

Učební pomůcka Microsoft SQL

Daniela Turečková

Bakalářská práce
2017



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Daniela Turečková**
Osobní číslo: **A13251**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační technologie v administrativě**
Forma studia: **prezenční**

Téma práce: **Učební pomůcka MS SQL 2014**
Téma anglicky: **Lecture Notes for the Microsoft SQL Server**

Zásady pro vypracování:

1. Seznamte se s MS SQL v aktuální edici.
2. Zpracujte popis instalace.
3. Zpracujte podklady pro cvičení a přednášky v rozsahu 14 týdnů.
4. Realizujte sadu vzorových příkladů.
5. Navrhněte sadu 120 testovacích úkolů včetně řešení ve formátu vhodném pro moodl.
6. Uvedte možné směry rozvoje učební pomůcky.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. JORGENSEN, Adam. Professional microsoft sql server 2014 administration. 1st edition. Indianapolis, IN: John Wiley and Sons, 2014, pages cm. ISBN 1118859138.
2. BEN-GAN, Itzik, Dejan SARKA a Ron TALMAGE. Querying Microsoft SQL Server 2012: exam 70-461 training kit. Sebastopol, Calif.: Microsoft, c2012, xxx, 704 p. ISBN 0735666059.
3. KROENKE, David a David J AUER. Databáze. 1. vyd. Brno: Computer Press, 2015, 496 s. ISBN 978-80-251-4352-0.
4. MASOOD-AL-FAROOQ, B. A. SQL Server 2014 Development Essentials. UK.: Packt Publishing, 2014. ISBN 978-1-78217-255-0.
5. CORONEL, Carlos a Steven MORRIS. Database systems: design, implementation, and management. 11e [edition]. United States: Course Technology Cengage Learning, 2015, xxvii, 751 pages. ISBN 128519618x.

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

3. února 2017

Termín odevzdání bakalářské práce:

30. května 2017

Ve Zlíně dne 3. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



Ing. Miroslav Matýsek, Ph.D.
ředitel ústavu

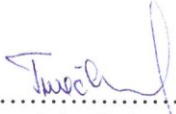
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 25. května 2017


.....
podpis diplomanta

ABSTRAKT

Cílem této bakalářské práce je vytvořit podklady, které budou mít využití pro výuku v programu Microsoft SQL Server 2016. Práce je rozdělena na teoretickou a praktickou část. Teoretická část objasňuje základní pojmy a úvod do problematiky databázových systémů. Dále obsahuje popis programu Microsoft SQL a jeho edice a verze. Praktická část práce se věnuje tvorbě databáze, popisuje tvorbu dotazů a práci s nimi v prostředí Microsoft SQL Server 2016. Přílohou této práce jsou prezentace, které obsahují materiály pro výuku v jednom semestru.

Klíčová slova: Microsoft SQL Server 2016, databáze, databázové systémy

ABSTRACT

The aim of this bachelor thesis is to create educational materials that will be used for teaching in Microsoft SQL Server 2016. The thesis is divided into a theoretical and practical part. The theoretical part explains the basic concepts and introduction to database systems. It also contains a description of Microsoft SQL and its edition and version. The practical part includes creating SQL queries, description of commands, keywords for work in Microsoft SQL Server 2016. Attachment of this bachelor thesis are presentations, which contain materials for teaching in semester.

Keywords:

Microsoft SQL Server 2016, database, database system

Děkuji svému vedoucímu bakalářské práce Ing. Petru Šilhavému, Ph.D., za ochotu a užitečné rady při vypracovávání práce. Děkuji také mé rodině a kamarádům za podporu při psaní bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 DATABÁZOVÉ SYSTÉMY	11
1.1 DATABÁZOVÉ MODELY	11
1.1.1 Hierarchický model.....	11
1.1.2 Síťový model.....	12
1.1.3 Relační model.....	12
1.1.4 Objektový model.....	13
1.1.5 Objektově relační model	13
1.2 RELAČNÍ DATABÁZE.....	13
1.2.1 Entity.....	13
1.2.2 Atributy	14
1.2.3 Relace.....	14
1.2.4 Tabulky	16
1.2.5 Typy klíčů	17
1.2.6 Sloupce a datové typy	17
1.3 SQL JAZYK.....	19
1.4 ZABEZPEČENÍ.....	20
1.4.1 Autorizace	21
1.4.2 Autentizace.....	21
2 MICROSOFT SQL 2016	22
2.1 VÝVOJ MS SQL.....	22
2.2 POŽADAVKY NA SYSTÉM	23
2.3 EDICE	23
2.3.1 Enterprise	23
2.3.2 Standard.....	23
2.3.3 Express	24
2.3.4 Developer	24
2.3.5 Porovnání	24
II PRAKTICKÁ ČÁST	26
3 PRÁCE V PROSTŘEDÍ MS SQL SERVER 2016	27
3.1 INSTALACE MS SQL SERVERU	27
3.2 INSTALACE MANAGEMENT STUDIA.....	31
3.3 ZÁKLADNÍ OPERACE PRO TVORBU DATABÁZE	33
3.3.1 Vytvoření databáze.....	34
3.3.2 Vytvoření a změna tabulky	34
3.3.3 Vkládání dat do tabulky	36
3.3.4 Zobrazení vložených dat	36
3.3.5 Zobrazení vložených dat s podmínkou	37
3.3.6 Spojování tabulek.....	39
3.3.7 Agregáčn� funkce	41
3.3.8 Pr�ce s datem a �asem.....	42
3.3.9 Vnořen� dotazy	43
3.3.10 Skl�d�n� dotazů	45

3.3.11	Proměnné.....	46
	Podmínky.....	47
3.3.12	Uložené procedury	47
3.3.13	Trigger.....	48
4	PODKLADY PRO VÝUKU	49
	ZÁVĚR	50
	SEZNAM POUŽITÉ LITERATURY.....	51
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	54
	SEZNAM OBRÁZKŮ	55
	SEZNAM TABULEK.....	57
	SEZNAM PŘÍLOH.....	58

ÚVOD

S databázovými systémy se setkává každý z nás v každodenním životě. Funkci databázových systémů využíváme denně v zaměstnání, například docházkový systém nebo seznam zaměstnanců. Pokud půjdeme do městské knihovny, narazíme opět na databázový systém, ve kterém nejprve knihu vyhledáme, poté ji najdeme na konkrétním místě a nakonec si ji vypůjčíme domů. Při běžném nákupu také najdeme na využití databázového systému, například evidence čárových kódů a odpovídajícího zboží k nim.

V současné době mají databázové systémy rozšířené využití a do budoucna předpokládáme stále větší integraci databázových systémů. Tato práce je vhodná pro začátečníky v oblasti informační techniky a databázových systémů. Jsou zde popsány základní operace pro práci s běžnou databází.

Cílem této bakalářské práce je vytvořit podklady pro výuku databázových systémů. Teoretická část práce obsahuje seznámení s databázemi v podobě definic a pojmů. Objasňuje funkce a principy databázového systému, které by měly být osvojeny pro práci s databázemi. Druhá část představuje Microsoft SQL Server, jeho vznik a vývoj. V praktické části práce je využíván systém Microsoft SQL Server 2016, což je nejnovější aktualizovaná verze programu, která je dostupná na oficiálních webových stránkách společnosti ve zkušební verzi. Je zde popsán postup při instalaci, spuštění systému a základní operace.

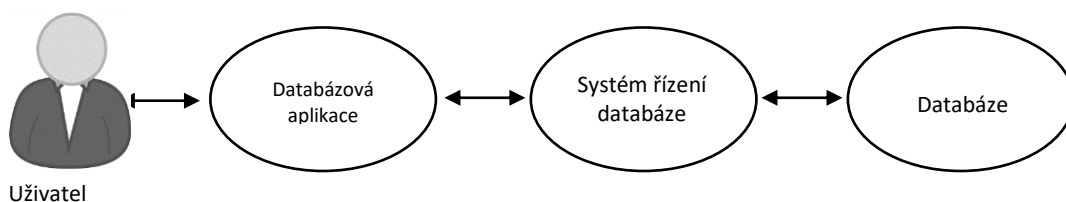
Přílohu této bakalářské práce tvoří prezentace, které slouží jako podkladový materiál pro přednášejícího a také prezentace, které budou využívány na cvičení. V prezentacích jsou obsaženy i úlohy, pro ověření pochopení učiva v daném týdnu. Další přílohu tvoří testové úlohy, které slouží pro využití v Moodle a obsahují otázky jak z teoretické, tak i praktické části.

I. TEORETICKÁ ČÁST

1 DATABÁZOVÉ SYSTÉMY

Databází rozumíme souhrn dat, se kterými lze dále pracovat, jako s ucelenou jednotkou. Jedná se o vhodný nástroj pro udržování přehledu v různých odvětvích.

Databázový systém se skládá z uživatele, databázové aplikace, systému řízení databáze a vlastní databáze [3]. Databáze je souhrn tabulek. Systém řízení databáze je software, který umožňuje práci s databází. Požadavky zadané v jazyce SQL jsou převáděny na aktivity databáze. Jedná se o licencovaný program. Databázová aplikace slouží pro komunikaci mezi uživatelem a systémem řízení databáze. Aplikace čte data a odesílá příkazy SQL systému řízení databáze. Naopak prezentuje uživateli data v podobě sestav a formulářů. Pomocí databázové aplikace uživatelé načítají a zadávají data pomocí formulářů a vytvářejí sestavy. Příkladem takového softwaru je Microsoft Access, Microsoft SQL, Oracle atd.



