Linq to SQL jak se pripojit k db

<https://www.itnetwork.cz/csharp/databaze/c-sharp-tutorial-linq-to-sql-classes>

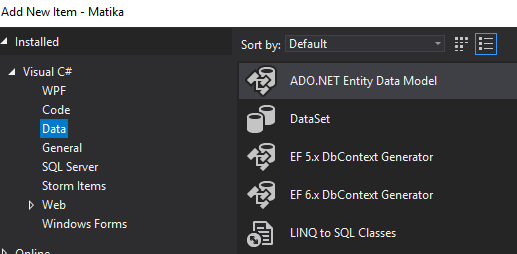
Musim nainstalovat LinqToSql ve VisualStudioInstalleru.

Databazi muzu vytvorit primo v projektu. Add -> new item-> Service based database

LinqToSqlClasses je pod Project -> Add -> new item->

Bangar Raju :

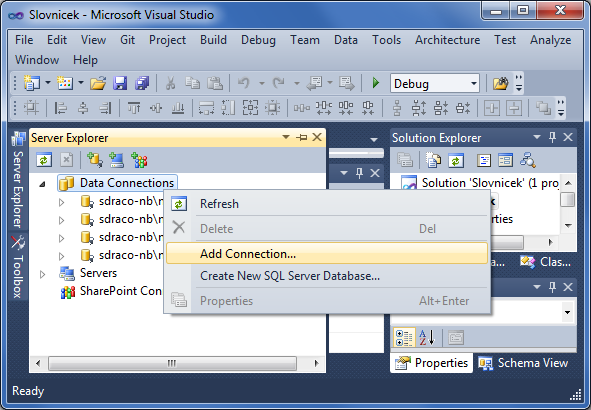
<https://www.youtube.com/watch?v=RYzIKKb-0SQ&t=1369s>



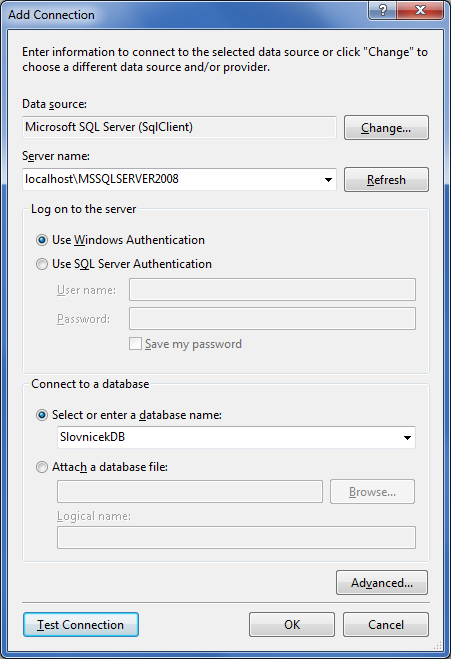
V [minulém dílu našeho seriálu tutoriálů o C# .NET](https://www.itnetwork.cz/csharp/databaze/c-sharp-tutorial-pristupy-pro-praci-s-relacni-databazi) jsme si popsali různé přístupy pro práci s databází v objektových jazycích a také technologie, které jsou v .NET frameworku pro tyto účely obsaženy. Dnes si konečně vytvoříme svou první databázovou aplikaci v C# .NET.

**Připojení k databázi z Visual Studia**

Ve Visual Studio si vytvoříme nový projekt, Windows Form Application. Pojmenujeme ji Slovnicek. Nyní otevřeme Server Explorer (ve verzích Express se okno jmenuje Database Explorer), klikneme pravým na Data Connections a zvolíme Add Connection.



V následujícím dialogu zkontrolujeme, že máme v DataSource nastaveno "Microsoft SQL Server (SqlClient)". Do Server name zadáme "localhost\JME­NO\_VASEHO\_SER­VERU" a dole vybereme naši databázi SlovnicekDB. Potvrdíme.



