# MS SQL Server 2008 Management Studio – Tutoriál

### Vytvoření databáze

Při otevření management studia a připojením se ke konkrétnímu sql serveru mám v levé části panel s názvem "Object Explorer". V tomto panelu vidíme náš sql server, ke kterému jsme se připojili a jednotlivé složky v něm (Databases, Security apod.). Nás v tuto chvíli bude zajímat pouze složka "Databases", ve které budeme dělat základní práce s databází. Nejdříve si musíme nějakou databázi vytvořit. Zde je postup:

1. Pravým tlačítkem myši klikneme na složku "Databases" a zvolíme volbu "New Database…" (viz. Obrázek 1)



Obrázek 1

 Objeví se nám dialogové okno (viz. Obrázek 2), ve kterém zadáváme základní informace o naší nové databázi. Zadáme název databáze k položce "Database name" a můžeme vyplnit další informace, jako např. jaká bude kapacita naší databáze apod. implicitně je to 3MB.

📕 New Database					
Select a page Page General	🔄 Script 🝷 💽 Help				
😭 Options 🚰 Filegroups	Database <u>n</u> ame:		moje_nova_datab	aze	
	<u>0</u> wner:		<default></default>		
	🔽 Use full-text indexing				
	Database files:				
	Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
	moje_nova_databaze	Rows	PRIMARY	3	By 1 MB, unrestricted growth
	moje_nova_databa	Log	Not Applicable	1	By 10 percent, unrestricted gr
Connection					
Server: HOME-NOTEBOOK					
Connection:					
HOME-NOTEBOOK\Admin					
View connection properties					
Progress					
C Ready	<			∆dd	<u>R</u> emove
					OK Cancel

Obrázek 2

3. Po stisku tlačítka "OK" můžeme naší novou databázi vidět v levém panelu pod složkou "Databases" (viz. Obrázek 3)



#### Vytváření tabulek v databázi

Nyní máme vytvořenou databázi, ale nemáme v ní žádné tabulky pro práci s daty. V této části si tedy ukážem jak vytvářet tabulky, jak do tabulek přidávat atributy a taky jak nastavovat atributům konkrétní vlastnosti. V levém panelu rozevřeme naší databázi a uvidíme další podsložky, které už se týkají naši konkrétní databáze. Zde se budeme zabývat podsložkou "Tables".

1. Pravým tlačítkem myši klikneme na složku "Tables" a zvolíme volbu "New Table…"



Obrázek 4

2. Po vytvoření tabulky se v hlavní části management studia objeví tabulka, kterou chceme vytvářet. Základním krokem bude si naši tabulku pojmenovat. V pravé části management studia se obvykle nachází panel s vlastnostmi konkrétního prvku (Properties). V položce "Name" dáme naší tabulce jméno (viz. Obrázek 5). Pojmenovat tabulku můžeme taky při ukládání záložek, které máme aktuálně otevřené, management studio se zeptá, jak se tabulka má jmenovat.



Obrázek 5

 Nyní můžeme do tabulky vkládat atributy. Jednotlivým atributům přiřazujeme název, datový typ, a jestli hodnota atributu může být null, implicitně je nastaveno povolení vkládání hodnot null (viz. Obrázek 6). K dalším vlastnostem atributů se vrátíme později.

Column Name	Data Type		Allow Nulls
id_tabulka_1	int		
atribut_1	nvarchar(50)		~
atribut_2	datetime		
atribut_3	bit	~	<b>V</b>
	bit	~	
	char(10) date datetime datetime2(7) datetimeoffset decimal(18, 0) float	•	



4. Nyní vybereme atribut, který bude naši tabulku identifikovat, čili zvolíme primární klíč. Klikneme pravým tlačítkem myši na náš zvolený atribut a zvolíme možnost "Set Primary Key" (viz. Obrázek 7). Po nastavení bude u atributu zobrazen klíč, který určuje primární klíč v dané tabulce (viz. Obrázek 8).

