

LABORATORIUM SYSTEMÓW MOBILNYCH

ZAPOZNANIE SIĘ ZE SPOSOBEM PRZECHOWYWANIA DANYCH NA URZĄDZENIACH MOBILNYCH

I. Temat ćwiczenia

Zapoznanie się ze sposobem przechowywania danych na urządzeniach mobilnych

II. Wymagania

Podstawowe wiadomości z zakresu obsługi baz danych i języka SQL

III. Ćwiczenie

1. Wprowadzenie

Microsoft Compact Framework umożliwia użytkownikom wybór sposobu przechowywania danych na urządzeniu mobilnym.

1.1. Pliki binarne

Stanowią jedną z podstawowych form przechowywania danych. Charakteryzują się niską zajętością pamięci i małą nadmiarowością danych strukturalnych. Ponadto są ściśle związane z daną architekturą komputera co czyni je trudno przenaszalnymi.

1.2. Pliki XML

Podobnie jak pliki INI, dzięki ustrukturyzowanej postaci, pliki XML stanowią podstawowy mechanizm przechowywania danych w systemach heterogenicznych, w których sposób zapisu danych jest uniezależniony od wewnętrznej architektury aplikacji. Wiąże się to ze znacznym wzrostem rozmiarów plików spowodowanym koniecznością występowania dodatkowych danych strukturalnych oraz tekstowym przechowywaniem informacji. Microsoft Compact Framework dostarcza w pełni funkcjonalne klasy *XmlDocument* oraz *XmlReader* jednak brak jest mechanizmu walidacji, XPath oraz Generycznej Serializacji.

1.3. Baza danych

Baza danych stanowi alternatywę w mechanizmie przechowywania danych zarówno dla plików binarnych jak i plików XML. Użytkownik tworzący aplikację dla urządzenia mobilnego może skorzystać z bazy danych Microsoft SQL Server przeznaczonej dla urządzeń stacjonarnych bądź też z wersji *Microsoft SQL Server Mobile Edition* bazy danych, przeznaczonej do pracy na urządzeniach mobilnych. Decydując się na rozwiązanie z bazą danych zlokalizowaną na urządzeniu stacjonarnym niezbędne jest zapewnienie dostępu do danych m.in. wykorzystując mechanizm WebService'ów.

Istnieje szereg ograniczeń w dostępie do danych zlokalizowanych w bazie danych Microsoft SQL Server z urządzenia Pocket PC. Ograniczenia te związane są przede wszystkim dostępnością typów danych, ich wsparciem ze strony Pocket PC i ich mapowaniem. Szczegółowe informacje dotyczące dostępnych typów danych i ich mapowaniem można znaleźć pod adresem: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sqlce/htm/_lce_accessing_data.asp

1.4. Mechanizm przechowywania danych

Tworząc aplikacje bazodanowe z wykorzystaniem technologii Microsoft Framework bądź też Microsoft CompactFramework musimy zaznaczyć się z nowym schematem przechowywania i dostępu do danych, który realizuje technologia *Microsoft ADO.NET*. Technologia ta umożliwi programiście uzyskanie połączeniowego jak i bezpołączeniowego dostępu do danych, z których każdy charakteryzuje się określonymi cechami.

Dostępne tryby:

- **Tryb połączeniowy** – bardzo popularny wśród rozwiązań i aplikacji pracujących na urządzeniach stacjonarnych, w których to fizyczny dostęp do danych z reguły jest zapewniony. Rozwiązanie to charakteryzuje się szybkim dostępem do danych oraz wykonywaniem wszystkich DML bezpośrednio na źródle danych. Minusem tego podejścia jest konieczność zapewnienia stałego połączenia z bazą danych przez cały czas działania aplikacji. Podejście to realizuje obiekt klasy *SqlDataReader*,
- **Tryb bezpołączeniowy** – wydajny w przypadku aplikacji mobilnych oraz rozwiązań, w których to fizyczny dostęp do bazy danych nie może być zapewniony przez cały czas życia aplikacji. Na uwagę zasługuje tutaj obiekt *DataSet*, który jest reprezentacją struktury relacji znajdującej się w określonej bazie danych. Obiekt ten ma budowę hierarchiczną i składa się m.in. z:
 - **DataTable** – odpowiednik pojedynczej relacji z bazy danych,
 - **DataRow** – element obiektu DataTable będący odpowiednikiem krotki,
 - **DataColumn** – element obiektu DataTable, odpowiednik atrybutu relacji,
 - **DataRelation** – odpowiednik ograniczenia referencyjnego schematu relacji,

