

# Nástroje na správu databáze

---

## SQL Server Management Studio

Databázový server je koncipován jako služba na pozadí. Aby se s ním dalo pracovat, tedy spravovat ho na administrátorské úrovni a také zadávat a ladit SQL příkazy, je součástí dodávky i množina nástrojů. Nejuniverzálnější z nich je *SQL Server Management Studio*. Umožňuje vytvářet nové databáze, vytvářet a spravovat účty jednotlivých uživatelů a přidělovat a rušit jejich oprávnění pro práci s jednotlivými objekty databáze.

SQL Server Management Studio je integrované prostředí na správu databázového serveru SQL Server 2012; jeho součástí je i prostředí pro zadávání a ladění SQL příkazů. SQL Server Management Studio je vybudováno na bázi vývojového prostředí Visual Studio. Nástroj se spouští standardním způsobem z operačního systému Windows.

Po spuštění se zobrazí dialog připojení k databázovému serveru. Pomocí ovládacího prvku **Server type** je možné změnit typ připojení a připojit se k analytické, reportovací nebo integrační službě. Implicitně je nastaveno připojení k databázovému serveru.

Po stisknutí tlačítka **Options** se zpřístupní rozšířený mód přihlašovacího dialogu s kartami:

- Login
- Connection Properties
- Connection Parameters

Na kartě **Connection Properties**, která je určena pokročilým administrátorům a speciálním režimům práce s databázovým serverem, můžete nastavit databázi, ke které se chcete připojit, typ síťového protokolu, časový limit pro připojení a časový limit pro vykonání příkazu.

Pracovní plocha SQL Management Studia je rozdělena na několik částí. Levé podokno **Object Explorer** poskytuje grafic-

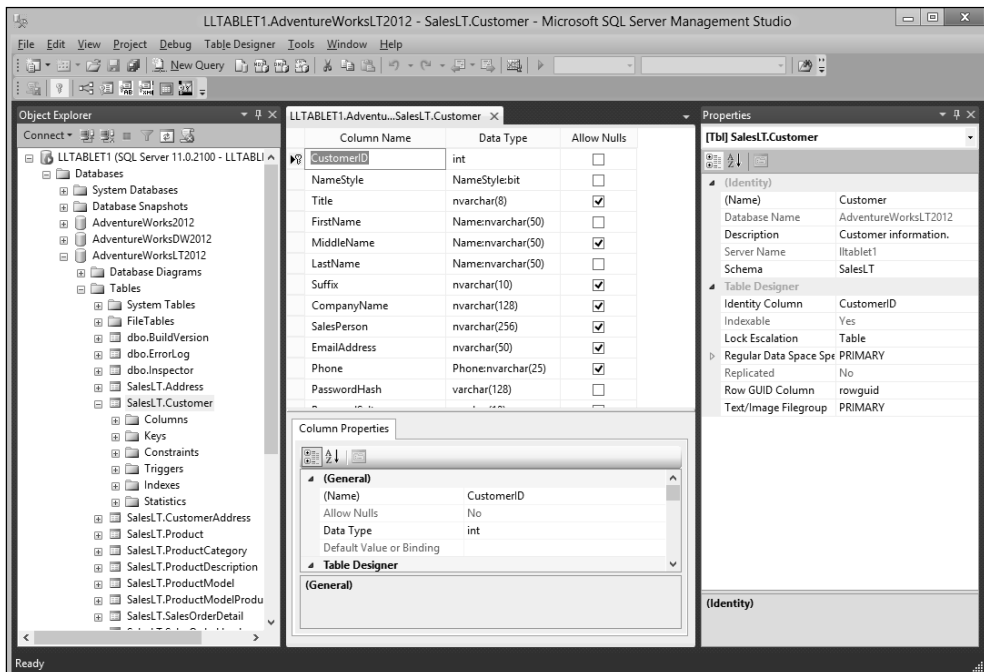
### Témata kapitoly:

- SQL Server Management Studio
- Nastavení parametrů databázového serveru
- Konzolová aplikace SQLCMD
- Přístup k SQL Serveru prostřednictvím PowerShellu
- SQL Server Configuration Manager
- SQL Server Data Tools (SSDT)

ký, přehledný a hierarchicky uspořádaný pohled na objekty v databázi, například v případě databázových tabulek až na úroveň sloupců a indexů. Objekty jsou asociované se schématem. V případě potřeby je možné do levé části pracovní obrazovky zobrazit i podokno **Registered Servers**, které zobrazuje seznam zaregistrovaných serverů, k nimž se můžete pomocí Management Studio připojit. Pomocí tohoto podokna se mohou administrátoři, kteří spravují více serverů, přepínat mezi jednotlivými instancemi.



**Obrázek 4.1:** Dialog nástroje SQL Server Management Studio pro připojení k databázovému serveru



**Obrázek 4.2:** SQL Server Management Studio

Hlavní podokno uprostřed pracovní plochy se používá na vizuální návrh databázových objektů nebo na zadávání příkazů Transact-SQL, XMLA, MDX či DMX. Výsledky je možné zobrazovat v textové nebo tabulkové formě. Pravé podokno **Properties** je určeno k zobrazení parametrů vybraného objektu.