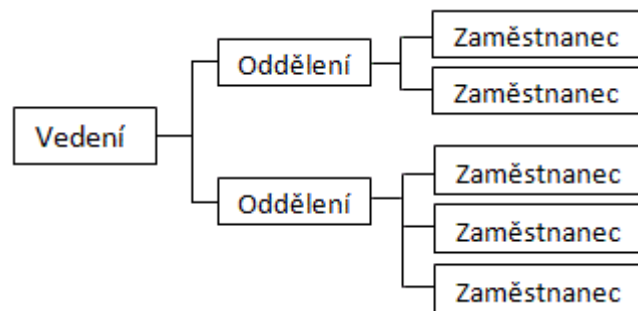
Obr. 1. Komponenty databázového systému

1.1 Databázové modely

Databázové modely určují způsob uspořádání dat. Je to architektura, podle které jsou data ukládána do databáze a následně vzájemně provázána [6]. Rozlišujeme hierarchický model dat, síťový model, relační model, objektový model a objektově relační model dat.

1.1.1 Hierarchický model

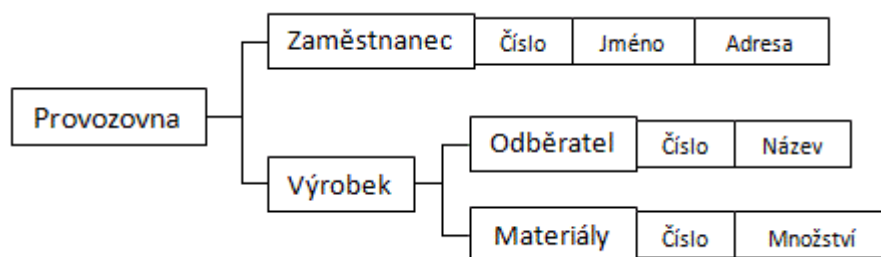
Jedná se o model, podle kterého byly postaveny první databáze. Vznikl v roce 1960 jako náhrada souborových systémů, kde byly záznamy uspořádány dle hierarchie. Záznamy jsou mezi sebou propojeny tak, aby bylo možné definovat nadřízenost nebo podřízenost jednotlivého záznamu [3]. Struktura vychází z klasického vztahu rodič – potomek. Každý rodič může mít několik potomků. Naopak každý potomek může mít pouze jednoho rodiče. Tento model má striktně definovanou strukturu databáze [6], což může být v některých případech nevýhoda.



Obr. 2. Hierarchický model [17]

1.1.2 Síťový model

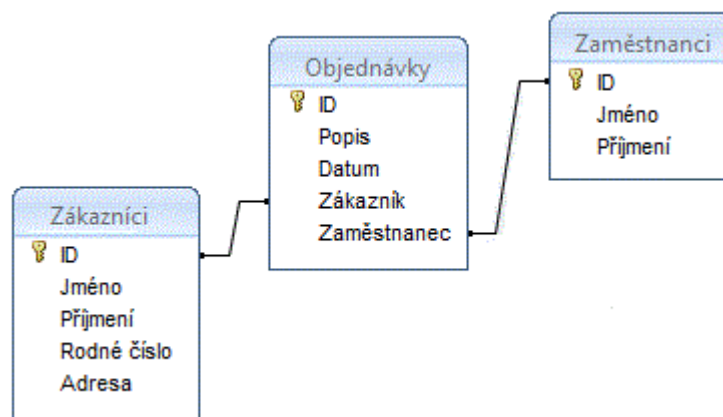
Tento model vznikl přibližně ve stejné době jako hierarchický model. Na rozdíl od předchozího modelu, umožňuje mnohonásobné vztahy v databázi. Prostřednictvím konkrétní relace může být nařízen přechod z jednoho záznamu na jiný. Jeden typ záznamu může být účastníkem několika relací. Tento databázový model je nepružný a změna struktury systému je velmi obtížná, což jsou klíčové nevýhody [3]. V dnešní době se tento model již nevyužívá.



Obr. 3. Síťový model [17]

1.1.3 Relační model

Nejrozšířenější model databázového systému. Ztotožňuje se s myšlenkou, že jedna předem definovaná cesta, je omezující. V tomto datovém modelu máme možnost data svázat podle potřeby. Hierarchický i síťový datový model pracuje pouze s jednotlivými záznamy, relační model umožňuje práci s určitou datovou množinou [6]. Umožňuje tvorbu jednorázových dotazů. Data jsou ukládány do tabulek, které mohou být vzájemně propojeny pomocí atributů. Záznamy jsou ukládány do řádků tabulek a je nutné zachovat integritu dat [3]. Tento model je využíván například v programu Microsoft Access, Microsoft SQL nebo MySQL.



Obr. 4. Relační model [17]

1.1.4 Objektový model

Tento model představuje alternativu relačního modelu dat, s tím rozdílem, že je schopen pracovat se složitějšími datovými typy, například obrázky a výkresy, zvukové soubory nebo video [3]. Data jsou ukládána do tzv. objektů, což je seskupení příbuzných dat a programové logiky, která reprezentují věc z reálného světa, například zákazníka, zaměstnance, výrobek. U tohoto datového modelu se nachází proměnné datové položky, které jsou uloženy u každého objektu, jedná se o identifikátory. K těmto proměnným lze přistupovat jen pomocí metod, což znamená zapouzdření objektů [6].

1.1.5 Objektově relační model

Tento model představuje kombinaci relačního modelu a prvků objektového modelu dat. Stejně jako relační model dat umožňuje tvorbu jednorázových dotazů, z objektového modelu využívá zapouzdření [3]. V praxi nenachází hojné využití.

1.2 Relační databáze

1.2.1 Entity

Entita je reálný objekt, který využíváme pro potřeby naší databáze. Je důležitá pro logické oddělení dat. Každá entita může tvořit samostatnou tabulku. Entita je například Zaměstnanec a vyjadřuje všechny zaměstnance dané společnosti. Instance této entity jsou potom jednotliví zaměstnanci [6]. Pod pojmem entita si můžeme představit jednotlivé řádky tabulky v databázi.

Dále rozlišujeme externí entitu. Jedná se o takovou entitu, o které není nutné uchovávat nějaké informace. Například to můžou být informace o úvěrové společnosti, když společnost svým zákazníkům poskytuje úvěrový limit [6]. Externí entity se v praxi moc nepoužívají.

1.2.2 Atributy

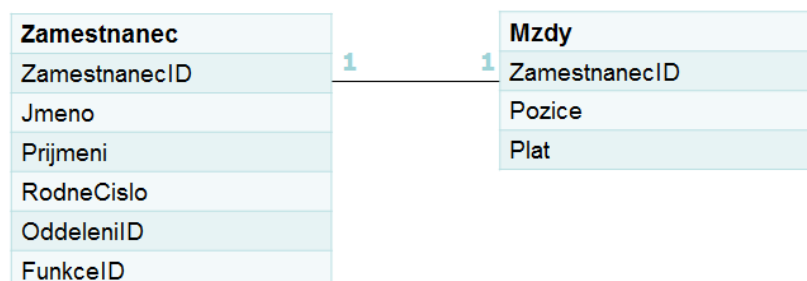
Atribut je jednotka, která identifikuje nebo popisuje entitu. Rozlišujeme jedinečný identifikátor entity, což je unikátní kód, který charakterizuje entitu. Například pro entitu Zaměstnanec, je jedinečný identifikátor Číslo Zaměstnance, proto musí mít každý atribut jinou hodnotu. Každý atribut by měl být atomický, což znamená, že by to měla být nejmenší definovaná hodnota [6]. Pokud máme atribut Adresa, je zřejmé, že jej lze dále rozdělit na Ulici, Číslo Popisné atd. Pod pojmem atribut si můžeme představit jednotlivé sloupce v tabulce.

1.2.3 Relace

Do každé databáze vkládáme data, která jsou nějakým způsobem propojena. Toto propojení zajišťují relace, které znázorňují vztahy mezi jednotlivými entitami. Slouží k vzájemné komunikaci mezi tabulkami. Rozlišujeme tři základní typy relací: relace typu jedna k jedné, relace typu jedna k více a relace typu více k více. Dále existuje rekurzivní relace.