W momencie nawiązania połączenia z bazą danych, na obiekcie DataSet możemy wykonać jedną z dwóch podstawowych czynności:

- **Fill** – wypełnienie obiektu DataSet bądź konkretnego obiektu DataTable w wyniku wykonania zapytania SQL,
- **Update** – aktualizacja źródła danych na podstawie modyfikacji danych wprowadzonych w obiekcie DataSet. Metoda ta realizuje polecenia DML,

Na obiekcie DataSet możemy operować w podobny sposób jak na fizycznym źródle danych.

DataSet występuje w dwóch rodzajach:

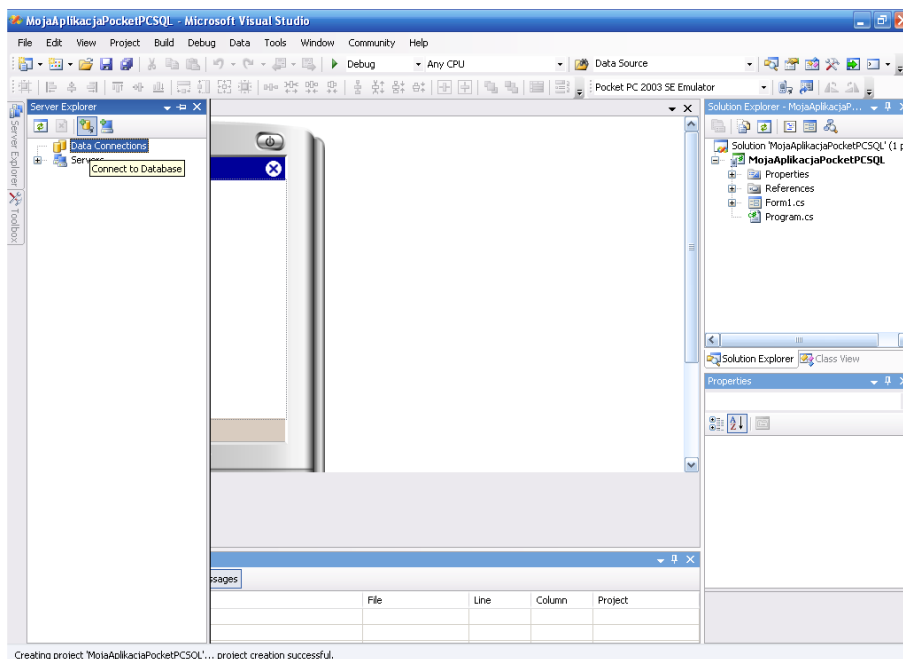
- **Typowany obiekt DataSet** – reprezentuje on własny schemat klasy DataSet, który zawarty jest w pliku z rozszerzeniem *.xsd*. Dostęp do danych w nim zawartych odbywa się poprzez schemat:
`<nazwa_obiektu_DataSet>.<nazwa_obiektu_DataTable>`
- **Generyczny DataSet** – jest to standardowa instancja klasy DataSet udostępniana przez technologię ADO.NET. Jest on bardziej niewygodny w odwoływaniu się do poszczególnych jego składowych, gdyż realizowane jest za pomocą następującego schematu:
`<nazwa_obiektu_DataSet>.Tables[\"nazwa_relacji_bazodanowej\"].Rows`