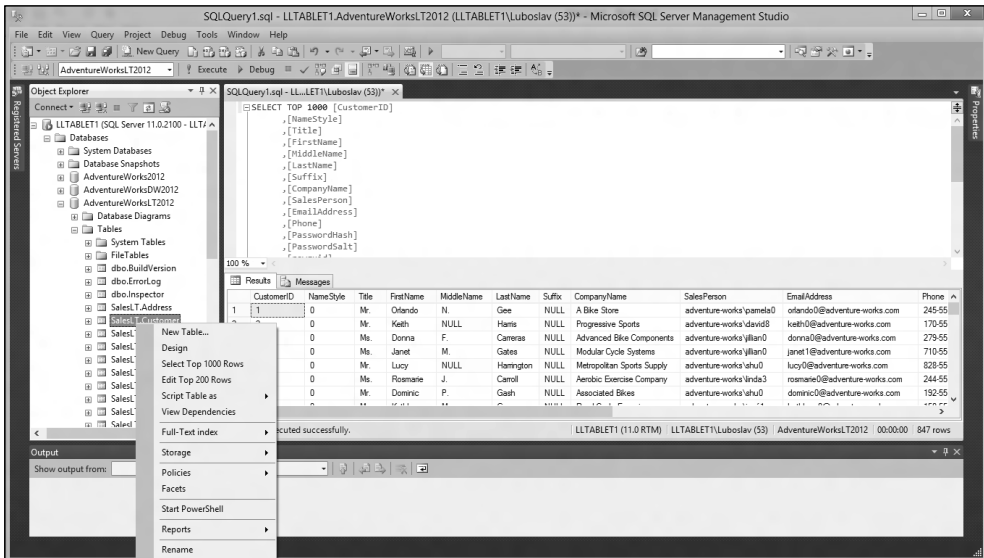
## POZNÁMKA

Hierarchická stromová struktura levého podokna nástroje SQL Server Management Studio bude mít zřejmě vliv na vaše vnímání uspořádání objektů v databázi, hlavně pokud jste začátečníci. Aplikace tohoto typu totiž představuje jakési „oči“, prostřednictvím kterých nahlížíte do hlubin struktur databáze.

## Spouštění a ladění SQL příkazů

SQL příkaz vložený do horní části pracovního podokna je možné spustit pomocí tlačítka **Execute**, případně v režimu ladění pomocí tlačítka se symbolem zelené šipky.

Výsledky dotazování je možné zobrazit ve formě tabulky nebo textového výpisu, případně uložit do souboru. V místní nabídce databázové tabulky je připravena funkce, která vygeneruje top 1000 záznamů z příslušné tabulky. Připomínáme, že v této funkci není definované kritérium řazení, takže se de facto vypíše 1 000 záznamů v náhodném pořadí. Nejčastěji to bude v pořadí, jak byly záznamy do databáze vloženy, ale ani to není zaručené.



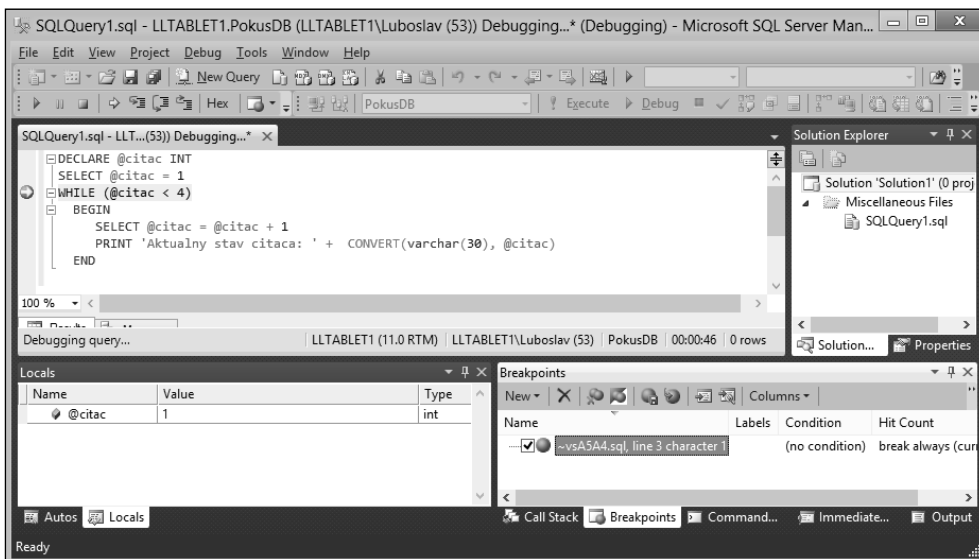
**Obrázek 4.3:** SQL Server Management Studio v režimu zadávání SQL příkazů

Důležitou součástí každého víceřádkového kódu jsou komentáře. V jazyku SQL začíná komentář dvěma pomlčkami (--).

### TIP

V SQL Server Management Studiu můžete okomentovat vybraný blok kódu pomocí ikony na panelu nástrojů (na obrázku pátá zprava) nebo v nabídce **Edit** → **Advanced** → **Comment Selection**.

Podobně jako do Visual Studia je i do nástroje SQL Server Management Studio implementované ladění kódu. Klepnutím na šedý svislý pás vlevo vedle příslušného řádku kódu se vytvoří zarážka (breakpoint). Zarážka se zobrazí jako kulatý červený terčík. Ladění se použít na panelu nástrojů ikonou ve tvaru zelené šipky, která je umístěná vpravo vedle tlačítka **Execute**. Tlačítka pro krokování kódu se zobrazí v ladicím módu na panelu nástrojů vpravo. Během ladění je možné v příslušných podoknech sledovat hodnoty proměnných.



