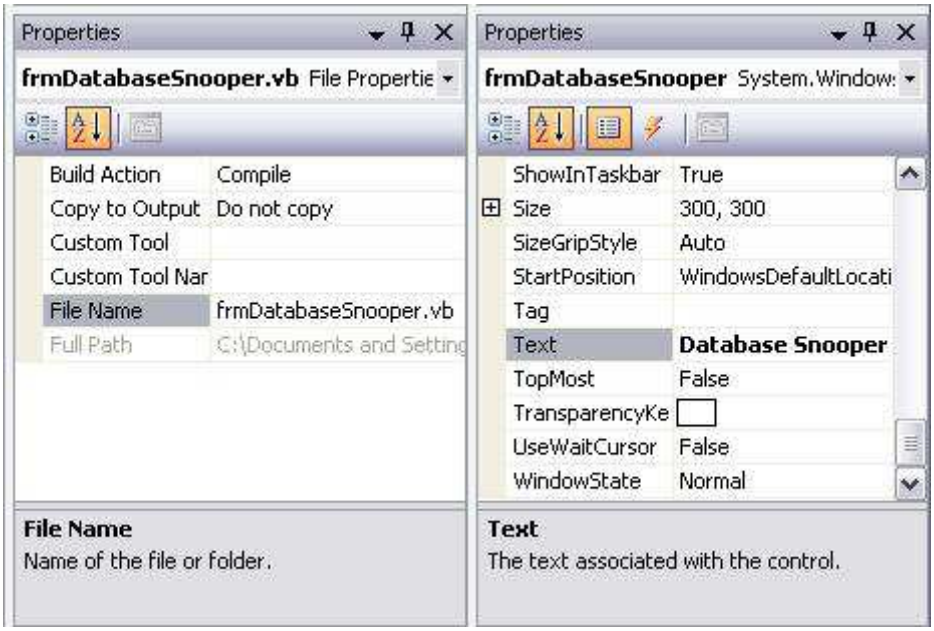
Przeglądarka bazy danych pozwoli użytkownikowi na przeglądanie komputera lub sieci w poszukiwaniu znajdujących się tam plików Access. Następnie umożliwi zbadanie zawartości plików bazy danych Microsoft Access.

Funkcjonalność docelowa:
1. Pozwala użytkownikowi przeglądać komputer lub siec, aby wybrać plik bazy danych.
2. Lista wszystkich tabel, które zawiera plik bazy danych.
3. Wyświetla kolumny, nazwy i ich typów danych dla tabeli wybranej przez użytkownika z listy tabel.
4. Wyświetla rekordy wybranej tabeli w kontrolce DataGridView.
5. Pozwala użytkownikowi na zmianę kolejności sortowania rekordów w siatce danych, wybierając pole *sortować według* z listy pól.

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.



Ze względu na funkcjonalność aplikacji nie można nawiązać połączenia z bazą danych w czasie projektowania. W tworzonym właśnie projekcie musimy pozwolić użytkownikowi na wybranie plików bazy danych dynamicznie, w czasie wykonywania. Nie można wiec użyć w czasie projektowania żadnych kontrolek typu BindingSource czy kreatorów typu Data Source Configuration.

3. Zaprojektowanie głównego formularza aplikacji

Umieść na głównym formularzu następujące kontrolki tak jak pokazuje rysunek poniżej:

- Dodatkowo umieść na formularzu kontrolkę **OpenFileDialog**. Jest to kontrolka należąca do grupy kontrolek niewizualnych ale dostarczająca możliwości uzyskania standardowego okna typu Open.

. 4. Krokiem następnym jest dodanie kodu dla przycisku **Open Database** pozwalającego na wyświetlenie okna typu **OpenFileDialog**. Pozwoli ono użytkownikowi na wybór bazy danych.

Wstepny kod dla procedury ***btnOpen_Click***:

```
Dim iResult As DialogResult
' Ustwienie tytułu oraz filtru taka by tylko pliki z rozszerzeniem.mdb były wyświetlane
OpenFileDialog1.Title = "Select a Database File"
OpenFileDialog1.Filter = "MDB Files (*.mdb) | *.mdb"


OpenFileDialog1.CheckFileExists = True
iResult = OpenFileDialog1.ShowDialog()

If iResult <> Windows.Forms.DialogResult.Cancel And  _
        OpenFileDialog1.FileName.Length <> 0 Then

End if

End Sub
```

Kodem tym wybieramy plik bazodanowy ale nic się nie dzieje. Musimy go teraz rozszerzyć o kawałek kodu pozwalający na otwarcie połączenia z wybraną bazą oraz stworzenie dynamicznie całej infrastruktury.