2. Utworzenie aplikacji mobilnej z wykorzystaniem Microsoft SQL Server Mobile Edition

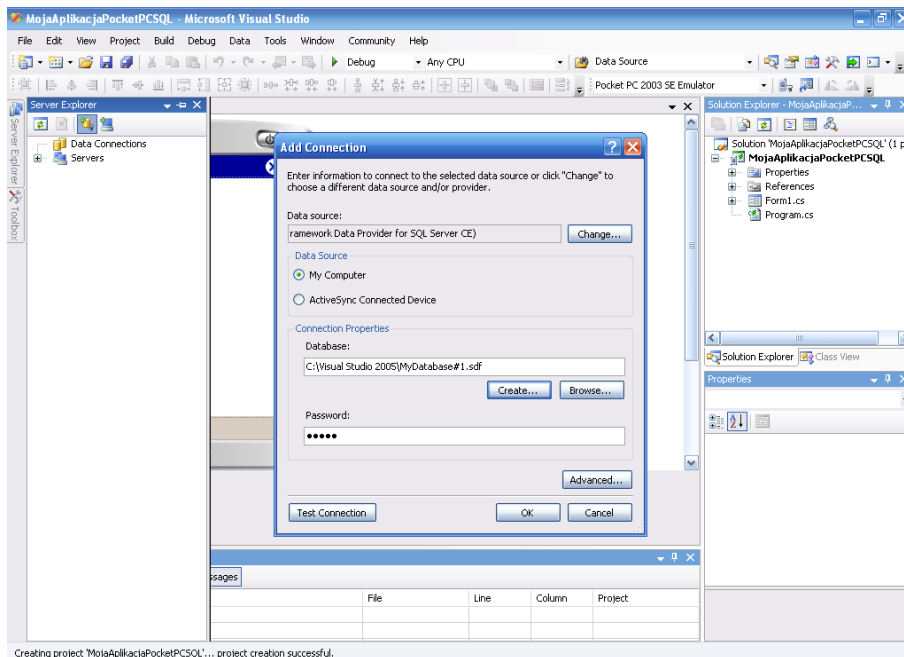
Stworzenie aplikacji na urządzenie mobilne wykorzystujące bazę danych możemy rozpocząć od przygotowania schematu bazy danych. Wykonać to możemy bezpośrednio z poziomu środowiska Visual Studio. Aby sprawdzić czy serwer bazy danych jest uruchomiony, wykorzystać możemy narzędzie *SQL Server Configuration Manager* i w razie konieczności uruchomić serwer bazy danych ręcznie. Jeżeli nie jest możliwe włączenie usługi należy wybrać własność **Properties** i na zakładce **Service** ustawić pole **Start Mode** w tryb **Manual** lub **Automatic**.

Po uruchomieniu środowiska Visual Studio 2005 i stworzeniu pustej aplikacji dla urządzenia Pocket PC (patrz laboratorium 1) możemy przystąpić do zdefiniowania schematu relacji bazy danych. Jako baza danych posłuży nam *MS SQL Server 2005 Mobile Edition*.

Zdefiniowanie schematu bazy danych rozpoczynamy od utworzenia połączenia z bazą danych. W tym celu z dostępnej zakładki **Server Explorer** wybieramy **Connect to Database** (Rysunek 1).