**Obrázek 4.4:** Ladění kódu v prostředí SQL Server Management Studia

## Spouštění a ladění uložených procedur

V podokně pro zadávání příkazů v jazyku SQL je možné spouštět a testovat nejen SQL příkazy, ale i psát a testovat kód funkčních bloků jazyka T-SQL (Transact SQL), například funkcí, spouští uložených procedur a podobně. Ke spouštění takovýchto bloků slouží příkaz EXEC. Jeho použití si můžete vyzkoušet například na systémové uložené proceduře sp\_who, která vypíše informace o uživateli a procesech:

```
EXEC sp_who;
```

Uloženou proceduru je možné volat i s parametry. Syntaxe parametrů procedury sp\_who je:

```
sp_who [[@login_name =] 'login']
```

Tedy se vypíše jen údaje pro konkrétního uživatele v SQL Server Management Studiu. Nebo můžete v konzolové aplikaci SQLCMD tuto proceduru zavolat s parametrem, například pro login TABLETW8\LL ve tvaru

```
EXEC sp_who 'TABLETW8\LL';
```

## Automatické generování SQL příkazů pro vybranou tabulku

Kromě funkce Intellisense, která dokáže významným způsobem napovídat části zdrojového kódu při jeho psaní, hlavně názvy objektů a metod, obsahuje Management Studio i užitečnou pomůcku pro vytváření šablon SQL příkazů pro jednotlivé objekty. V podokně **Object Explorer** vyberte databázi a v ní databázovou tabulku, pro kterou chcete vytvářet šablonu příkazu. V místní nabídce zvolte položku **Script Table As** a vyberte, jaký typ šablony chcete vygenerovat. K dispozici jsou volby: **CREATE TO**, **DROP TO**, **SELECT TO**, **INSERT TO**, **UPDATE TO** a **DELETE TO**. Například pro tabulku SalesLT.CustomerAdres z cvičné databáze AdventureWorksLT2012 se vygeneruje šablona příkazu INSERT ve tvaru:

```
USE [AdventureWorksLT2012]
GO

INSERT INTO [SalesLT].[CustomerAddress]
    ([CustomerID]
    ,[AddressID]
    ,[AddressType]
    ,[rowguid]
    ,[ModifiedDate])
VALUES
    (<CustomerID, int,>
    ,<AddressID, int,>
    ,<AddressType, Name,>
    ,<rowguid, uniqueidentifier,>
    ,<ModifiedDate, datetime,>)
GO
```

## Šablony příkazů

Při tvorbě SQL příkazů vám mohou pomoci i šablony. V nabídce **View** klepněte na položku **Template Browser**. Zobrazí podokno se seznamem objektů, přičemž pro každý objekt je k dispozici několik šablon.

Například k vytvoření nové databáze je k dispozici šablona příkazu ve tvaru:

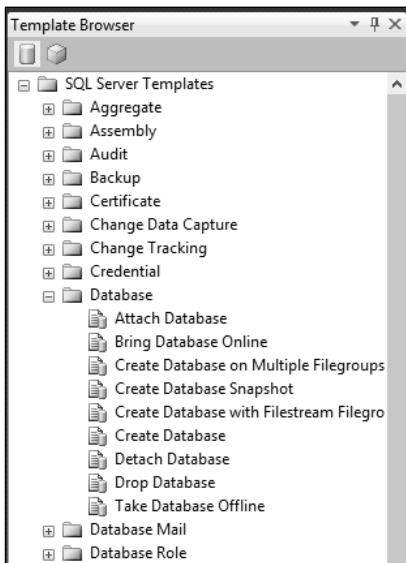
```
-- =====
-- Create database template
-- =====
USE master
GO
```

```

-- Drop the database if it already exists
IF EXISTS (
    SELECT name
      FROM sys.databases
     WHERE name = N'<Database_Name, sysname, Database_Name>'
)
DROP DATABASE <Database_Name, sysname, Database_Name>
GO

CREATE DATABASE <Database_Name, sysname, Database_Name>
GO

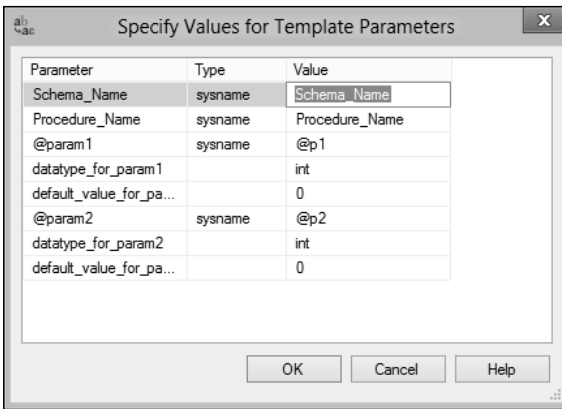
```



**Obrázek 4.5:** Nabídka šablon v podokně Template Browser

Template Browser můžete rozšiřovat i o vlastní šablony. Vytvořte například novou šablonu pro uloženou proceduru. V místní nabídce složky **Stored Procedure** pomocí položky **New** → **Folder** vytvořte novou složku pro svoje šablony a v ní pomocí místní nabídky **New** → **Template** vytvořte kostru šablony.