/HO	ME-NOTEBOOe_tabulka	_1*	
	Column Name	Data Type	Allow Nulls
<b>)</b> i	d_tabulka_1 ir	it	
1	Set Primary Key	:har(50)	
1	Insert Column	ime	
Ψ	Delete Column		
×3	Relationships		
1	Indexes/Keys		
AB	Fulltext Index		
Real Party	XML Indexes		
	Check Constraints		
<u>88</u>	Spatial Indexes		
-	Generate Change Script		

Obrázek 7

Column Name	Data Type	Allow Nulls
id_tabulka_1	int	
atribut_1	nvarchar(50)	Image: A start of the start
atribut_2	datetime	
atribut_3	bit	~

- Obrázek 8
- 5. V dolní části hlavního panelu se nachází další panel, který se nám zobrazí, když pracujeme s tabulkou. V tomto panelu definujeme vlastnosti konkrétního zvoleného atributu. Kromě jména, datového typu a povolení či zakázání vkládání null hodnot, zde můžeme určit další vlastnosti, které bude náš atribut mít. Mezi důležitou vlastností je "Identity". Tato vlastnost se nastavuje pouze pro primární klíč a nastavuje konkrétní atribut automaticky podle zvolených pravidel. Tato vlastnost je užitečná, abychom náhodou při ručním vkládání hodnot pro primární klíč nevložili hodnotu, která už v tabulce existuje. Pro vlastnost "Identity" můžeme nastavit krok, který bude mezi jednotlivými záznamy používat apod. (viz Obrázek 9).

IOME-NOTEBOOe_ta	ibulka_1*		
Column Name	Data Type	Allow Nulls	
id_tabulka_1	int		
atribut_1	nvarchar(50)	$\checkmark$	
atribut_2	datetime		
atribut_3	bit	~	
olumn Properties			
olumn Properties			
olumn Properties 2↓ IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII			
olumn Properties 2↓   3 Full-text Specification Has Non-SOL Server :	Subscriber		
olumn Properties 2↓ / 2↓ / 1 Journal Specification Has Non-SQL Server 3 3 Identity Specification () () () () () () () () () () () () () (	Subscriber		
Jumn Properties 2 Full-text Specification Has Non-SQL Server : 3 Identity Specification (Is Identity) Identity Increment	Subscriber		

Obrázek 9

#### Vytváření relací mezi tabulkami

Již umíme vytvářet tabulky včetně jejich atributů a jim definovat konkrétní vlastnosti. Nyní si ukážeme jak vytvářet vazby mezi tabulkami neboli relace. Tyto vazby určují vztah mezi jednotlivými tabulkami. Ukážeme si nyní jak takový to vztah, relaci či vazbu vytvořit. Proto, abychom vytvořili vazbu mezi dvěma či více tabulkami, si musíme vytvořit druhou tabulku. Podle postupu vytváření tabulky si vytvoříme druhou tabulku. Tato tabulka bude kromě svého primárního klíče obsahovat také atribut, který bude určovat cizí klíč v této tabulce (tzn. primární klíč s tabulky první). Na obrázku 10 je označen atribut, který je cizí klíč z tabulky první (já jsem ji pojmenoval "moje\_tabulka\_1" a primární klíč "id\_tabulka\_1").

Column Name	Data Type	Allow Nulls
🎖 id_tabulka_2	int	
id_tabulka_1	int	
atribut_1	float	
atribut_2	varchar(50)	
atribut_3	image	

#### Obrázek 10

A nyní přejdeme k vytvoření vazby mezi těmito tabulkami.

1. Klikneme pravým tlačítkem myši na cizí klíč v tabulce a zvolím volbu "Relationships…" (viz. Obrázek 11).



Obrázek 11

2. Zobrazí se nám dialogové okno, kde můžeme vytvářet vazby mezi tabulkami. Vazbu přidáme kliknutím na tlačítko "Add". Po stisku tlačítka se nám přidá implicitně vazba "sama na sebe". Abychom implicitní vazbu změnili, musíme rozkliknout tlačítko s třemi tečkami v práve části dialogového okna (je označen šipkou na obrázku 12).