Relace 1:1

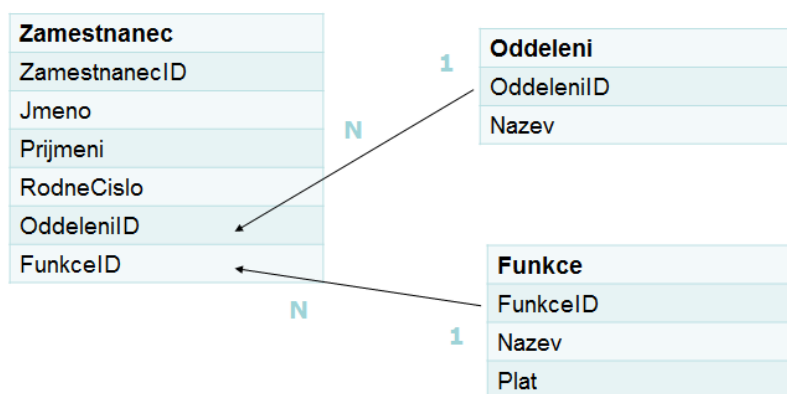
Relace typu jedna k jedné. Jde o relaci, kdy jeden záznam z první tabulky odpovídá právě jednomu záznamu z tabulky druhé a naopak [4]. V praxi se tento typ téměř nevyužívá, protože lze záznamy spojit do jedné tabulky.



Obr. 5. Relace typu 1:1 [4]

Relace 1:N

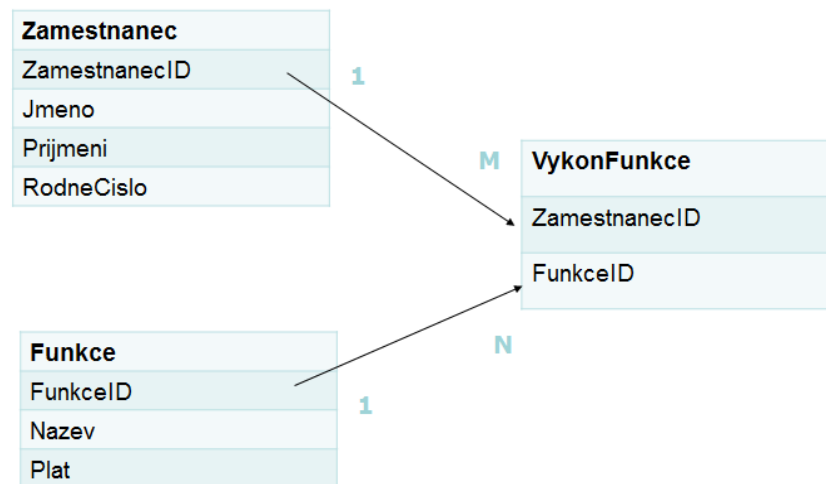
Relace typu jedna k více. Jde o relaci, kdy jeden záznam z první tabulky odpovídá jednomu nebo více záznamům z tabulky druhé. Naopak každý záznam z druhé tabulky může být propojen právě s jedním záznamem z tabulky první [4]. Například každý zákazník může mít v databázi několik objednávek, ale každá objednávka může mít pouze jednoho zákazníka. V praxi se jedná o nejpoužívanější typ relací.



Obr. 6. Relace typu 1:N [4]

Relace M:N

Relace typu více k více. Jedná se o relaci, kdy několik záznamů z první tabulky odpovídá několika záznamům z tabulky druhé a naopak [4]. Například jeden zaměstnanec obsluhuje více zákazníků a jeden zákazník je obslužen několika zaměstnanci. Tento druh relace není zpravidla relačním modelem podporován. Pro použití této relace musí být vytvořena pomocná tabulka, která je s dalšími dvěma propojena relací typu 1:N.



Obr. 7. Relace typu M:N [4]

Rekurzivní relace

Rekurzivní relace znázorňuje vztahy v rámci jedné entity, nikoli vztahy mezi entitami jak bylo uvedeno výše. Rozlišujeme tři typy: jedna k jedné, jedna k více a více k více [6]. Relaci typu 1:1 využijeme v případě, kdy budeme hledat příbuzenský vztah mezi zaměstnanci společnosti. Nejčastěji využívaná relace typu 1:N může sledovat vztahy na pracovišti. Jeden zaměstnanec je podřízen jednomu nebo žádnému zaměstnanci. Naopak pracovník na vedoucí pozici bude mít několik podřízených pracovníků. Relace typu M:N je využita v případě, že daný výrobek se skládá z několika součástí a zároveň je součástí dalšího výrobku.

1.2.4 Tabulky

Tabulka je hlavní jednotkou [6] pro ukládání dat, která se skládá z řádků a sloupců. Entitu reprezentuje řádek tabulky a každý sloupec tabulky reprezentuje atribut dané entity. Tabulky jsou ukládány do tzv. tabulkového prostoru, což jsou soubory operačního systému fyzické vrstvy. Do takového prostoru můžeme umístit několik tabulek. Naopak tabulka může být rozdělena a umístěna v několika tabulkových prostorech.

Při vytvoření tabulky je nutné uvést název tabulky. Ten by měl být jednoduchý a výstižný. Některé databázové programy umožňují používat mezery, které se vymykají oborovým standardům, a může nastat problematika v komunikaci mezi databázemi [6]. Jiné databázové programy umožňují použití velkých počátečních písmen, což usnadňuje čitelnost názvu tabulky. Naopak některé databázové programy vyžadují použití výhradně velkých

písmen. Pro lepší čitelnost se mohou jednotlivá slova oddělovat pomlčkou nebo podtržítkem.

1.2.5 Typy klíčů

Každý záznam v tabulce by měl obsahovat klíč, který slouží jako identifikátor. Klíče rozlišujeme na složené, primární, náhradní nebo cizí.

Složené klíče

Jedná se o klíč, který se skládá ze dvou a více atributů [3]. Například v tabulce Zaměstnanci, může být klíčem Příjmení a Oddělení, ale pouze v případě, že není shoda v příjmení žádného zaměstnance. Jedná se o jedinečný klíč. Pokud nastane shoda v příjmení, můžeme použít složený klíč se třemi atributy, kterými bude Jméno, Příjmení a Oddělení.

Primární klíč

Primární klíč musí být jedinečný [3] a nezaměnitelný. Můžeme ho vytvořit pomocí datového typu, který zajistí, že údaj nebude nabývat nulové hodnotě. Vhodné použití primárního klíče jsou čísla dokladů nebo rodná čísla.

Náhradní klíče

Jedná se o unikátní identifikátor, který je doplněn do tabulky, aby splňoval podmínky primárního klíče [3]. Měl by být vytvořen pomocí číselného datového typu tak, aby nedocházelo k nulové hodnotě. Je vhodné jej použít v případě, že máme tabulku s adresami, která obsahuje sloupce Ulice, Číslo popisné, PSČ a Město. Ani jeden z uvedených atributů není vhodným primárním klíčem, proto můžeme do tabulky přidat sloupec s náhradním klíčem s názvem AdresaID.

Cizí klíče

Cizí klíč slouží pro vytváření relací [3]. Například v tabulce Zaměstnanci máme sloupec Oddělení, na kterém zaměstnanec pracuje. Jedná se o cizí klíč. Tato tabulka je spojena s tabulkou Oddělení, kde nalezneme jednotlivá oddělení a kontakty na ně.

1.2.6 Sloupce a datové typy

Sloupec v tabulce představuje určitý atribut, který musí mít svůj název a datový typ. Název musí být unikátní v rámci jedné tabulky. Datový typ představuje formát jednotlivého sloupce, například číslo, text, datum nebo měna.

Výhodou je integritní omezení, což znamená omezení pouze na takové znaky, které jsou pro daný sloupec použitelné. Například tam, kde má být kalendářní rok, nemůžeme vložit jiný text. Další výhodou je úspora úložiště, jelikož čísla se ukládají v interním formátu, který je efektivnější [6]. Datové typy se liší v závislosti na používaném programu.

Pro ukládání znaků používáme typ CHAR nebo VARCHAR [33]. Rozdíl mezi těmito datovými typy je v ukládání dat. Například 5 znaků bude uloženo pomocí typu CHAR(20), kde bude využita paměť všech 20 B. Naopak u datového typu VARCHAR(20) bude využito pouze 5 B + něco navíc, což je výhodnější.

Tab. 1. Textové datové typy používané v MS SQL Server [10]

Datový typ	Popis	Paměťová náročnost
VARCHAR (x)	Slouží pro vyjádření textu s proměnnou délkou. Je nutné zadat maximální počet znaků.	x v rozmezí 1 – 8000 znaků, 1B na znak + 2B navíc
CHAR (x)	Do tohoto datového typu je možné uložit až 255 znaků.	x v rozmezí 1 – 8000 znaků, 1B na znak

Základním datovým typem pro ukládání celého čísla je integer, neboli INT. Na základě rozsahu lze použít TINYINT, SMALLINT nebo BIGINT. K uložení desetinného čísla se používá datový typ DECIMAL nebo NUMERIC. Typy jsou totožné.

Tab. 2. Číselné datové typy používané v MS SQL Server [10]

Datový typ	Popis	Paměťová náročnost
TINYINT	Vyjadřuje celé číslo v rozsahu 8bitů.	Paměťová náročnost 1B. V intervalu -128 – 127.
SMALLINT	Vyjadřuje celé číslo v rozsahu 16bitů.	Paměťová náročnost 2B. V intervalu -32 768 – 32 767.
INT	Vyjadřuje celé číslo v rozsahu 32bitů.	Paměťová náročnost 4B. V intervalu -2 147 483 648 – 2 147 483 647.
BIGINT	Vyjadřuje celé číslo v rozsahu	Paměťová náročnost 8B.