Před uložením šablony je výhodné ji vyzkoušet, například tak, že ji spustíte doplněnou o konkrétní hodnoty. Případně šablonu nejprve uložte, následně ji obvyklým postupem (tedy poklepnutím myši) vložte do podokna pro zadávání SQL příkazů a v nabídce **Query** vyberte položku **Specify Values for Template Parameters** a zadejte parametry.



**Obrázek 4.6:** Dialog pro zadávání parametrů do šablony

## Databázové diagramy

SQL Server Management Studio poskytuje prostředky i pro modelování a interaktivní návrh databázových struktur. Pokud si chcete databázové diagramy vyzkoušet prakticky, můžete tak učinit v cvičné databázi AdventureWorks2012 nebo v její odlehčené verzi AdventureWorksLT2012.

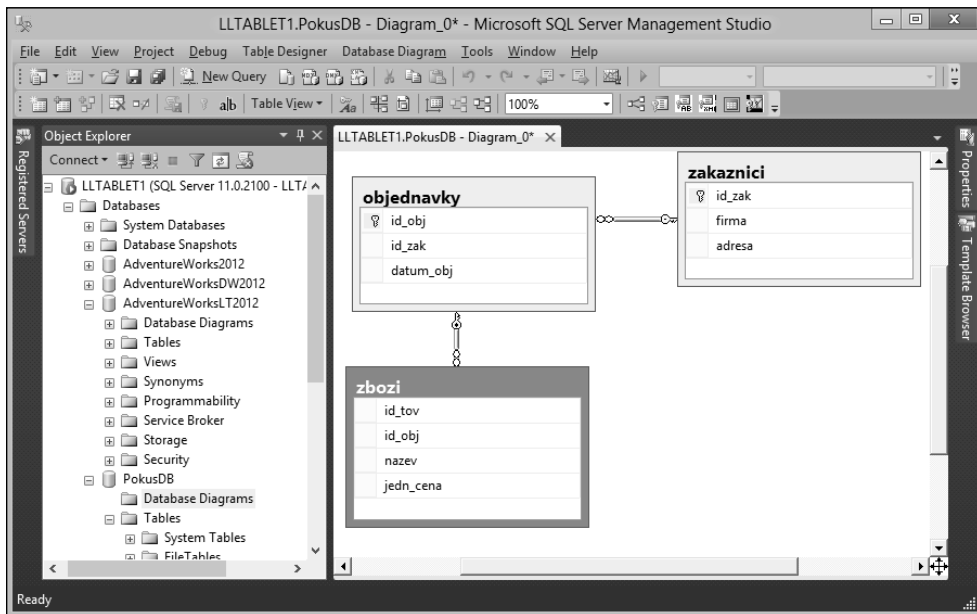
Každá databáze má v podokně **Object Explorer** složku **Database Diagrams**, na které je možné interaktivně vytvářet diagramy z vybraných tabulek. Abyste neměli problém zorientovat se v cvičných databázích s desítkami tabulek, nabízíme skript na vytvoření tří databázových tabulek propojených pomocí cizích klíčů:

```
CREATE TABLE zakaznici
(
    id_zak int PRIMARY KEY,
    firma varchar(20) NOT NULL,
    adresa varchar(30)
);

CREATE TABLE objednavky
(
    id_obj int PRIMARY KEY,
    id_zak int NOT NULL,
    datum_obj datetime
    FOREIGN KEY (id_zak) REFERENCES zakaznici(id_zak)
);

CREATE TABLE zbozi
(
    id_tov int NOT NULL,
    id_obj int NOT NULL,
    nazev varchar(30),
    jedn_cena money
```

```
FOREIGN KEY (id_obj) REFERENCES objednavky(id_obj)
);
```



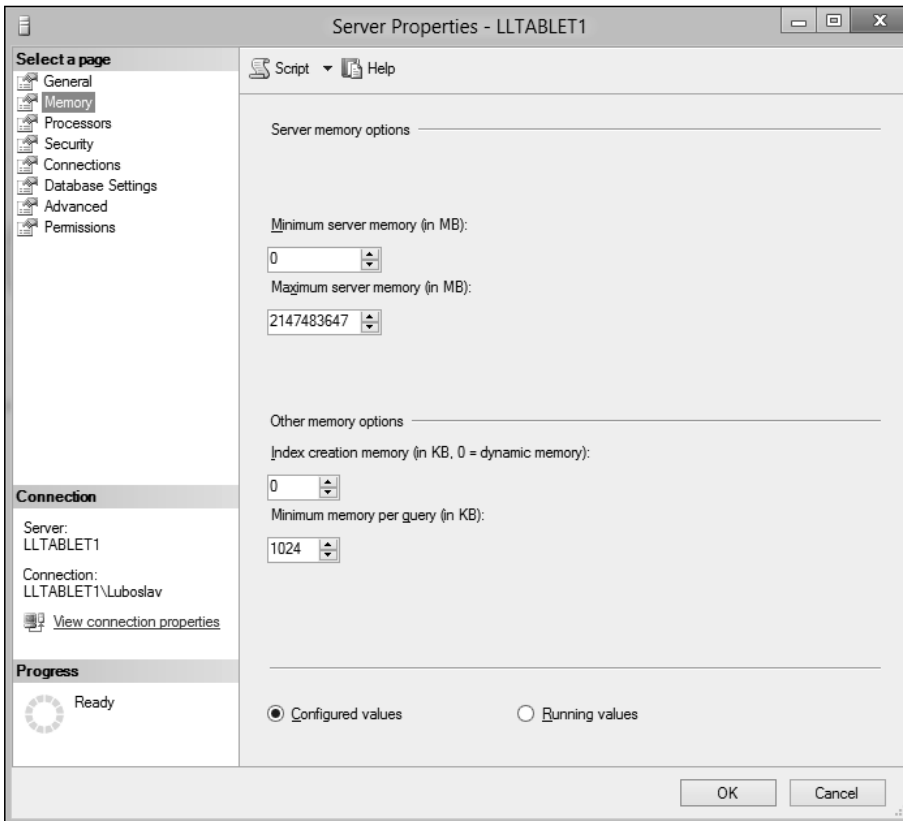
Obrázek 4.7: Databázové diagramy v prostředí SQL Server Management Studia