W tym celu należy najpierw w sekcji ***Declarations*** dodać bibliotekę

```
Imports System.Data.OleDb  oraz

Dim dbConnection As OleDbConnection
```

A następnie wewnątrz instrukcji warunkowej

```
If iResult <> Windows.Forms.DialogResult.Cancel And  _
        OpenFileDialog1.FileName.Length <> 0 Then

End if
```

Dodac następujacy kod:

```
Dim sConStr As String
‘tutaj budujemy ConnectionString składajac go z cześci stałej oraz
```

```
‘ dynamicznej reprezentującej nazwę pliku
sConStr = "Provider=Microsoft.Jet.OLEDB.4.0;"
sConStr &= "Data Source=" &  OpenFileDialog1.FileName & ";"

If Not dbConnection Is Nothing Then
   dbConnection.Close()
   dbConnection = Nothing
End If
dbConnection = New OleDbConnection(sConStr)
dbConnection.Open()
```

Powyższa procedura pozwala na nawiązanie połączenia z bazą danych. Krok kolejny to identyfikacja zawartości bazy danych. Zidentyfikowane tabele zostaną wyświetlone w ListBox widocznym na formularzu.

5. Identyfikacja tabel.

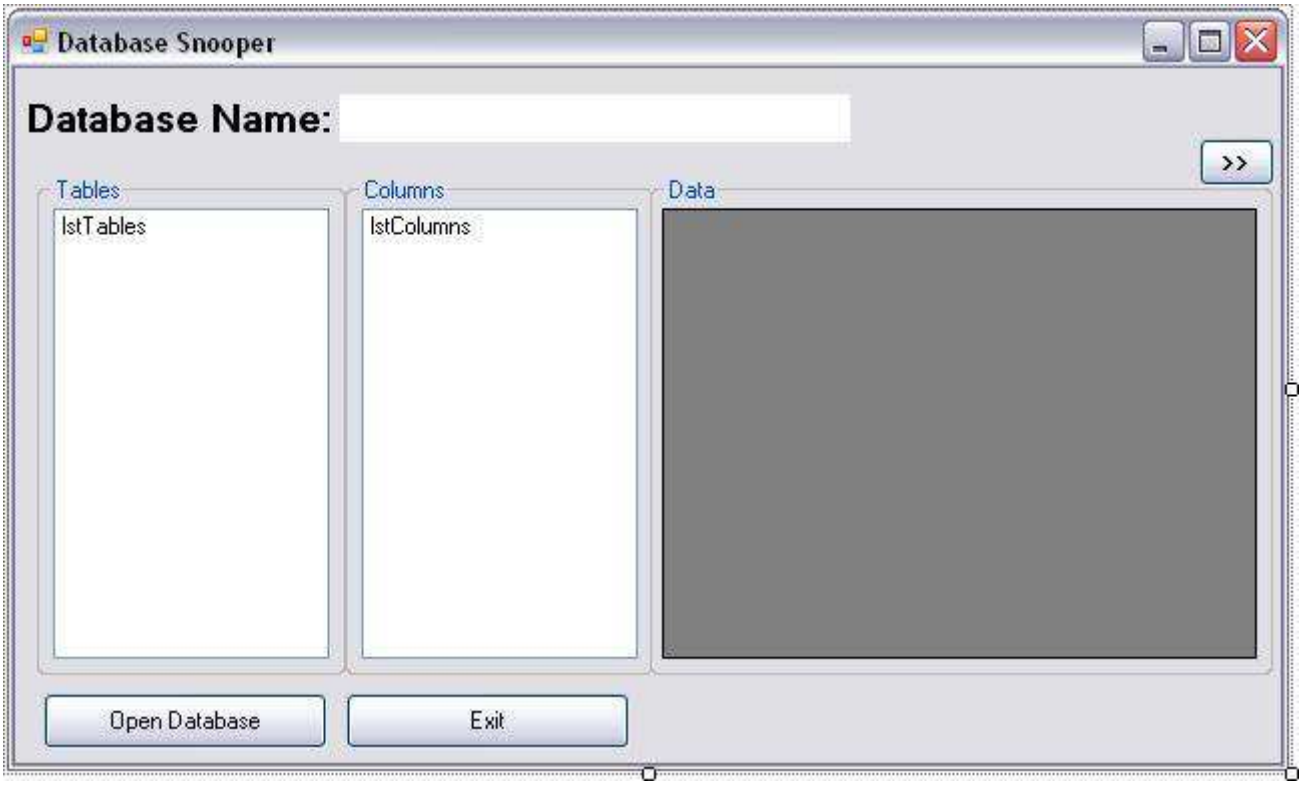
Do identyfikacji tabel skorzystano z metody GetOleDbSchemaTable pozwalającej na wyliczenie tabele w bazie danych
Nasz obiekt OleDbConnection (dbConnection) ma metodę GetOleDbSchemaTable. GetOleDbSchemaTable zwraca obiekt DataTable, zawierającą nazwy i inne informacje na temat tabel w bazie danych.
Dodaj następujący kod do procedury zdarzenia btnOpen_Click poniżej linii kodu dbConnection.Open () wewnątrz instrukcji warunkowej

```
Dim schemaTable As DataTable = _
      dbConnection.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, _
      New Object() {Nothing, Nothing, Nothing, "TABLE"})
dbConnection.Close()

'NA wszelki wypadek czyścimy ListBox  lstTables przed  dodaniem
lstTables.Items.Clear()
For i = 0 To schemaTable.Rows.Count - 1
   'Kolekcja  Items to kolumny w  schemaTable.  Kolumna 0 i 1 jest Null,
   '    kolumna 2 zawiera nazwę tabeli .
   lstTables.Items.Add(schemaTable.Rows(i).Item(2))
Next i
```

