

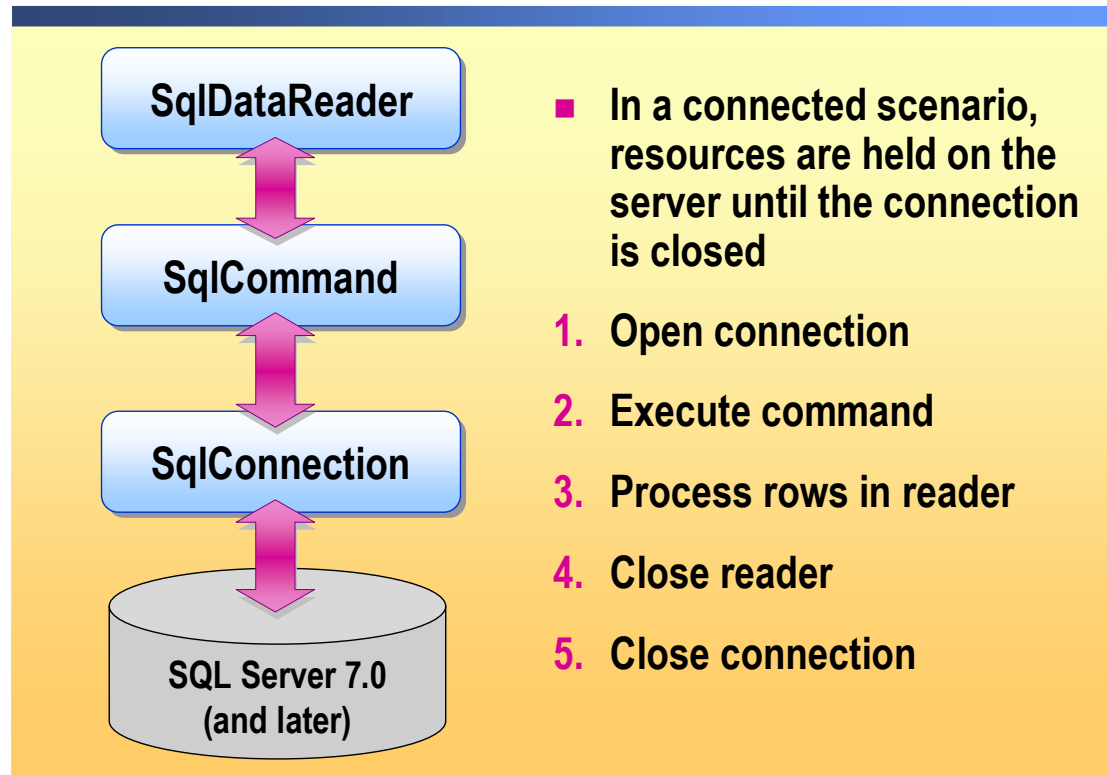
ACTIVE DATA OBJECTS

Podczas zajęć omówione zostaną dwie warstwy modułu ADO.NET

- Połączeniowa
- Bezpołączeniowa

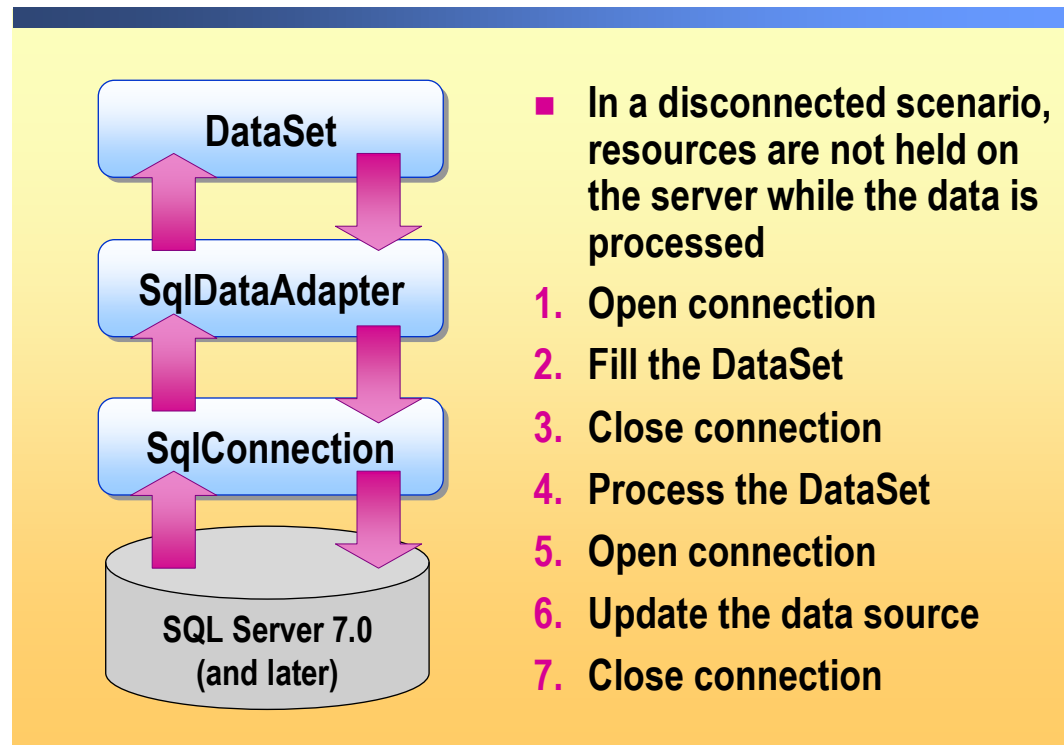
Schemat warstwy połączeniowej