## Nastavení parametrů databázového serveru

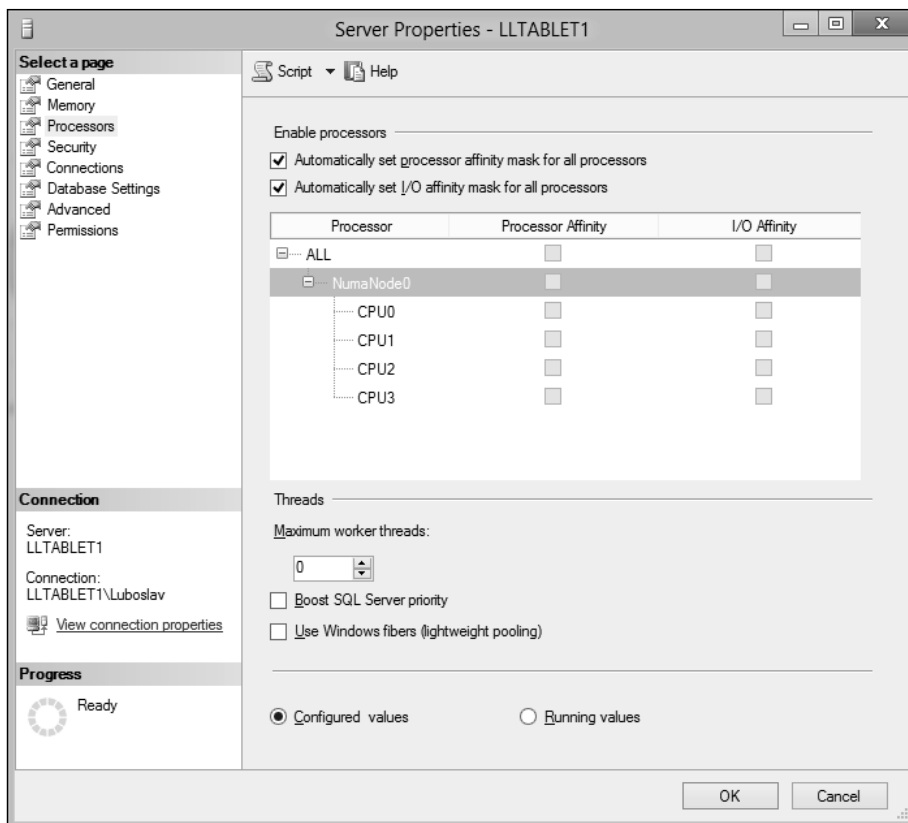
V příkazovém režimu se k nastavení základních parametrů používá uložená procedura `sp_configure`. Řadu parametrů je možné nastavit i v interaktivním módu pomocí nástroje SQL Server Management Studio. Z místní nabídky příslušné instance databázového serveru vyberte položku **Properties**. Na kartě **Memory** můžete nastavit limity, kolik paměti bude mít databázový server k dispozici. Je to důležité hlavně v případě, kdy na stejném fyzickém serveru běží i middlewarové aplikace.



Nastavuje se minimální a maximální množství paměti, které instance může použít ve svém prostředí. Nastavení se projeví až po restartu. Všimněte si přepínače ve spodní části. **Running values** ukazuje aktuální stav, hodnoty, které SQL Server v současné době používá. **Configured values** zobrazí předkonfigurované hodnoty, které SQL Server bude používat po dalším restartu. Na kartě **Processors** můžete nastavit, jak bude databázový server využívat jednotlivá procesorová jádra. Toto nastavení je užitečné, pokud máte na víceprocesorovém serveru víc než jednu instanci SQL Serveru. Více zatěžované instance můžete přidělit více procesorové kapacity. Můžete nastavit, aby operační systém Windows přiřadil procesům SQL Serveru vyšší prioritu. Nastavení je třeba dobře uvážit a otestovat, aby některé jiné důležité aplikace netrpěly nedostatkem procesorové kapacity.