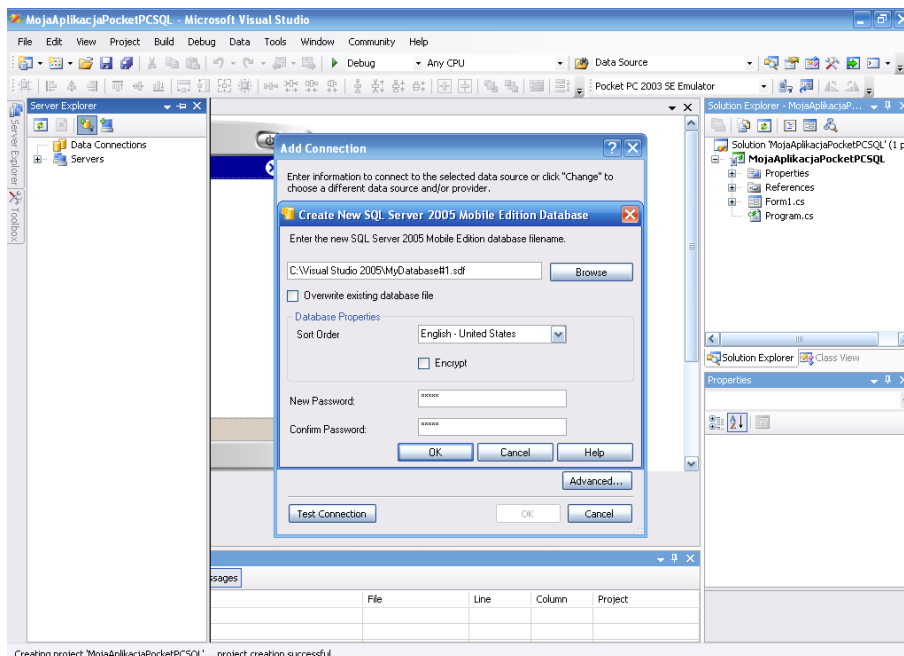
Rysunek 1 – Zakładka Server Explorer



Rysunek 2 – Okno dialogowe tworzenia połączenia z bazą danych

W wyświetlonym oknie dialogowym (Rysunek 2) poleceniem **Change** definiujemy rodzaj bazy danych z której będziemy korzystać. Wybieramy opcję **Microsoft SQL Server Mobile Edition**. W kolejnym kroku jako **Data source** wybieramy **My Computer**, a następnie tworzymy nową bazę danych poprzez polecenie **Create**.

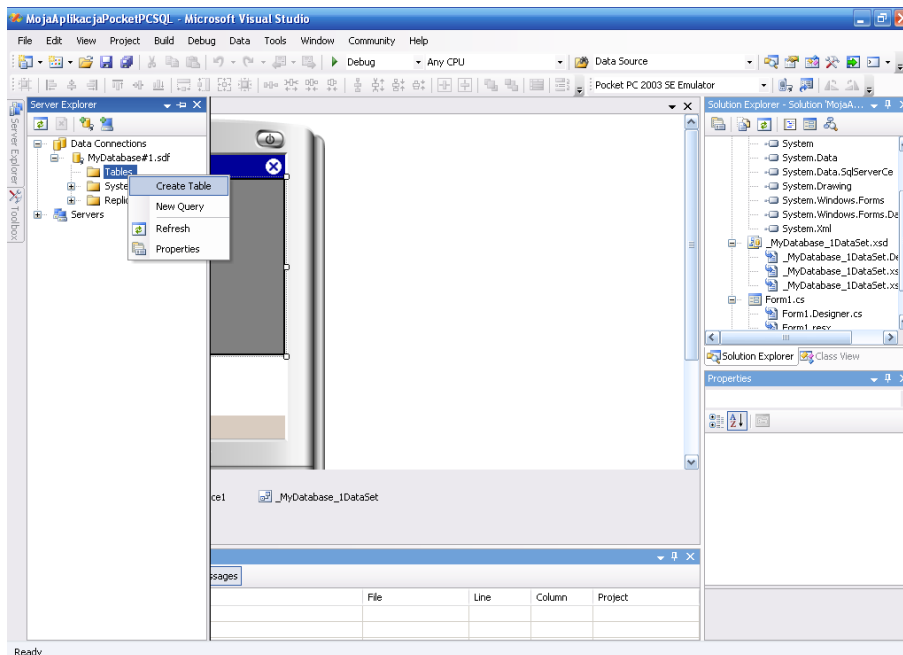
W nowym oknie dialogowym wybieramy lokalizację dla tworzonej bazy danych (polecenie **Browse**), oraz podajemy hasło dostępu do bazy danych (Rysunek 3).