Using ADO.NET Classes in a Connected Scenario

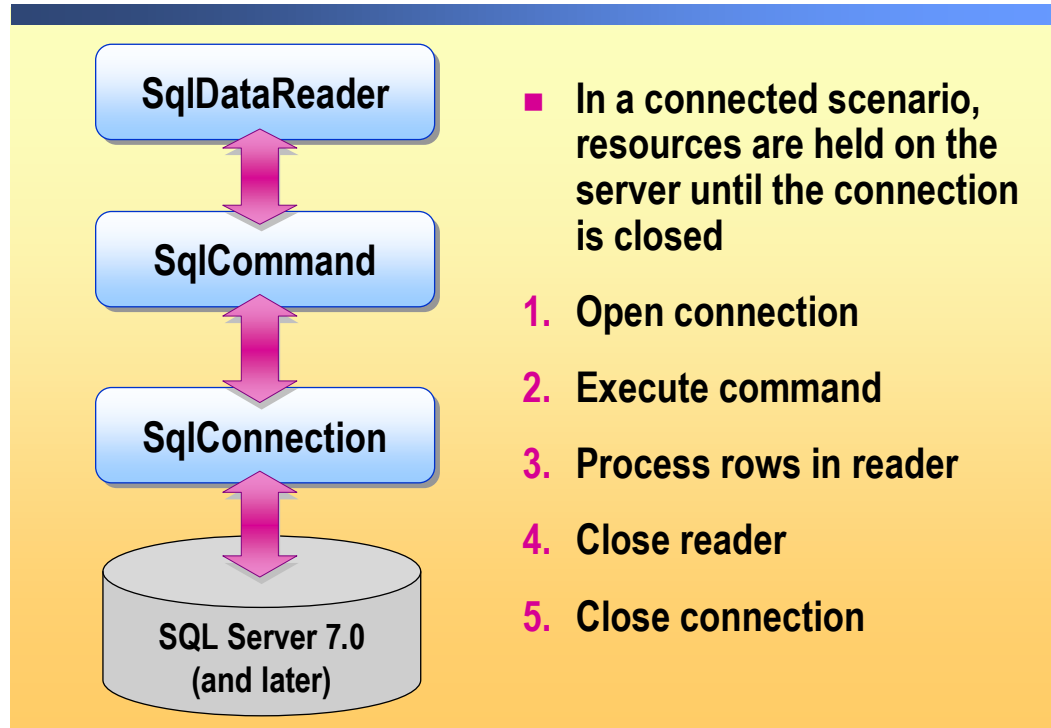


Schemat warstwy bezpołączeniowej

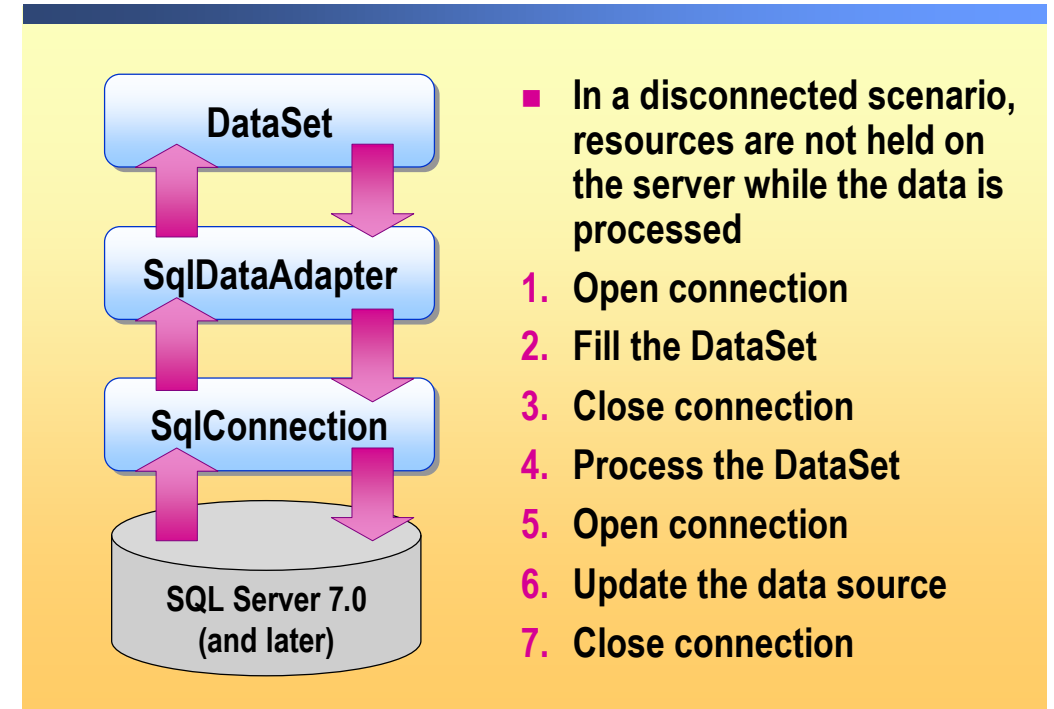
Using ADO.NET Classes in a Disconnected Scenario