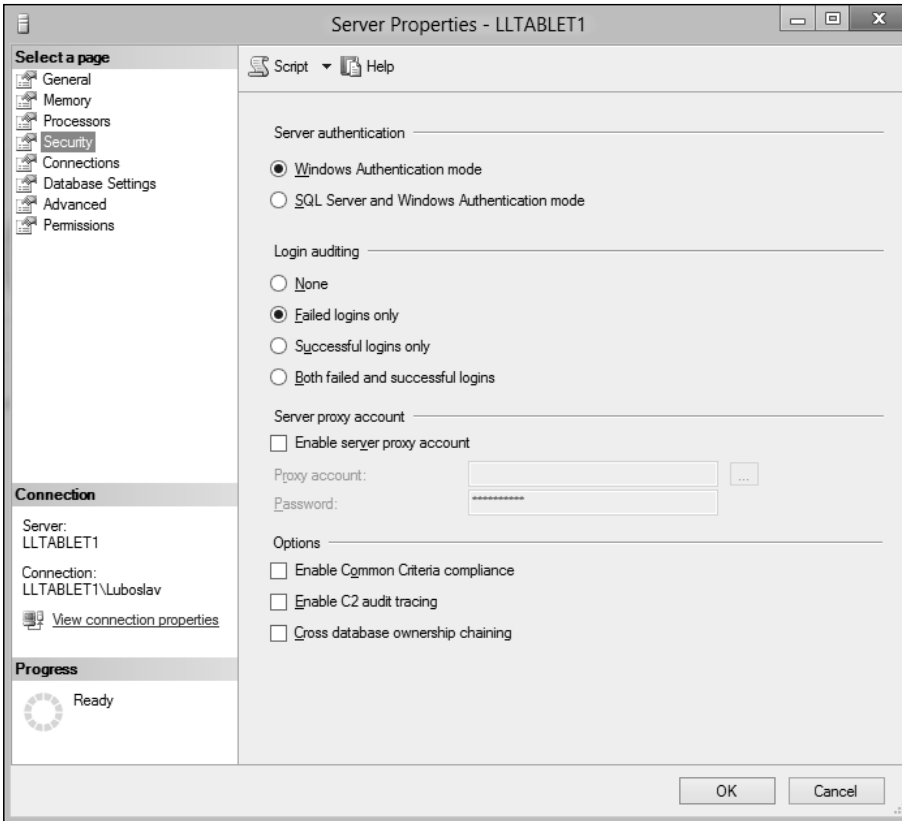


**Obrázek 4.8:** Nastavení limitů paměti pro databázový server



**Obrázek 4.9:** Nastavení využívání procesorových jader

Na kartě **Security** můžete nastavit typ autentizace a způsob auditování přihlášení. Implicitně je nastavena volba ověřování přihlášení, která se nezdařila.



Obrázek 4.10: Nastavení autentizace

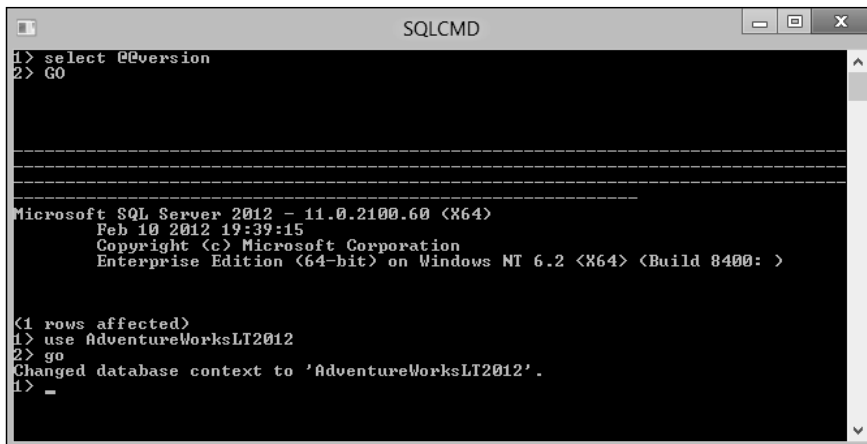
## Konzolová aplikace SQLCMD

Ke správě databázového serveru a ladění databázové části aplikací je možné využít i jednoduchou interaktivní textovou konzolovou aplikaci SQLCMD. Slouží k zadávání příkazů jazyka SQL databázovému serveru a zobrazování výstupů vygenerovaných databázovým serverem, například výpisů údajů z databázových tabulek, potvrzení vykonání příkazů, chybových hlášení a podobně. Spouští se například pomocí funkce Spustit (Run) operačního systému Windows.

Na připojení k databázovému serveru využívá tato aplikace SQL Native Client. Ke konkrétní instanci databázového serveru se dá připojit příkazem:

```
SQLCMD -S Server
```

Při realizaci některých administrátorských úkonů vyžadujících prioritu zdrojů je užitečné připojit se přes dedikované spojení administrátora s parametrem SQLCMD -A. Konzola umožňuje zadávat jako parametry i názvy souborů obsahujících SQL kód. Nápovědu obsahující i parametry pro spuštění aplikace je možné zobrazit pomocí parametru SQLCMD -?.



**Obrázek 4.11:** Konzolová aplikace SQLCMD

Instanci, ke které je konzolová aplikace aktuálně připojena, je možné identifikovat příkazem:

```
SELECT @@version
GO
```

Pokud není určeno jinak, konzola se po spuštění připojí k databázi Master. Na přepnutí k jiné databázi použijte příkaz `use <Název databáze>`.