Foreign Key Relationships		? 🛛
Selected Relationship:		
FK_moje_tabulka_2_moje_tabu	Editing properties for new re Specification' property needs will be accepted.	lationship. The 'Tables And Columns s to be filled in before the new relationship
	🗆 (General)	
	Check Existing Data On Cr	rea Yes
	표 Tables And Columns Speci	ific. 🔶 🛄
	Identity	
	(Name)	FK_moje_tabulka_2_moje_tabulka_2
	Description	
	🗉 Table Designer	
	Enforce For Replication	Yes
	Enforce Foreign Key Cons	tra Yes
	INSERT And UPDATE Spece	zific
< >>		
Add Delete		Close

Obrázek 12

3. Po rozkliknutí se zobrazí další dialogové okno (Obrázek 13), kde můžeme určit vazbu mezi dvěmi tabulkami. V levé části vybíráme tabulku, kde je atribut primárním klíčem a napravo vybereme tabulku, kde je atribut cizím klíčem. Musí se jednat o atributy mající stejný význam v těchto tabulkách. Vazby mohu samozřejmě určit i mezi tabulkami, kde primárním klíčem není pouze jeden atribut, ale je složen s více atributů. Z tabulky pak akorát nevyberu pouze jeden atribut, ale všechny, které mi určují primarní klíče popř. cizí klíč.

ables and Columns		?
Relationship <u>n</u> ame:		
FK_moje_tabulka_2_moje_tab	oulka_1	
Primary key table:		Foreign key table:
moje_tabulka_1	*	moje_tabulka_2
		id tabulka 1
id_tabulka_1		u_tabuka_1
id_tabulka_1		



4. Po stisku tlačítka "OK" mám vazbu mezi tabulkami vytvořenou. Aby jí ale management studio bralo na vědomí, musíme změny uložit. Při ukládání nám někdy management studio zobrazí upozornění, že ukládáte změny, které se netýkají pouze tabulky, kterou máte zobrazenou, ale taky tabulky, která se na ní váže. To je v pořádku, kliknětě na tlačítko "Yes". Pokud již nechcete, aby vám management studio tyto změny hlásil, tak odškrtněte volbu "Warn about Tables Affected". Dialogové okno je vidět na obrázku 14.

Save		? 🛛
The following tab	les will be saved to your databas	e. Do you want to continue?
moje_tabulka_1 moje_tabulka_2		
✓ ✓ <u>W</u> arn about Tables Al	fected	Save Text File

Obrázek 14

### Vytvoření databázového diagramu

Vytvořené tabulky a vazby mezi nimi je nejlépe patrno pomocí databázového diagramu, který lze pomocí management studia jednoduše vytvořit.

1. Klikneme pravým tlačítkem na složku "Database Diagrams" a zvolíme "New Database Diagram".



Obrázek 15

2. Vybereme tabulky z naší databáze, které chceme mít v diagramu zobrazeny.

Add Table		? 🔀
Tables moje_tabulka_1 moje_tabulka_2		
	Refresh Add C	lose

Obrázek 16

3. Na obrázku 17 vidíme tabulky, které jsme si v předešlém dialogu vybrali a vidíme taky vazby, které jsme vytvořili. Tabulky můžeme zobrazit taky ve standardním módu, abychom viděli i jednotlivé datové typy a povolení či zakázání vkládání null hodnot (Obrázek 18).