Rysunek 3 – Okno dialogowe tworzenia nowej bazy danych

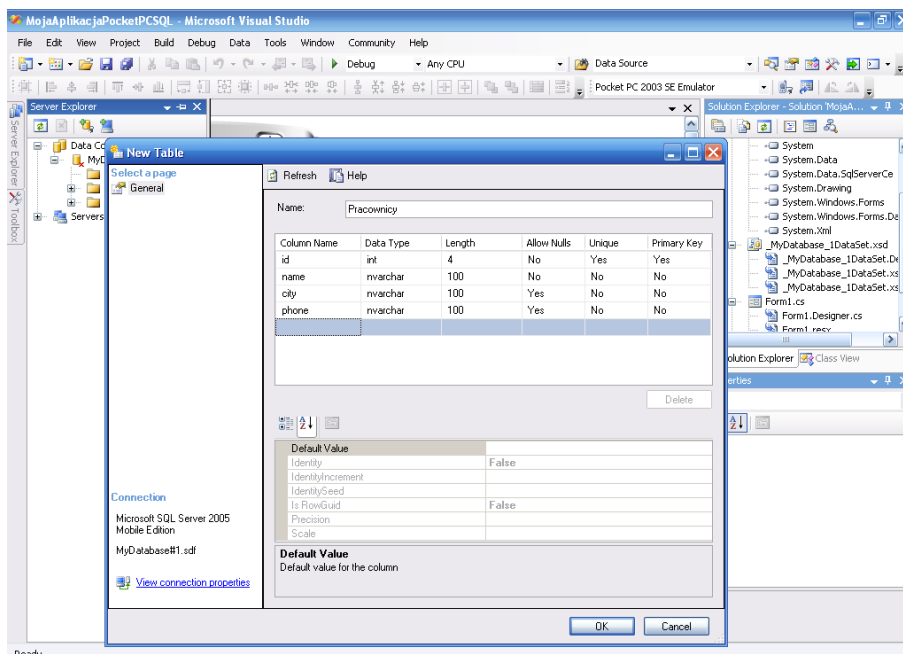
Obydwa okna zatwierdzamy poprzez polecenie **OK**.

Możemy przystąpić do zdefiniowania nowej tabeli w bazie danych. W zakładce **Server Explorer** wybieramy naszą bazę danych (*MyDatabase#1.sdf*) i na elemencie **Tables** z menu kontekstowego wybieramy **Create Table** (Rysunek 4).



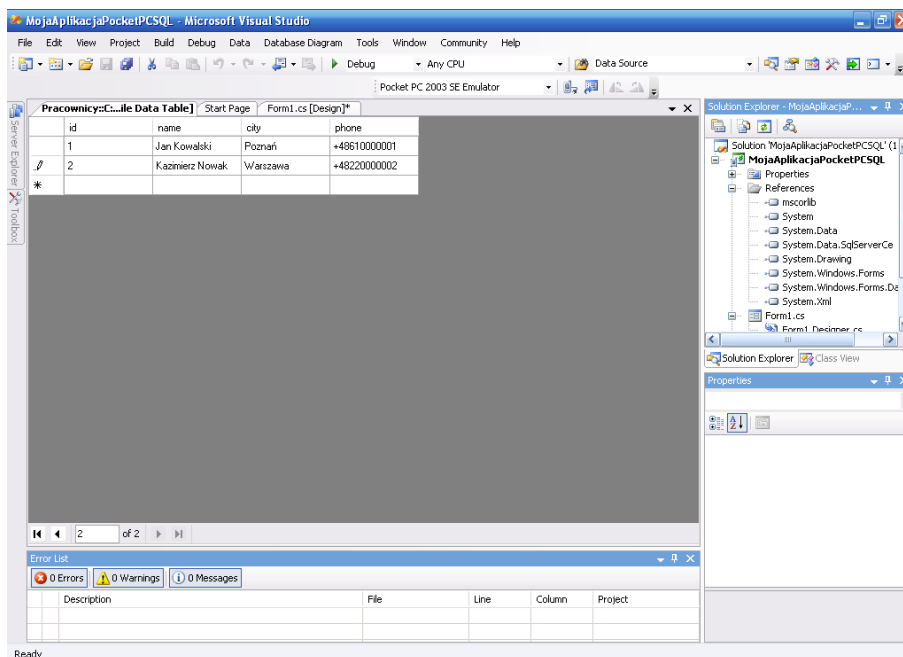
Rysunek 4 – Tworzenie nowej tabeli w bazie danych

W oknie dialogowym podajemy nazwę tworzonej tabeli i definiujemy atrybuty tabeli (dla każdego atrybutu określić możemy jego typ, długość, obowiązkowość, unikalność oraz czy stanowi część klucza podstawowego relacji) (Rysunek 5). Całość zatwierdzamy.