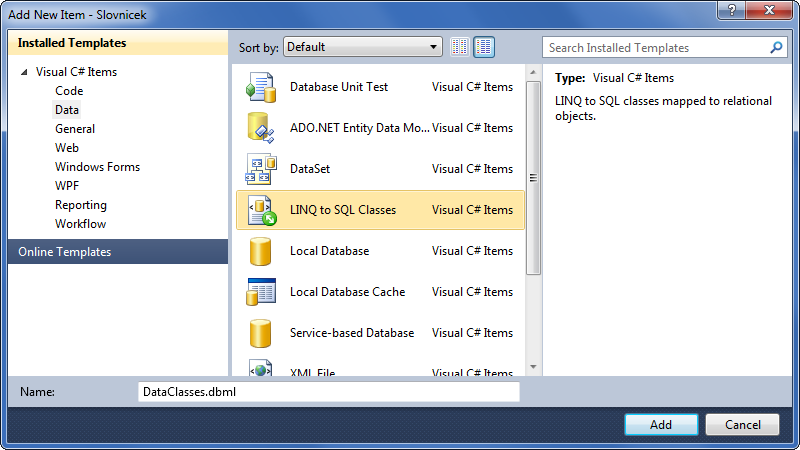
Visual Studio o naší databázi ví a dokáže se k ní připojit.

**LINQ to SQL**

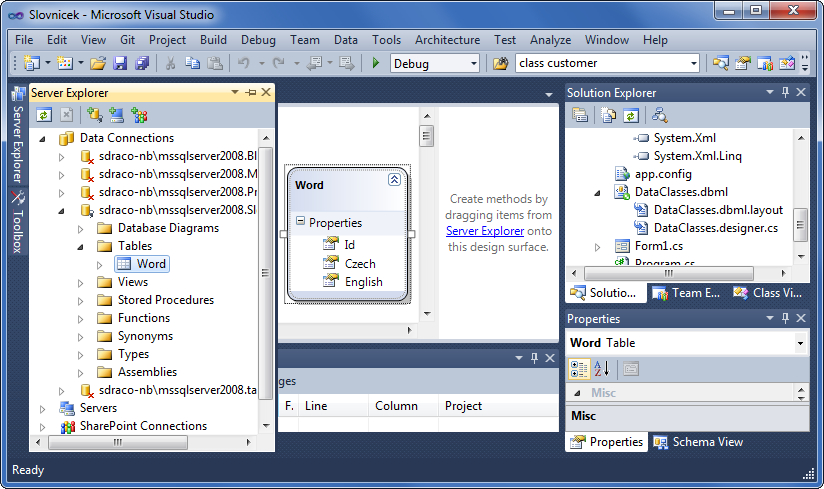
My budeme pro práci s databází používat technologii LINQ to SQL. Již podle názvu se jedná o LINQ provider, který umožňuje komunikovat s MS-SQL databází pomocí technologie LINQ. Technologii LINQ z minulých lekcí umíme používat.

LINQ to SQL je na používání poměrně jednoduché ORM, díky kterému budeme s řádky z databáze pracovat jako s objekty.

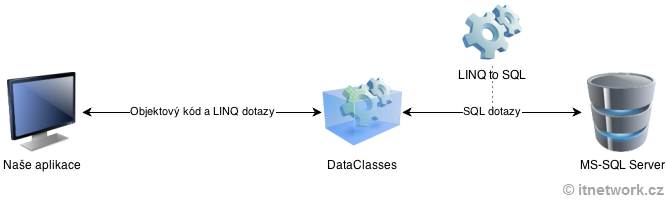
K projektu Slovnicek si připojíme nový item, LINQ to SQL Classes. Jméno klidně ponecháme na DataClasses, jen smažeme tu jedničku.



Otevře se nám prázdná plocha. K projektu jsme nyní přidali třídy, které představují jednotlivé entity v databázi. Pro každou tabulku bude v DataClasses třída, která ji reprezentuje. Třída bude mít takové vlastnosti, jako má tabulka sloupce. Dále nám umožní jednoduše používat relace, které v naší situaci s jedinou tabulkou zatím nevyužijeme. DataClasses za nás samozřejmě vygeneruje Visual Studio, přesněji nástroj DBMetal. Stačí otevřít Server Explorer, označit všechny tabulky v databázi a myší je přesunout na bílou plochu DataClasses.