	64bitů.	V intervalu -9 223 372 036 854 775 808 – 9 223 372 036 854 775 80.
DECIMAL	Vyjadřuje číslo s desetinnou čárkou. Je nutné zadat celkový počet číslic a počet číslic za desetinnou čárkou. Při zápisu desetinného čísla v příkazovém řádku používáme jako oddělovač tečku.	Velikost 5 – 17B v závislosti na přesnosti. V intervalu -1038 + 1 – 1038 – 1.

Mezi základní datové typy pro práci s datem a časem patří DATE, TIME a DATETIME. TIME představuje místní čas ve 24 hodinovém formátu, paměťová náročnost je závislá na přesnosti. DATE znázorňuje datum jako textový řetězec.

Tab. 3. Datové typy pro datum a čas používané v MS SQL Serveru [10]

Datový typ	Popis	Paměťová náročnost
DATE	Datum jako textový řetězec RRRR-MM-DD, 0001-01-01 až 9999-12-31	3 B
TIME	Čas jako textový řetězec HH:mm:ss.	3 – 5 B
DATETIME	Vyjádření data a času ve formátu YYYY-MM-DD hh:mm:ss.	8 B

1.3 SQL jazyk

Jazyk Structured Query Language (SQL) vznikl v 70. letech 20. století [3] ve společnosti IBM Corporation. Jedná se o příkazově orientovaný nástroj relačních databází, který podporují všechny databázové systémy. Příkazy zadáváme pomocí tzv. klauzulí, které obsahují klíčová slova. V názvech tabulek nebo sloupců nesmí být tato klíčová slova použita, jelikož se jedná o slova rezervovaná [6]. Příkazy se zadávají v určeném pořadí a každý příkaz končí středníkem.

SQL příkazy rozdělujeme do čtyř skupin: jazyk pro manipulaci s daty, jazyk pro definici dat, jazyk pro řízení dat a jazyk pro řízení transakcí.

Jazyk pro manipulaci s daty

Pro tuhle skupinu používáme název Data Manipulation Language (DML) a řadíme sem příkazy, které provádějí změny v tabulkách [18]. Patří sem příkazy INSERT (vloží jeden nebo více řádků do tabulky), UPDATE (změní jeden nebo více řádků v tabulce), DELETE (odstraní jeden nebo více řádků z tabulky). Najdeme zde i příkazy, které v databázi neprovádějí žádné změny, jen se dotazují [4], takový příkaz existuje pouze jeden, příkaz SELECT. Tento příkaz zobrazuje řádky z jedné nebo více tabulek.

Jazyk pro definici dat

Název pochází z anglického Data Definition Language (DDL) umožňuje vytvářet a upravovat databázové struktury [18], například tabulky a indexy. Zahrnujeme sem příkazy CREATE (vytvoření tabulky), ALTER (změna tabulky) a DROP (odstranění tabulky). Dále zajišťují správu datových kontejnerů [4]. Na rozdíl od předchozí skupiny, která pracuje přímo s daty a komunikuje s daty uvnitř těchto kontejnerů.

Jazyk pro řízení dat

Tato skupina se nazývá Data Control Language (DCL) a řadí se zde příkazy, které upravují oprávnění každého uživatele [4]. Jedná se o příkazy GRANT (přidělení přístupových práv a rolí uživatele) a REVOKE (odebrání přístupových práv a rolí uživatele).

Jazyk pro řízení transakcí

Tato skupina se nazývá Transaction Control Language (TCL) [18] a umožňuje seskupit příkazy SQL do transakce. Ve výsledku je možné SQL příkazy v transakci potvrdit nebo odvolat jako jeden příkaz. Patří sem příkazy BEGIN TRANSACTION (začátek transakce), COMMIT (potvrzení transakce) a ROLLBACK (odvolání transakce).

1.4 Zabezpečení

Slouží k zamezení neoprávněného přístupu [9], ochranou před změnou nebo destrukcí. Jedná se o autorizaci a autentizaci.

1.4.1 Autorizace

Autorizace znamená umožnění přístupu do databáze konkrétnímu uživateli. Je možné kontrolovat k jakým objektům má jednotlivý uživatel přístup a zda jej může nějakým způsobem editovat [15]. Dle úrovně pohledu na databázi jsou určena privilegia daného uživatele. K databázi může být přístup jen pomocí pohledu [9], což zahrnuje zákaz určitých operací s daty. Je prováděna na základě seznamů pro řízení přístupu. Proces autorizace často přechází k autentizaci.

1.4.2 Autentizace

Autentizace znamená ověření identity konkrétního uživatele. Zajišťuje ochranu před falšování identity, předcházíme tomu, že se osoba vydává za někoho, kdo není [15]. Základní metodou je použití vhodné kombinace uživatelského jména a hesla. Dalším možným řešením je udělení hardwarového klíče nebo primárního klíče, nebo otisk prstu.

2 MICROSOFT SQL 2016

Microsoft SQL Server 2016 je dosud nejnovější vydaná verze programu pro tvorbu relačních databází. Je k dispozici ve čtyřech různých edicích a na oficiálních stránkách lze stáhnout zkušební verzi na 30 dní zdarma.

2.1 Vývoj MS SQL

Počátku SQL Serveru sahají do roku 1988. Jednalo se o aplikaci vytvořenou firmou Microsoft ve spolupráci s firmou Sybase [19]. V roce 1993 vznikla první databázová aplikace pro platformu Windows. Databáze měla nízkou funkčnost a byla schopna zpracovávat databáze s malým počtem dat. Společnost Microsoft ukončila spolupráci s firmou Sybase v roce 1994 a pokračovala už jako samostatná jednotka.

V roce 1995 vznikla verze MS SQL Server 6.0. Databáze měla vyšší výkon a obsahovala vylepšené funkce [19]. Byla vhodná pro zpracování databází malých firem. O rok později byla vydaná verze MS SQL Server 6.5.

Verze MS SQL Server 7.0 vznikla v roce 1998 [19], kde se objevila možnost využití webových databází. Novinkou byly business nástroje, například analytické služby nebo transformace dat.

Vyšší výkon a spolehlivost přinesla verze MS SQL Server 2000. Verze podporovala online operace a také se zvýšily náklady na jednotlivé edice. V následujících dvou letech Microsoft ovládl trh s databázovými systémy, jelikož ceny za systémy byly oproti konkurenčním produktům nízké.

MS SQL Server 2005 obsahuje nové funkce, například službu Integration Services a Analysis Services. Velkou inovací znamenalo zavedení .NET Framework.

Další verzí byl MS SQL Server 2008 a MS SQL Server 2008 R32. Obsahovalo nové datové typy a vlastnosti a rozšíření jazyka T-SQL.

Verzí 11.0 byl MS SQL Server 2012. Předposlední vydanou verzí je MS SQL Server 2014 a v současné době nejnovější je verze MS SQL Server 2016. Obě poslední verze vynikají výkonem, spolehlivostí a vysokou úrovní zabezpečení.

2.2 Požadavky na systém

Tab. 4. Požadavky na operační systém [16]

Paměť	Minimálně 1 GB (512 MB pro edici Express). Doporučeno 4 GB.
Procesor	Pro 64-bitovou instanci minimálně s rychlostí 1,4 GHz. Doporučená rychlost je 2.0 GHz a vyšší.
Typ procesoru	AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support.

2.3 Edice

Microsoft SQL 2016 je dostupný ve čtyřech edicích [8]:

- Enterprise
- Standard
- Express
- Developer

2.3.1 Enterprise

Edice Enterprise zajišťuje komplexní řešení business intelligence a umožňuje zpracovávat ty nejnáročnější databáze [11]. Technologie Always On okamžitě obnovuje data po havárii a slibuje vysokou dostupnost dat v případě havárie. Obsahuje technologii Always Encrypted, která zabezpečuje data na úrovni řádků, hloubkovým auditováním a dynamickým maskováním dat. Tím je zajištěno nejmenší množství slabých míst. Umožňuje sdílet data na různých mobilních zařízeních, které mají operační systém Windows, Android nebo iOS.

2.3.2 Standard

Edice Standard obsahuje až 24 jader a zajišťuje tak vyšší výkon. Slibuje rychlejší analýzu dat a uchování dat v paměti pro všechny úlohy [12]. Technologie Always Encrypted zajišťuje nejlepší zabezpečení dat v rámci organizace i mimo ni. Rychlá synchronní replika se dvěma uzly umožňuje okamžitou obnovu po havárii a zajišťuje, aby nedošlo ke ztrátě dat.

Služba SQL Server Analysis Services zobrazuje přehledy na základě zadaných dat, kterým je zajištěno komplexní řešení Business Intelligence. Edice obsahuje pokročilé analytické nástroje přímo v databázi.

2.3.3 Express

Jedná se o bezplatnou základní databázi, která je vhodná například pro výuku. Na oficiálních stránkách je ke stažení ve třech edicích [13]. První je zjednodušená verze, která neumožňuje vzdálený přístup do databáze. Druhá edice s pokročilými službami obsahuje kompletní funkce MS SQL Express. Třetí edice umožňuje vzdálenou správu, ale neobsahuje pokročilé služby.