Rysunek 5 – Okno dialogowe tworzenia nowej tabeli w bazie danych

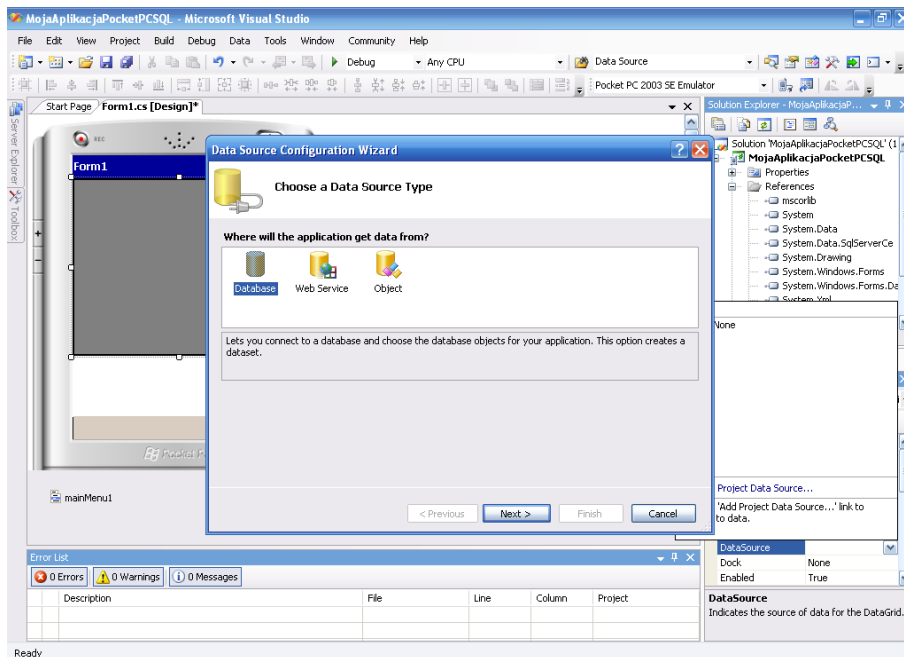
Dodanie krotek do bazy danych wykonujemy poprzez wybranie opcji **Open** z menu kontekstowego tabeli *Pracownicy* w zakładce **Server Explorer (Server Explorer ->Data Connections -> MyDatabase#1.sdf -> Tables -> Pracownicy -> Open)** (Rysunek 6).



Rysunek 6 – Dodawanie nowych krotek do bazy danych

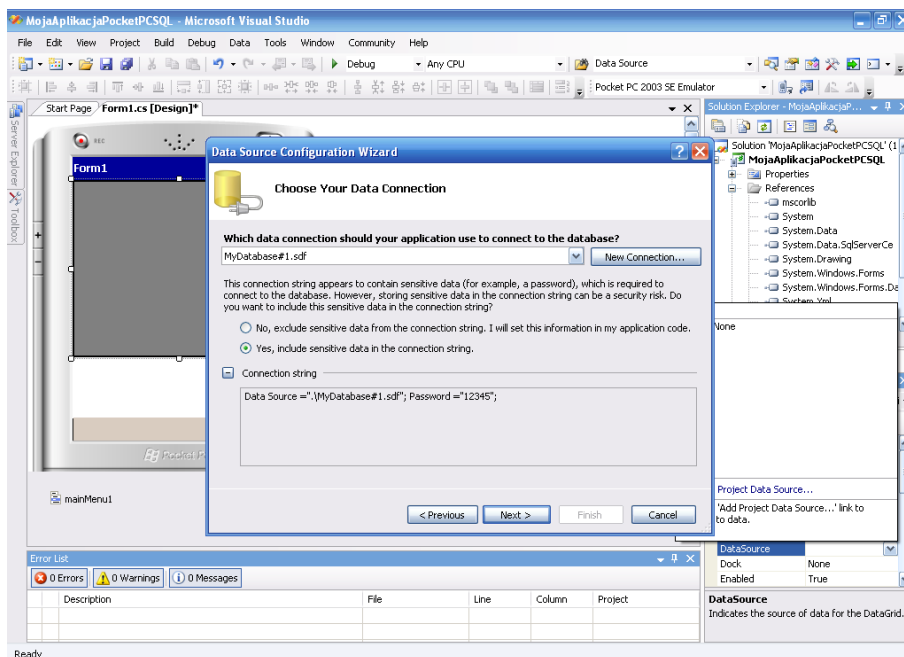
Do wyświetlenia danych z bazy danych możemy posłużyć się komponentem **DataGrid**. Umieszczamy go na formie aplikacji.