## Přístup k SQL Serveru prostřednictvím PowerShellu

Prostředí Windows PowerShell používá správcovské úlohy nazývané *cmdlet*. Každá úloha *cmdlet* má požadované i volitelné argumenty nazývané parametry. Tyto argumenty identifikují, které objekty budou zpracovávány, nebo určují způsob, jakým argumenty *cmdlet* tuto úlohu vykonávají. Úlohy *cmdlet* můžete ve skriptech kombinovat a dosáhnout tak realizace komplexnějších funkcí.

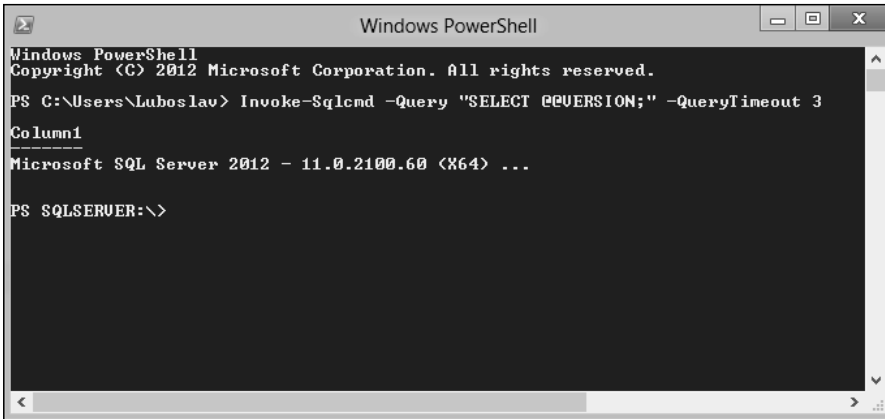
PowerShell můžete spustit z operačního systému Windows nebo prostřednictvím nástroje SQL Server Management Studio. V podokně **Object Explorer** vyberte instanci databázového serveru a v místní nabídce zvolte položku **Start PowerShell**.

Přístup k SQL Serveru se realizuje pomocí *cmdletu* `Invoke-Sqlcmd`. Začneme nejjednodušším příkladem. Po spuštění konzoly PowerShell můžete zjistit verzi SQL Serveru, ke které se připojíte, příkazem:

```
Invoke-Sqlcmd -Query "SELECT @@VERSION;" -QueryTimeout 3
```

Pokud chcete vypsát seznam databází pod správou SQL Serveru, můžete použít příkaz:

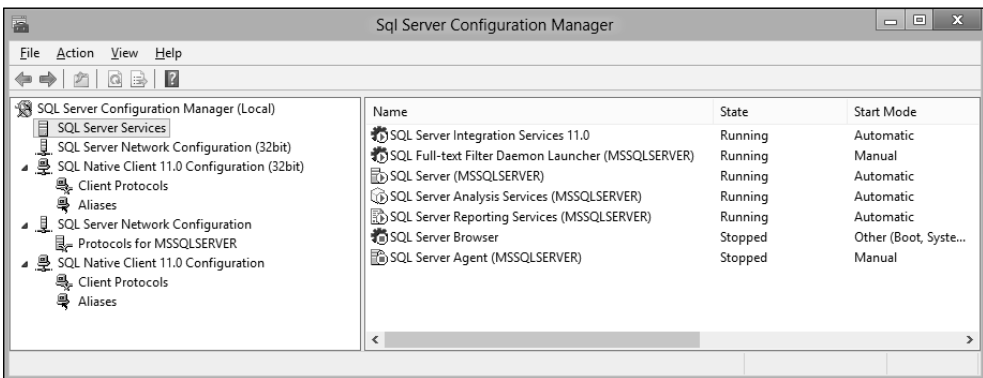
```
Invoke-Sqlcmd -Query "Cd databases foreach ($database in (Get-ChildItem))
{Write-Host $database.Name}" -QueryTimeout 3
```



Obrázek 4.12: Aplikace PowerShell

## SQL Server Configuration Manager

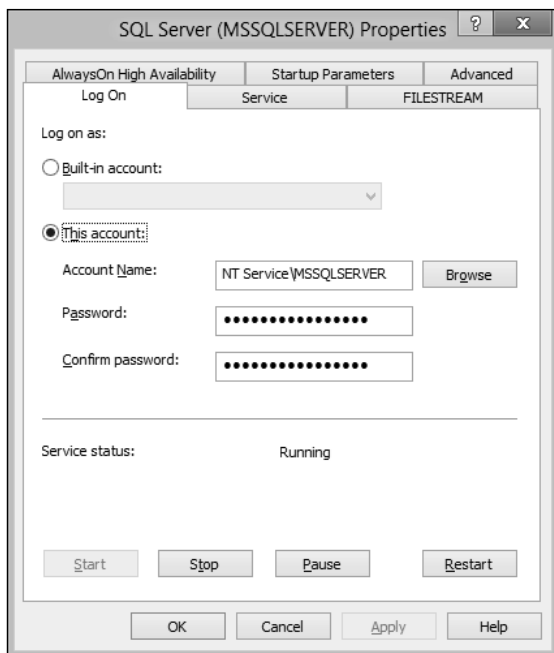
Nástroj SQL Server Configuration Manager je určen pro základní konfigurační úkony týkající se služeb, které tvoří platformu SQL Server 2012, včetně spouštění, zastavování a restartu těchto služeb a nastavování serverových a klientských protokolů. Implicitní instalaci, včetně integračních, analytických a reportovacích služeb, tvoří položky na obrázku 4.13 (za každou službou je vypsaný implicitně nastavený status spouštění služby).