2.3.4 Developer

Jedná se o plně funkční verzi edice MS SQL Server 2016 Enterprise, která je zdarma na 90 až 180 dní a umožňuje užití aplikace v neprodukčním prostředí. Tato edice neobsahuje licencovaný operační systém [14]. Není možné edici Developer používat pro práci s produkčními daty namísto softwaru s licencovaným operačním systémem. Je vhodná pro testování a předvádění aplikací, které jsou založené na MS SQL Serveru.

2.3.5 Porovnání

Tab. 5. Porovnání funkcí jednotlivých edic [8]

Funkce	Enterprise	Standard	Express	Developer
Počet jader	Neomezeno	24	4	Neomezeno
Využití paměti na instanci	Max. v rámci OS	128 GB	1 GB	Max. v rámci OS
Velikost	524 PB	524 PB	10 GB	524 PB
Produkční použití	Ano	Ano	Ano	Ne
Programovatelnost (T-SQL, XML)	Ano	Ano	Ano	Ano
Management Studio	Ano	Ano	Ano	Ano
Základní vysoká dostupnost	Ano	Ano	Ne	Ano
Pokročilá vysoká dostupnost	Ano	Ne	Ne	Ano
Správa podnikových dat (Master Data Services, Data Quality Services)	Ano	Ne	Ne	Ano
Zabezpečení Always Encrypted	Ano	Ano	Ano	Ano

Transparentní šifrování	Ano	Ne	Ne	Ano
Integrace dat	Ano	Ne	Ne	Ano
Datové tržiště a sklady	Ano	Ano	Ano	Ano
Základní řešení BI (základní tabulkový režim)	Ano	Ano	Ne	Ano
Pokročilé řešení BI (pokročilý tabulkový režim, úložiště Direct Query)	Ano	Ne	Ne	Ano
Stretch Database	Ano	Ano	Ano	Ano

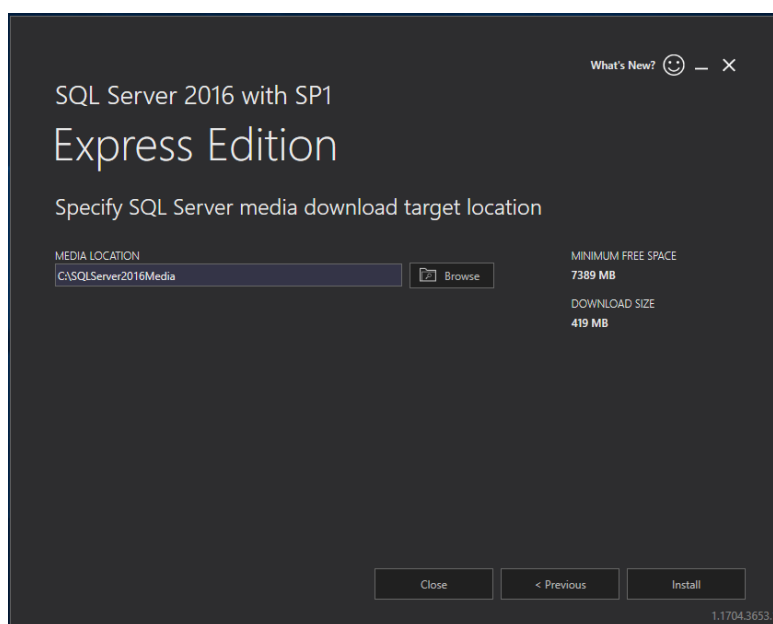
II. PRAKTICKÁ ČÁST

3 PRÁCE V PROSTŘEDÍ MS SQL SERVER 2016

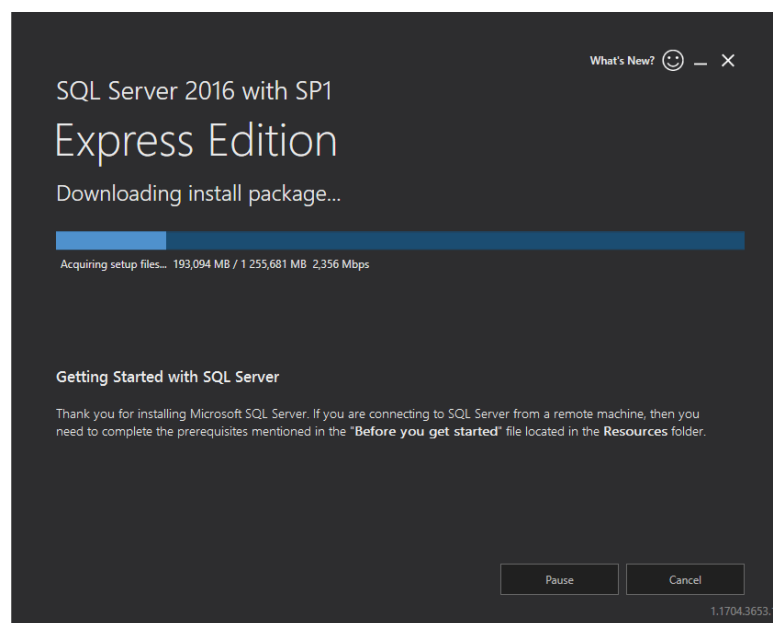
MS SQL Server 2016 ve verzi Express lze bezplatně stáhnout na stránce <https://www.microsoft.com/cs-cz/sql-server/sql-server-downloads>.

3.1 Instalace MS SQL Serveru

Po stažení vhodné verze se automaticky spustí průvodce MS SQL Server 2016. Vybereme složku pro instalaci souborů.

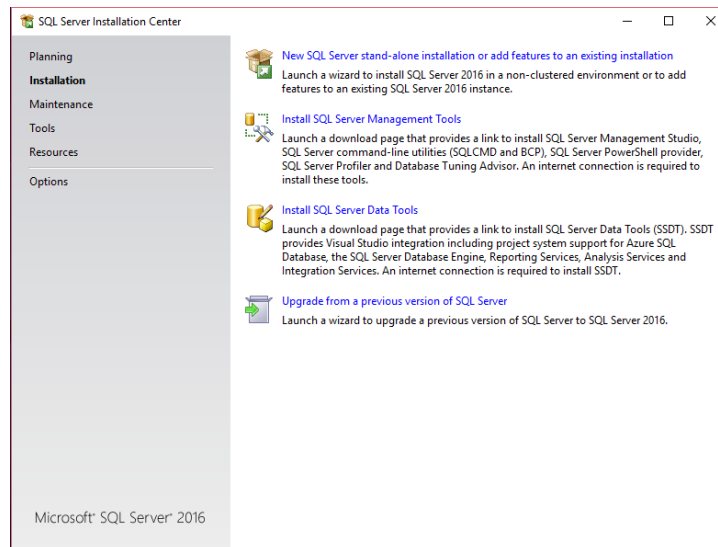


Obr. 8. Výběr složky pro instalaci souborů

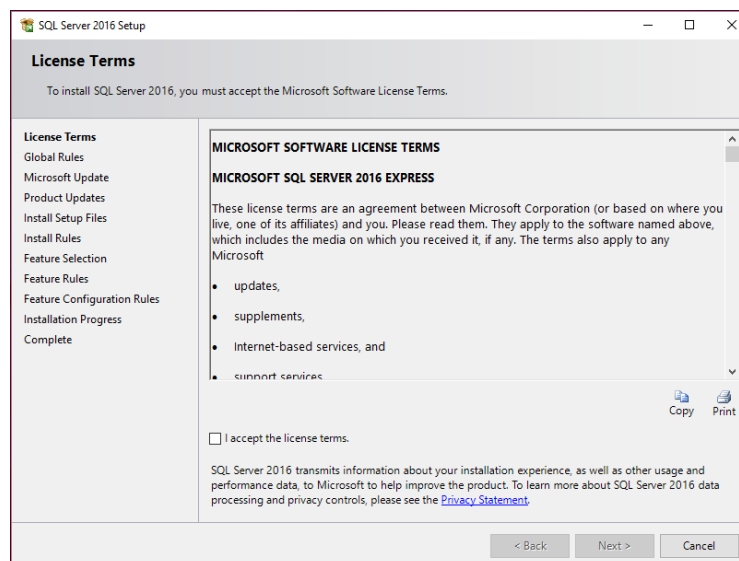


Obr. 9. Průběh stahování

Po stažení se objeví instalační nabídka, kde zvolíme možnost nainstalovat server jako samostatnou instanci – *New SQL Server stand-alone installation or add features to an existing installation*.