Nie zapomnij o zdefiniowaniu i jako liczba całkowita na początku procedury btnOpen_Click.

6. Rozszerzamy widok formularza o kolejne kontrolki zgodnie z widokiem poniżej:



Dodajemy więc kolejny GrupBox i ListBox oraz DataGridView oraz Buton>>.

Nowy ListBox będzie wyświetlał nazwy kolumn dla tabeli wybranej z listy Tabel bazy danych.

W tym celu potrzebujemy w sekcji ***Declarations*** dodać referencje do obiektu **OleDbDataAdapter**

```
Dim dbAdapter As OleDbDataAdapter = New Oledb.OleDbDataAdapter()
```

Następnie generujemy procedurę dla ***lstTables_SelectedIndexChanged*** (zmiana wybranej pozycji w listBox z tabelami):

W jej kodzie wpisujemy:

```
Dim sFieldInfo, sDataType As String
Dim sTableName As String
Dim i As Integer
'wybrana pozycja z listy table w sTableName
sTableName = lstTables.Items(lstTables.SelectedIndex)
Dim selectCMD As OleDbCommand = _
        New OleDbCommand("SELECT * FROM " & _
              "[" & sTableName & "]", dbConnection)

'wiążemy adapter z komedą SQL
dbAdapter.SelectCommand = selectCMD
```

```
' Tworzymy obiekt  DataSet do przechowaywania  informacji o wybranej tabeli Dim
dbDataSet As DataSet = New DataSet()
dbDataSet.Clear()
dbAdapter.Fill(dbDataSet, sTableName)

'czyścimy listBox z nazwami kolumn
lstColumns.Items.Clear()


'odczytujemy nazwy kolumn i  umieszczamy w liscie

For i = 0 To dbDataSet.Tables(0).Columns.Count - 1
      With dbDataSet.Tables(0).Columns(i)
        sDataType = .DataType.ToString.Substring( _
            InStrRev(.DataType.ToString, "."))
            sFieldInfo = .ColumnName & " -- " & sDataType.ToLower
      lstColumns.Items.Add(sFieldInfo)
   End With
Next i
```

7. *Dodajemy kod potrzebny do wyświetlenia w DataGridView wybranej tabeli*

W powyższej procedurze ***lstTables_SelectedIndexChanged*** poniżej linii **Next i** dodać następujący kod:

```
Dim MyBindingSource As New BindingSource
Dim MyDataTable As New DataTable

dbAdapter.Fill(MyDataTable)

'stworzyć powiazanie pomiedzy DataTable a BindingSource

MyBindingSource.DataSource = MyDataTable

DataGridView1.DataSource = MyBindingSource
```

Zadania dodatkowe do wykonania:

• Wyświetlanie nazwy pliku bazy danych
Jest już etykiety (lblDatabaseName), aby wyświetlić nazwę wybranego pliku bazy danych. Proszę to zrobić samodzielnie.
• Przycisk >>
przycisk >> ma umożliwić użytkownikowi rozszerzenie szerokości dateGrid, aby zobaczyć więcej kolumn kontroli DataGridView. Spróbowac samodzielnie to uzyskać .

• Sortowanie pozycji w DataGridView przez wybranego pola w listbox z kolumnmi
Gdy użytkownik kliknie na wartości w polu listy kolumn (lstColumns) upewnić się, że rekordy w DataGridView posortowane są według tej dziedzinie. (Wskazówka: Skorzystaj z pomocy jednej karty danych własności SelectCommand.CommandText).