Obrázek 17

no	je_tabulka_2			Table View	•	Standard
	Column Name	Data Type	Allow Nulls	Sol Drimary Kay		Column Names
8	id_tabulka_2	int				Column Mariles
	id_tabulka_1	int		Insert Column	_ <u>_</u>	Keys
	atribut_1	float		T Delete Column		Name Only
	atribut_2	varchar(50)		🔯 🛛 Delete Tables from Datab	oase 🔳	Custom
	atribut_3	image	Image: A start of the start	Pemove from Diagram	100	Modify Custom
				Keniove from biagram	124	Hodiry Coscon
-	8 🖌	<u> </u>		Add Related Tables Add Related Tables Autosize Selected Tables Arrange Selection		Houry cascon
		<u> </u>		Add Related Tables       Image: Autosize Selected Tables       Arrange Selection       Relationships		riddi y custon
10	je_tabulka_1			Add Related Tables       Add Related Tables       Autosize Selected Tables       Arrange Selection       Relationships       Indexes/Keys		riddi y custoin
10	je_tabulka_1 Column Name	Data Type	Allow Nulls	Add Related Tables       Add Related Tables       Autosize Selected Tables       Arrange Selection       Relationships       Indexes/Keys       Fulltext Index		ridany coston .
10	je_tabulka_1 Column Name id_tabulka_1	Data Type int	Allow Nulls	Image: Second		ridary coston .
10	je_tabulka_1 Column Name id_tabulka_1 atribut_1	Data Type int nvarchar(50)	Allow Nulls	Add Related Tables       Add Related Tables       Autosize Selected Tables       Arrange Selection       Relationships       Indexes/Keys       Fulltext Index       XML Indexes       Check Constraints		ridan y custom .

Obrázek 18

### Přidávání hodnot do tabulek

Nyní máme vytvořené tabulky a vztah mezi těmito tabulkami. Také jsme si tyto vytvořené vazby mohli prohlédnout pomocí databázového diagramu. V této chvíli jsou tyto tabulky pochopitelně prázdné. Zde si ukážeme, jak můžeme tyto tabulky "ručně" naplnit nějakými daty.

1. Klikneme pravým tlačítkem na tabulku, kde chceme data přidávat a zvolíme volbu "Edit Top 200 Rows" (v management studiu express se ta volba nazývá trochu jinak)



 Zobrazí se nám tabulka v editačním módu, kde můžeme "ručně" přidávat data. Všimněte si, že primární klíč ručně nevložíte. To proto, že tento atribut má aktivní vlastnost "Identity" a vkládá se automaticky podle vámi definovaných pravidel (viz. Obrázek 20).



#### Obrázek 20

3. Při vkládání dat do tabulky, která obsahuje cizí klíč, si musíme dávat větší pozor. Management studio implicitně nepovolí vložit záznam s cizím klíčem, když pro tento cizí klíč neexistuje klíč primární z tabulky, na kterou se váže. Všimněte si také, že vkládám hodnotu pro "atribut\_3" jako null. To mohu, protože mám nastavenou vlastnost "Allow null" na hodnotu "true".

	id_tabulka_2	id_tabulka_1	atribut_1	atribut_2	atribut_3
	1	1	3,4	textik	NULL
	2	1	3,543	textik	NULL
	3	2	6,5	text	NULL
*	NULL	NULL 👗	NULL	NULL	NULL

#### Obrázek 21

## Generování skriptu

Již máme vytvořené tabulky, vazby mezi nimi a dokonce jsme jednotlivé tabulky naplnily daty. Nyní potřebujeme vytvořit nějaký skript, který nám všechny tyto vytvořené fragmenty zabalí do jednoho balíčku (soubor apod.). Ukážeme si, jak na to.

1. Pravým tlačítkem myši klikneme na naši databázi v levém panelu management studia a zvolíme volbu "Generate Scripts…" jak vidíme na obrázku 22.



Obrázek 22

2. Vyskočí nám dialogové okno, které nás provede postupně k uložení skriptu. To, co chceme mít ve skriptu zobrazeno, musíme vybrat. Pokud chceme vygenerovat vše, tak označíme vždy možnost "Select All" v dolní části dialogu. Tzn. tabulky, popř. uložené procedury apod. Na obrázku 24 vidíme možnosti, co náš skript vygeneruje. Pokud chceme vygenerovat skrip i s daty, musíme povolit tuto možnost, na obrázku je zvýrazněna červenou barvou.