Obrázek 4.13: SQL Server Configuration Manager

Pro každou službu je možné pomocí místní nabídky nastavit mód spouštění. Služba může být spouštěna automaticky při startu operačního systému serveru, případně vývojářského počítače (volba **Automatic**), ručně (volba **Manual**) nebo je možné spouštění služby zakázat (volba **Disabled**).

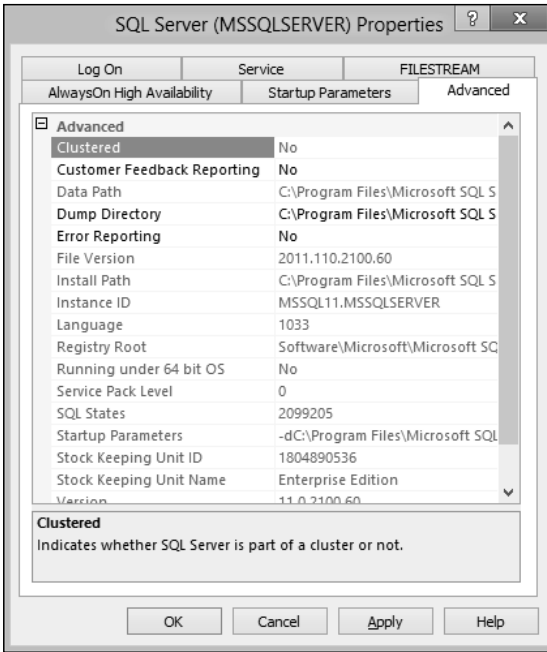
Přihlašovací parametry a účty pro jednotlivé služby se nastavují na kartě **Log On**. Na této kartě je taktéž možné každou službu spustit, zastavit, případně restartovat.



**Obrázek 4.14:** Vlastnosti databázového serveru – karta Log On

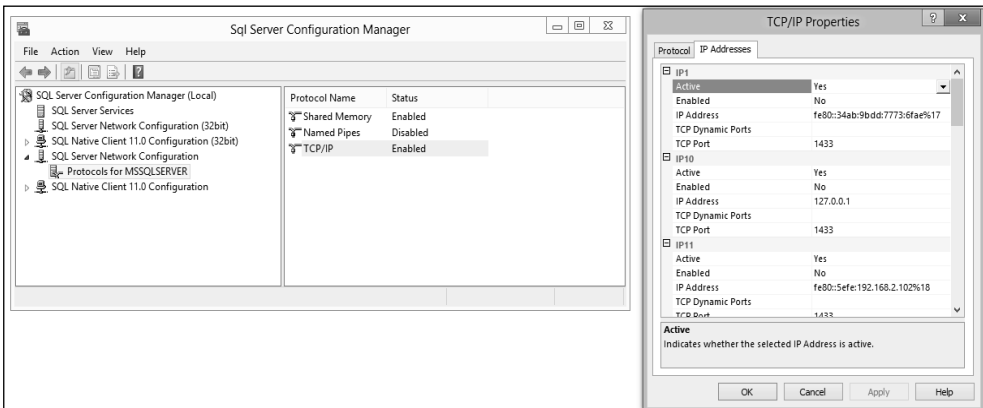
Na kartě **Advanced** je možné prohlížet a nastavovat různé parametry pro jednotlivé služby:

- **Clustered** – parametr indikuje, zda je služba nainstalovaná na clusterovém serveru.
- **Customer Feedback Reporting** – indikuje zapnutí nebo vypnutí služby Service Quality Monitoring pro monitorování provozu služby.
- **Data Path** – adresář, ve kterém jsou datové soubory.
- **Dump Directory** – adresář, kam se ukládá výpis obsahu paměti v případě výskytu chyby.
- **Error Reporting** – pokud je tento parametr nastavený na hodnotu **Yes**, program Dr. Watson, který je součástí operačního systému, posílá informace o chybě. Tento parametr je možné nastavit pomocí nástroje SQL Server Management Studio v podokně **Object Explorer** přes místní nabídku **Properties** → **Misc. Server Settings**.
- **File Version** – verze spustitelných souborů SQL Serveru.
- **Install Path** – adresář, ve kterém jsou binární soubory tvořící SQL Server.
- **Instance ID** – parametr indikuje instanci SQL Serveru, jejíž součástí je příslušná služba.
- **Language** – jazyk pro generování zpráv serveru.
- **Registry Root** – kořenový prvek příslušné aplikace v registrech operačního systému.
- **Service Pack Level** – číslo aplikovaného opravného balíčku.
- **Startup Parameters** – seznam parametrů pro spuštění služby.
- **Version** – verze instance SQL Serveru.
- **Virtual Server Name** – název virtuálního serveru, pokud je SQL server nainstalován na clusterovém serveru.



**Obrázek 4.15:** Vlastnosti databázového serveru – karta Advanced

Nástroj SQL Server Configuration manager slouží i k povolení a nastavení serverových a klientských protokolů.



**Obrázek 4.16:** Konfigurace protokolů a portů