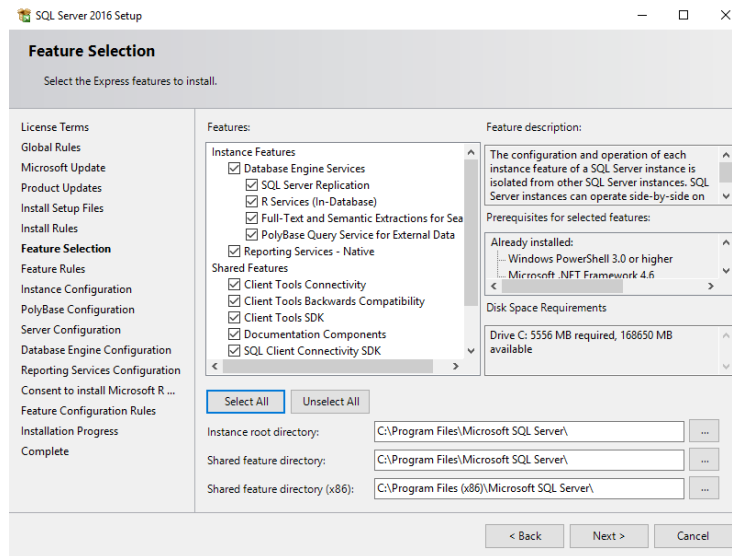


Obr. 10. Instalační nabídka



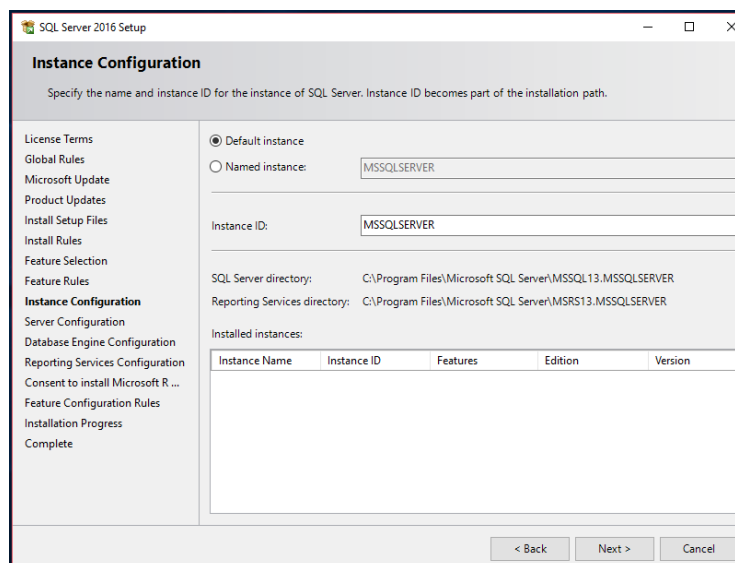
Obr. 11. Licenční podmínky

Po přečtení licenčních podmínek a souhlasu s nimi zaškrtneme tlačítko Next. Objeví se nám Feature Selection, což je nabídka funkcí a modulů, které budou součástí instance. Zvolíme možnost Select All, abychom mohly prozkoumat všechny funkce a pokračujeme opět tlačítkem Next.

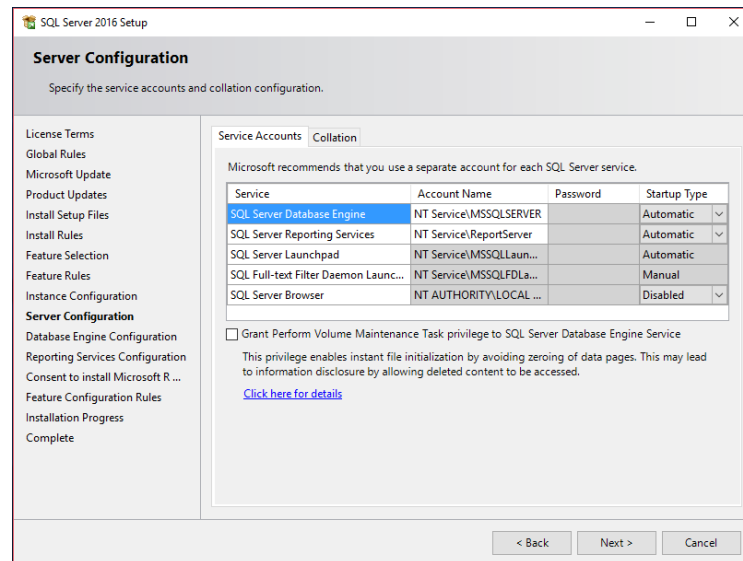


Obr. 12. Výběr modulů

V dalším kroku proběhne kontrola funkcí, které jsme si zvolili a zda mohou být problémy s budoucí instalací. Instance Configuration nám nabídne možnost instalovat instanci jako výchozí nebo pojmenovanou. Výchozí instanci zvolíme tehdy, když v počítači nemáme instalovaný žádný SQL Server. V případě, že již nějakou instanci SQL Serveru nainstalovanou máme, zobrazí se v dolním okně a my jí musíme přiřadit název. V našem případě zvolíme Default Instance. V dalším kroku instalujeme PolyBase jako samostatnou instanci. Server Configuration umožňuje nastavení účtů, pod kterými poběží jednotlivé služby SQL Serveru. Ponecháme doporučené nastavení a klikneme na tlačítko Next.

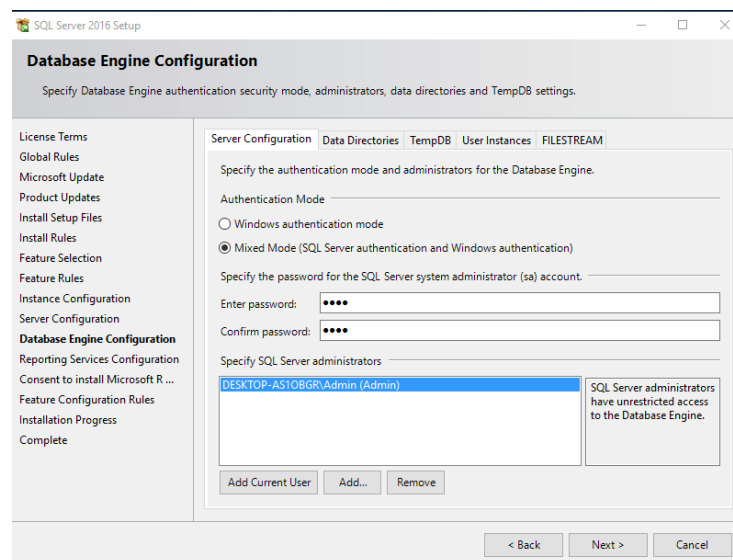


Obr. 13. Pojmenování instance



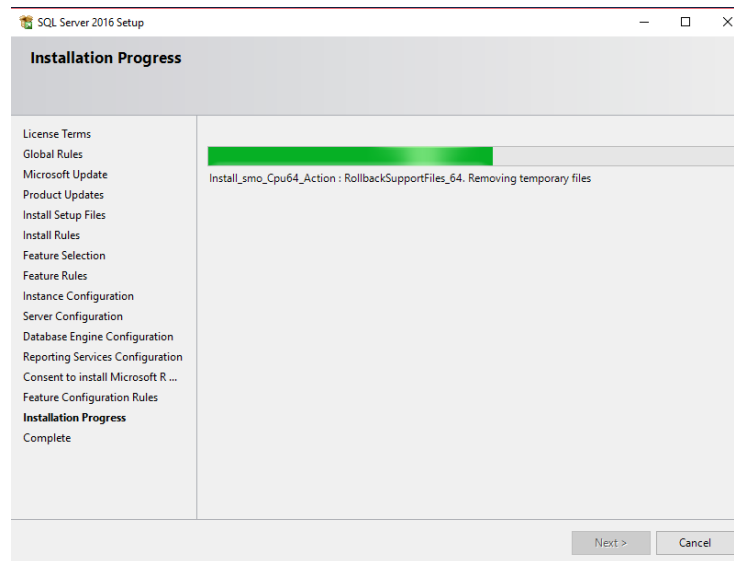
Obr. 14. Automatické nebo manuální spouštění služeb

V tomto kroku konfiguruje Database Engine [27], kde provedeme nastavení autentizace. Pokud zvolíme možnost Windows authentication mode, budeme se moci k instanci přihlašovat pomocí doménových a lokálních účtů. Při zvolení Mixed Mode můžeme nastavit přihlašovací jméno a heslo. Pokud v dolním řádku nenalezneme přihlašovací údaje administrátora, je nutné jej doplnit zvolením Add Current User. Nyní zvolíme heslo, pod kterým se budeme přihlašovat do MS SQL Serveru.



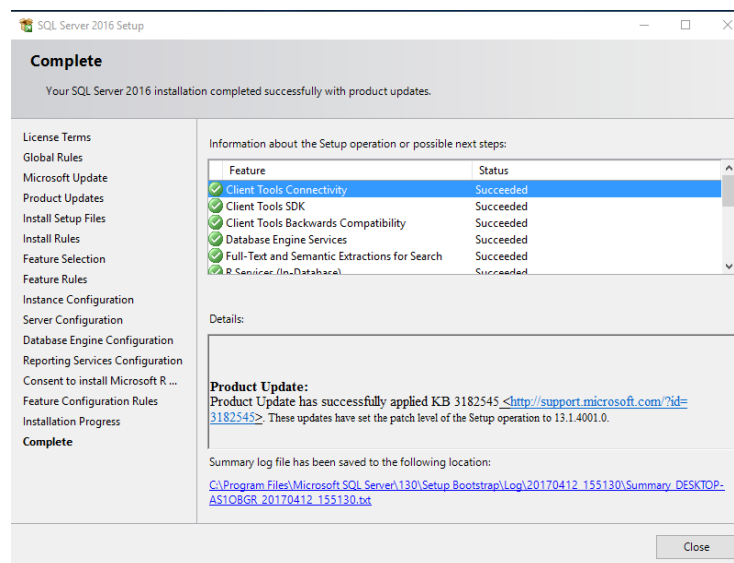
Obr. 15. Uživatelské jméno a heslo

Další krok nám nabídne možnost instalovat, nebo instalovat a konfigurovat. Zvolíme možnost Install only. Pokud bychom konfiguraci potřebovali, je možné ji v budoucnu dokončit. Nyní může začít samotná instalace MS SQL Serveru.