Script Wizard - HOME-NOTEBOOK	
Select Database Select the database you want to script.	
Select a database:	
db_kalkul_test eLogika_ eLogika_1 eLogika_271009 eLogikaTemp master model mode mode mode mode nova_databaze msdb ReportServer ReportServer ReportServer ReportServerTempDB tempdb	
Script <u>a</u> ll objects in the selected database	
Help < <u>B</u> ack <u>Next</u>	Cancel

Obrázek 23

<u>D</u> ptions S⊒∫ <b>2</b> ↓ □		
Script Logins	False	~
Script Object-Level Permissions	False	
Script Statistics	Do not script statistics	
Script USE DATABASE	True	
Schema qualify object names.	True	
3 Table/View Options		
Script Data	True	V)
Script Data Compression Options	False	
Script Foreign Keys	True	
Script Full-Text Indexes	False	1000
Script Change Tracking	False	×
Script Data Scripts the data for each table scripted.		

Obrázek 24

3. Postupujeme průvodcem dál, až dorazíme do kroku, kdy si vybíráme, kam chceme skript uložit. Na výběr máme možnost uložení do "clipboardu" (po stisku "ctrl + v" bude skript zobrazen), možnost uložení do nové záložky v hlavním panelu management studia nebo do souboru s implicitní příponou sql (viz. Obrázek 25).

📱 Script Wizard - HOME-NOTEBOOK		
Output Option Select the destination for th	ne Script Wizard output.	M)
Script mode		
💿 Script to file		$\overline{\mathbf{v}}$
File name:	C:\Documents and Settings\Admin\	Browse
Files to generate	⊙ Single file	
	O File per <u>o</u> bject	
<u>S</u> ave as:	O Unicode text	
	O ANSI text	
Script to <u>Clipboard</u>		
🔘 Script to New Query <u>W</u>	(indow	
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish >>	Cancel

#### Obrázek 25

4. Po stisku na tlačítko "Finish" spustíme generování.

# Spuštění skriptu

Máme vygenerovaný skript. Ale nyní potřebujeme vědět, jak tento vygenerovaný skript umět zpětně spustit a naši databázi společně s tabulkami, vazbami a naplněnými daty pomocí tohoto skriptu vytvořit.

1. Prvním krokem je otevřít skript, který jsme si vygenerovali. Tento úkon nám znázorňuje obrázek 26 a obrázek 27.

Aicrosoft SQL Server Manag	ement Studio
Edit View Project Debu Connect Object Explorer Disconnect Object Explorer New Open Add Close Close Solution Save Solution 1 Ctrl+S Save Solution 1 AS Save Solution 1 AS Save Solution 1 AS Save Solution 1 Ctrl+Shift+S Page Setup Print Ctrl+P Recent Files Recent Projects Exit South Setup Setup Agent	g Tools Window Community Help Toble View - Reference
Obraz	2ek 20
Delumentu	
Jokumenty	Northwind_log.LDF
ime:	Open Cancel
	icrosoft SQL Server Manage         Edit View Project Debut         Connect Object Explorer         Disconnect Object Explorer         New         Open         Add         Close Solution         Save Solution1         Save Solution1 As         Chrl+P         Recent Files         Recent Projects         Exit         Solumenty         Dakumenty         Solution1 As         Solution1 As         Solution2 As         Solution3 As         Solution4 As         Solution4 As         Solution5 As         Solution6 As         Solution7 As         Solution6 As         Solution6 As



2. Nyní se nám zobrazí skript, který jsme si vygenerovali. První příkaz nám říká, ve které databázi se má skript provézt (vždy to nemusí být první příkaz). Pokud databázi tohoto jména nemáme vytvořenou, management studio neví, kam má údaje vložit, tudíž se skript nikdy nespustí. Proto je radno databázi stejného jména vytvořit a poté skript spustit nebo vytvořit novou databázi s jiným jménem, ale potom musím jméno databáze přejmenovat také ve vygenerovaném skriptu (jméno databáze následuje po příkazu "use", viz Obrázek 28). Pro spuštění skriptu, stiskneme tlačítko "Execute" označené červenou barvou také na obrázku 28.



Obrázek 28

3. Výsledek spuštění skriptu bude buď úspěšný anebo neúspěšný. Informaci o úspěšném spuštění skriptu vidím na obrázku 29.



Obrázek 29