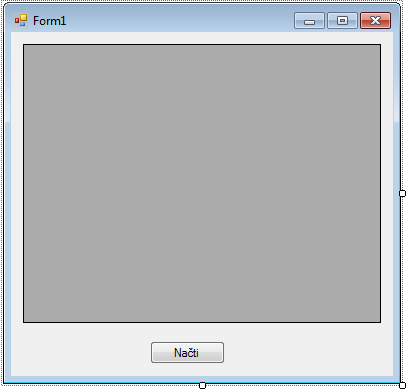
To je opravdu vše. Nyní můžeme s databází pracovat plně objektově. Pro lepší pochopení funkčnosti jsem pro vás připravil obrázek. Naše aplikace komunikuje **pouze** s DataClasses, kde jsou objekty s daty, o víc se my nestaráme. Na pozadí DataClasses komunikují s databází pomocí provideru LINQ to SQL a naše objektové dotazy jsou automaticky převáděny na čisté SQL, posílány databázi a výsledky z databáze jsou opět převáděny na objekty a vraceny pomocí DataClasses.



Možná se vám to zdá složité, ono také složité je, ale pouze uvnitř. Celá technologie je velmi dobře odladěná a jednoduše použitelná, objektový komfort při psaní aplikací je velmi příjemný, urychluje vývoj a zpřehledňuje kód.

**První databázová aplikace**

Pojďme nakonec konečně něco naprogramovat. Přejděme do Form1.cs a na jeho plochu si natahněme DataGridView, tu pojmenujeme dataGridViewSlo­vicka. Dolů umístěme tlačítko Načti, pojmenované buttonNacti.



Právě kontrolka DataGridView se dokonale hodí k zobrazení dat z databáze. Umí toho opravdu mnoho a lze ji velmi dobře přizpůsobovat.

Uvnitř formuláře si vytvořme tzv. datový kontext. To je proměnná, která referencuje na objektovou strukturu databáze. Kontext nám vygenerovalo Visual Studio a je součásti DataClasses. Přidejme si tuto proměnnou do formuláře:

**public** **partial** **class** Form1 : Form

{

DataClassesDataContext kontext = **new** DataClassesDataContext();

...

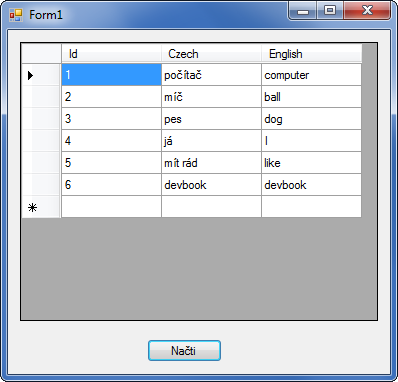
Zavoláním konstruktoru se nám aplikace k databázi připojí. Parametry připojení můžeme v konstruktoru blíže specifikovat nebo je můžeme upravit v souboru app.config, který se k našemu projektu přidal díky LINQ to SQL Classes. Zatím se tím však nebudeme zatěžovat.

Naklikněme si tlačítko buttonNacti, zde do zdroje dat dataGridView odkážeme kolekci Words z kontextu:

dataGridViewSlovicka.DataSource = kontext.Words;

Ano, Visual Studio nám při generování DataClasses vytvořilo pro každou tabulku kolekci. Kolekce slovíček (objektů typu Word) se jmenuje Words. Ve výchozím nastavení dochází k pluralizaci názvů, což je velmi intuitivní a přehledné. Funguje to samozřejmě jen v angličtině, kdybychom tabulku pojmenovali Slovicko, VS by došlo k názvu Slovickos, to není příliš pěkné. Proto budeme databázi preferovat v angličtině. Narážíme na další problém s používáním českých identifikátorů, proto to ve svých programech nikdy nedělám, nicméně jsem vám slíbil, že tutoriály budu psát česky.

Aplikaci spustíme a klikneme na tlačítko:



Vidíme, že jsme se úspěšně napojili na databázi a zobrazili si obsah tabulky Word, tedy kolekci Words v kontextu z DataClasses.