Obr. 16. Průběh instalace

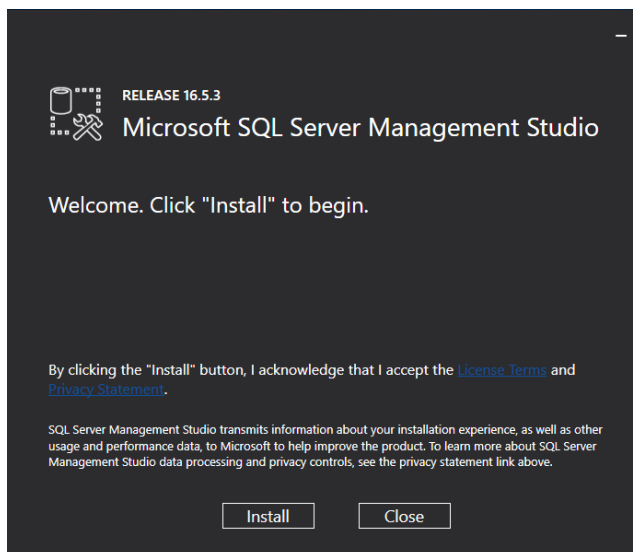
Po dokončení se nám zobrazí report, co vše bylo úspěšně nainstalováno. Můžeme provést vizuální kontrolu, zda nedošlo k selhání u žádného z instalovaných komponentů.



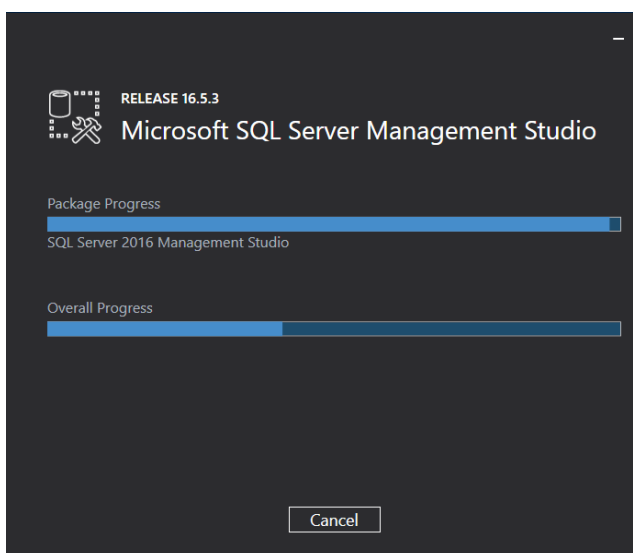
Obr. 17. Report

3.2 Instalace Management Studia

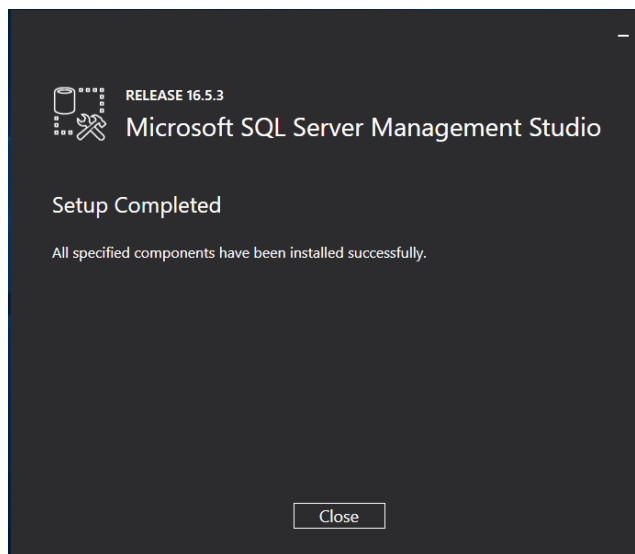
Abychom mohli instanci efektivně využít, je nutné provést instalaci Management Studia, které u této verze není součástí instalačního procesu. Důvodem je možnost vydávání nových verzí Management Studia nezávisle na vydání SQL Serveru.



Obr. 18. Instalace Management Studia

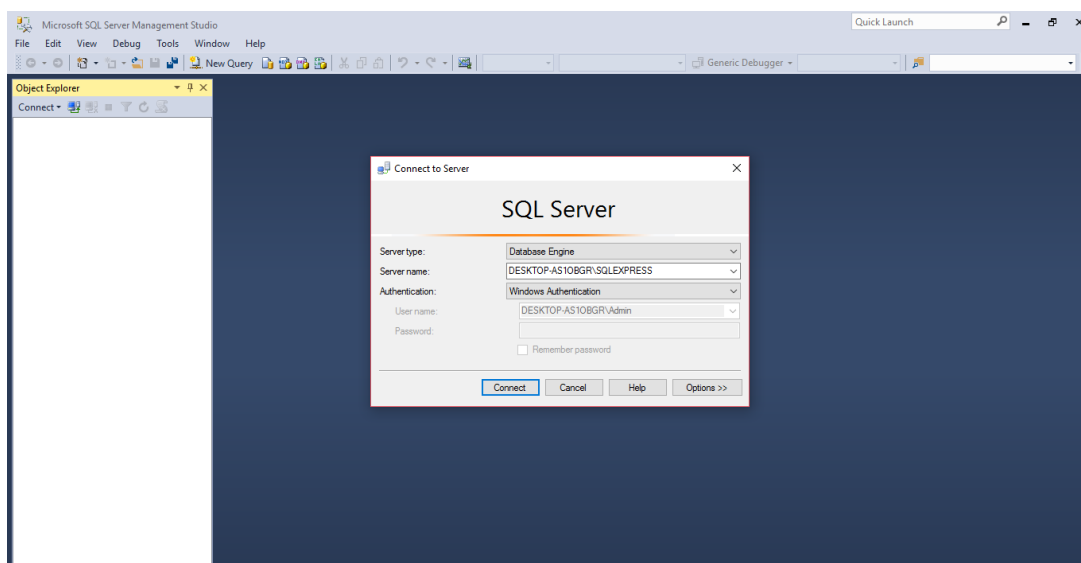


Obr. 19. Průběh instalace Management Studia



Obr. 20. Dokončení instalace Management Studia

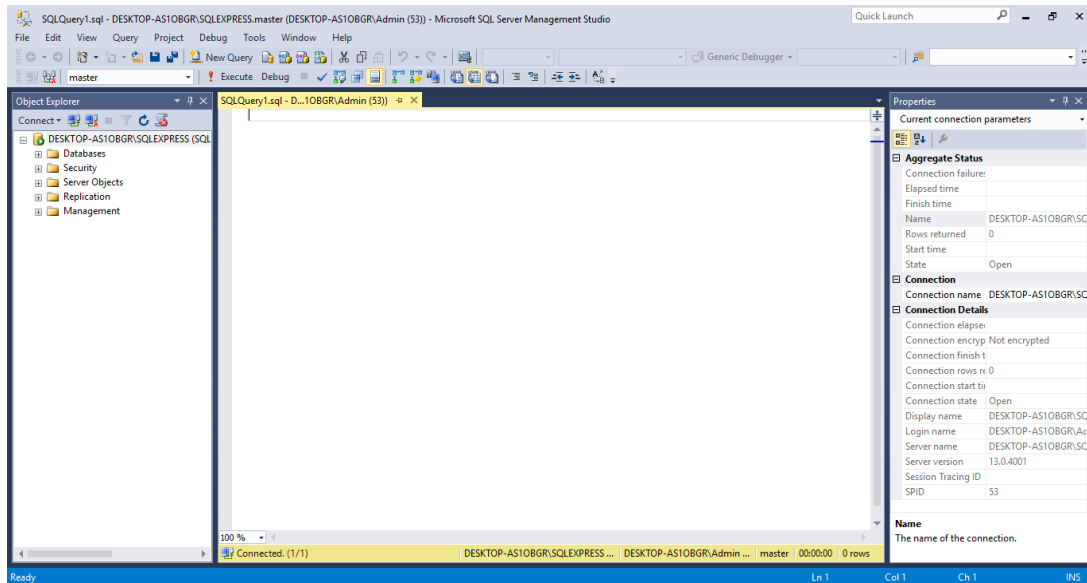
Po úspěšné instalaci spustíme program a zobrazí se nám přihlašovací dialog. Zde se přihlásíme buď pomocí doménového účtu, nebo pomocí přihlašovacích údajů, které jsme si zvolili v průběhu instalace.



Obr. 21. Přihlašovací dialog

3.3 Základní operace pro tvorbu databáze

Po zadání přihlašovacích údajů zvolíme tlačítko New Query, nebo zkratku Ctrl+N a spustí se nám příkazový řádek. Dále určíme dostupnou databázi, ve které chceme pracovat, v tomhle případě zvolíme master.