Using ADO.NET Classes in a Connected Scenario



Using ADO.NET Classes in a Disconnected Scenario



Rola dostawcy danych ?

Przestrzenie nazw związane z ADO.NET

- `System.Data` – zawiera klasy warstwy bezpołączeniowej ADO.NET
- `System.Data.SqlClient` – zawiera klasy obsługujące serwer MS SQL Server
- `System.Data.OleDb` – zawiera klasy wykorzystywane w dostępie do źródeł danych OLE DB
- `Microsoft.Data.Odbc` – zawiera klasy wykorzystywane w dostępie do źródeł danych ODBC

Dostawcy danych (sterowniki):

- OLE DB (*Object Linking and Embedding, Database*) - jest rozwiązaniem własnościowym i nieprzenośnym, przeznaczonym jedynie dla oprogramowania w środowisku Microsoft Windows:
 - * SQLOLEDB – dla SQL Servera
 - * MSDAORA – dla serwera Oracle
 - * Microsoft.Jet.OLEDB dla baz danych JET.4.0 – dla baz danych JET (*Joint Engine Technology*) (np. MS Access)
- ODBC (*Open DataBase Connectivity*) – sterownik niezależny od języka programowania, systemu operacyjnego i bazy danych
- dbProvider – sterownik przeznaczony do komunikacji z systemem zarządzania MySql

Połączenia – klasa **Connection**

- OleDbConnection
- SqlConnection
- OdbcConnection

- Do nawiązania połączenia konieczna jest deklaracja obiektu klasy **Connection** oraz określenie łańcucha połączeniowego (konfiguracja połączenia) we właściwości **ConnectionString**
- Metody **Open** oraz **Close**, które służą do otwierania oraz zamykania (można zdefiniować pulę połączeń) połączenia

Klasa Command

- Jej **obiekty** służą do uruchamiania poleceń SQL – czyli np. do wyświetlania danych
- Klasa Command nie może być wykorzystana bezpośrednio – aby uruchomić polecenie musimy dysponować otwartym połączeniem reprezentowanym przez **obiekt Connection**.
- Wykorzystanie klasy Command stanowi jedyny sposób na wczytanie danych do obiektów klasy SqlDataReader

Właściwości obiektu Command:

- CommandText – łańcuch zawierający treść zapytania
- Connection – przed wykonaniem zapytania należy przypisać do niego **obiekt** Connection

Metody klasy Command:

- **ExecuteReader**

Klasa `DataReader`

- udostępnia dane tylko do odczytu (nie można za jej pomocą modyfikować danych) – można za jej pomocą dokonać szybkiego odczytu danych z bazy.
- Dzięki niej możemy utworzyć **czytnik danych** – tj. obiekt klasy `DataReader` - bez słowa `New` ponieważ klasa ta nie podlega dziedziczeniu.

- Aby do **obiektu DataReader (czytnika)** wczytać dane należy na **obiekcie klasy Command** wywołać metodę **ExecuteReader** - służy ona do uruchamiania poleceń zwracających wiersze (na przykład z SELECT)
- Co ważne czytnik w danej chwili może przechowywać tylko jeden wiersz

- Do odczytywania kolejnych wierszy używamy metody **Read** z klasy `DataReader`
- Do odczytywania kolejnych kolumn używamy właściwości **Item** z klasy `DataReader`

- Połączenie obsługujące dany czytnik nie może obsługiwać żadnych innych operacji. Dlatego lepiej nie używać zbyt często i do zbyt dużych zbiorów danych klasy DataReader ponieważ można doprowadzić do „zakleszczeń”.

Przykład

- Odczytanie danych z pliku .mdb