W zakładki **Properties** komponentu **DataGrid**, w opcji **Data Source** wybieramy **Add Project Data Source**. Z wyświetlonego okna dialogowego wybieramy **Database** a następnie **Next** (Rysunek 7).



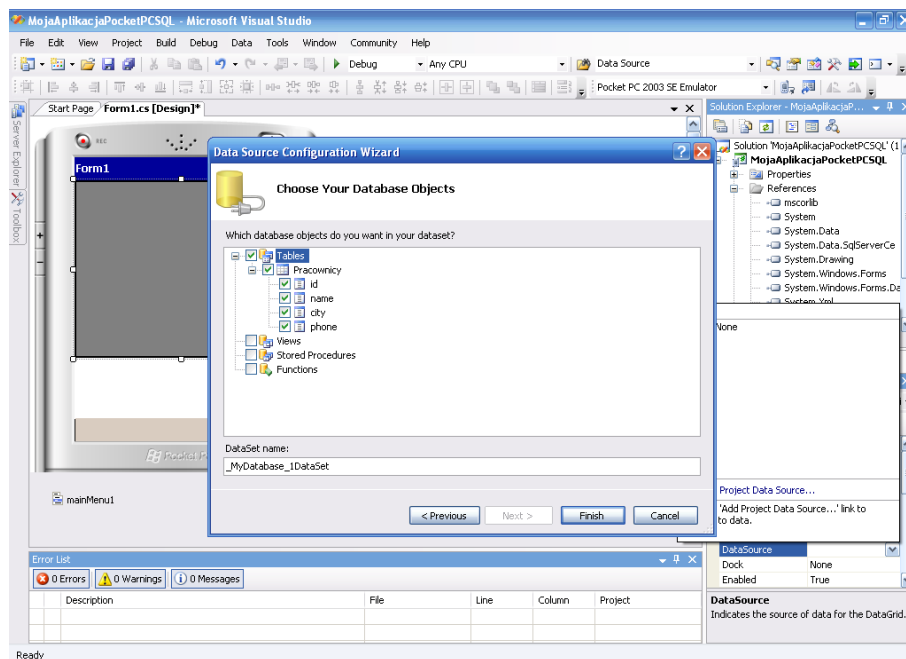
Rysunek 7 – Okno dialogowe definiowania źródła danych dla komponentu

W zależności od wymagań stawianych bezpieczeństwu aplikacji możemy zdefiniować czy hasło dostępu do bazy danych ma być przesyłane jako część adresu bazy danych (**Connection string**) (Rysunek 8).



Rysunek 8 – Określenie sposobu przesyłania hasła dostępu do bazy danych

Zaznaczamy elementy bazy danych, które mają znaleźć się w obiekcie DataSet (Rysunek 9).



Rysunek 9 – Wybór tabel do obiektu DataSet

Zatwierdzamy poprzez **Finish**.

Uruchamiamy aplikację.

3. Zadanie

Należy stworzyć prosty system magazynowy wspomagający pracę magazyniera.

Przygotować następujące dwie tabele w bazie danych o nazwie MAGAZYN:

KATEGORIE_CZESCI

#id_kategorii int – klucz podstawowy relacji

nazwa varchar(100)

CZESCI_MOTORYZACYJNE

#id_czesci int – klucz podstawowy relacji

id_kategorii int – klucz obcy do relacji KATEGORIE_CZESCI

nazwa varchar(200)

opis varchar(500)

Następnie należy utworzyć na formatce element ComboBox, w którym będą znajdować się wszystkie dostępne do wyboru Kategorie Części pobrane z tabeli KATEGORIE_CZESCI z bazy danych.

Następnie należy stworzyć obiekt DataGridView, w którym wyświetlone będą wszystkie Części Motoryzacyjne z danej Kategorii wybranej z ComboBox. Jako parametr posłuży Id Kategorii wybranej Kategorii części motoryzacyjnych.