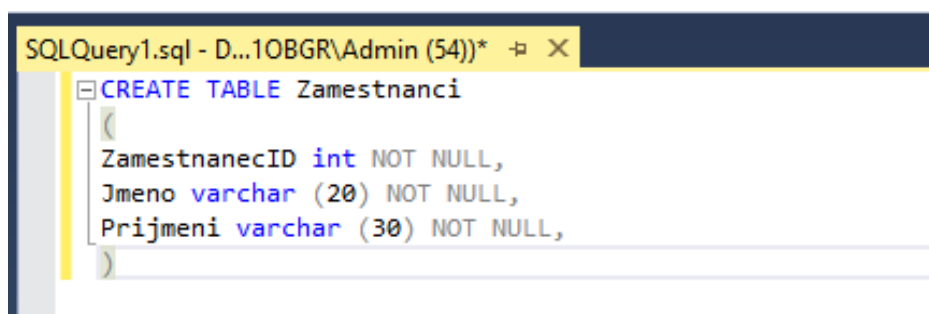
Obr. 22. Pracovní prostředí MS SQL Server 2016

3.3.1 Vytvoření databáze

Pro vytvoření databáze slouží příkaz `CREATE DATABASE`, vyžaduje název databáze ihned za příkazem: `CREATE DATABASE NazevDatabaze`.

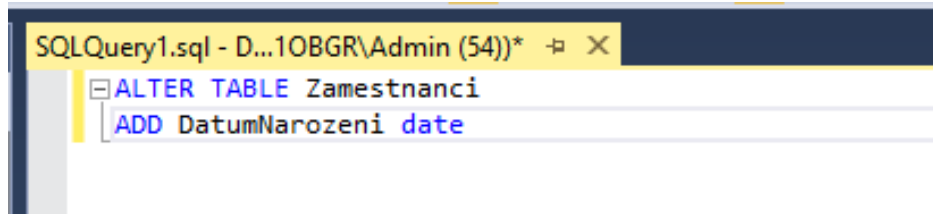
3.3.2 Vytvoření a změna tabulky

Než začneme tabulku vytvářet, je nutné nastavit databázi, ve které chceme pracovat. Nastavení provedeme ručně v menu, nebo pomocí příkazu `USE: USE NazevDatabaze`. Tabulku vytvoříme pomocí příkazu `CREATE TABLE`. Tabulka obsahuje sloupce, kterým musíme přiřadit název, datový typ a integritní omezení. Název musí být jedinečný a bez diakritiky.



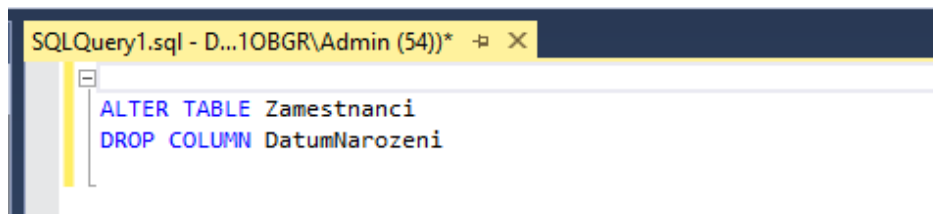
Obr. 23. Vytvoření tabulky

Pro úpravu dat v tabulce slouží příkaz `ALTER TABLE`. Můžeme přidat sloupce do tabulky, odstranit sloupce nebo například změnit datový typ sloupce.



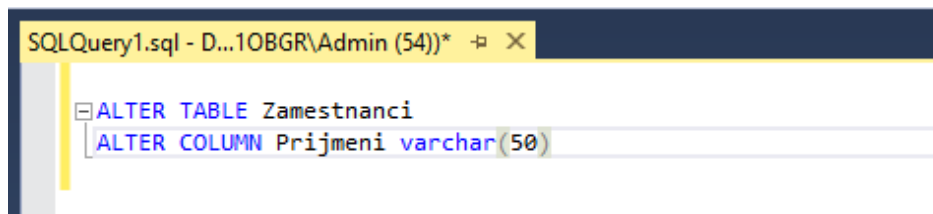
```
SQLQuery1.sql - D...1OBGR\Admin (54))*  X  
ALTER TABLE Zamestnanci  
ADD DatumNarozeni date
```

Obr. 24. Přidání sloupce do tabulky



```
SQLQuery1.sql - D...1OBGR\Admin (54))*  X  
ALTER TABLE Zamestnanci  
DROP COLUMN DatumNarozeni
```

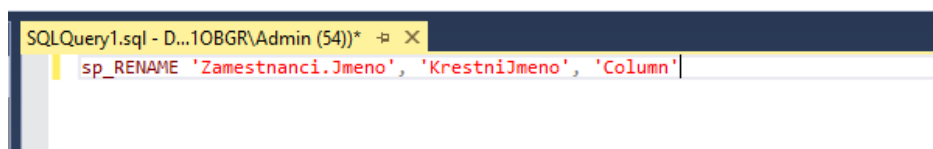
Obr. 25. Odstranění sloupce z tabulky



```
SQLQuery1.sql - D...1OBGR\Admin (54))*  X  
ALTER TABLE Zamestnanci  
ALTER COLUMN Prijmeni varchar(50)
```

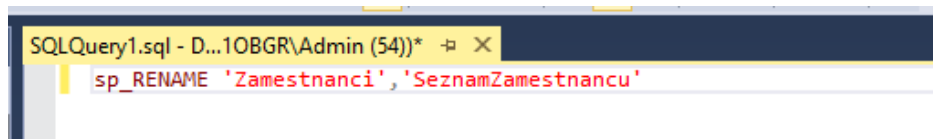
Obr. 26. Změna datového typu sloupce

Změna tabulky může vyžadovat i přejmenování názvu tabulky nebo názvu sloupce v tabulce. K tomu slouží příkaz `sp_RENAME`. Pro přejmenování sloupce musíme uvést název tabulky a původní název sloupce a poté až nový název sloupce. Tabulku smaže příkaz `DROP`.



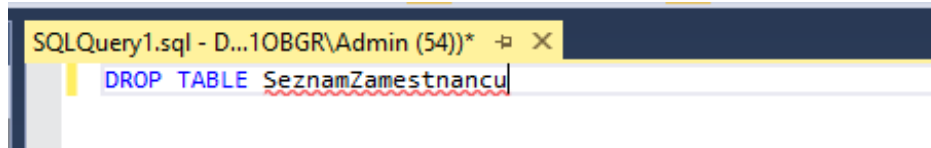
```
SQLQuery1.sql - D...1OBGR\Admin (54))*  X  
sp_RENAME 'Zamestnanci.Jmeno', 'KrestniJmeno', 'Column'
```

Obr. 27. Změna názvu sloupce



```
SQLQuery1.sql - D...1OBGR\Admin (54))*  X  
sp_RENAME 'Zamestnanci', 'SeznamZamestnancu'
```

Obr. 28. Změna názvu tabulky

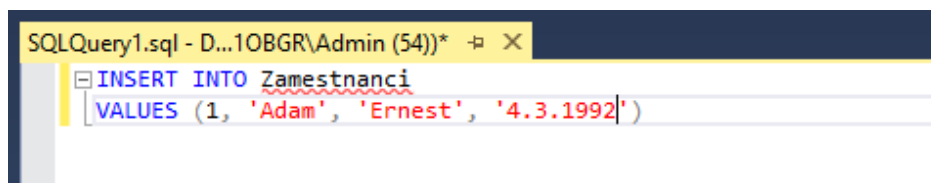


```
SQLQuery1.sql - D...1OBGR\Admin (54))* -p X  
DROP TABLE SeznamZamestnancu
```

Obr. 29. Odstranění tabulky

3.3.3 Vkládání dat do tabulky

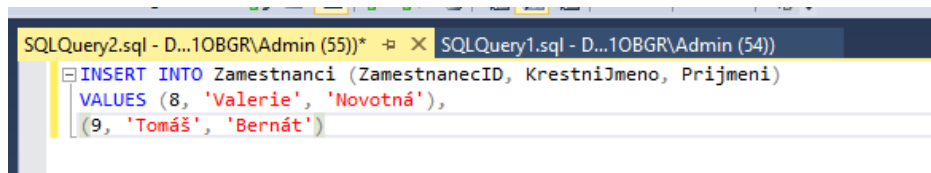
Pro vložení záznamů do tabulky použijeme příkaz INSERT INTO. Musíme rozlišit, zda zadáváme data do všech sloupců nebo jen do některých sloupců v tabulce. Pokud vkládáme data do všech sloupců v tabulce, musíme dbát na to, abychom dodrželi pořadí a správné datové typy, jaké byly použity při vytvoření tabulky.



```
SQLQuery1.sql - D...1OBGR\Admin (54))* -p X  
INSERT INTO Zamestnanci  
VALUES (1, 'Adam', 'Ernest', '4.3.1992')
```

Obr. 30. Vkládání záznamů do všech sloupců tabulky

Pokud vkládáme data jen do určitých sloupců v tabulce, je nutné vypsát názvy sloupců, do kterých chceme data vkládat.



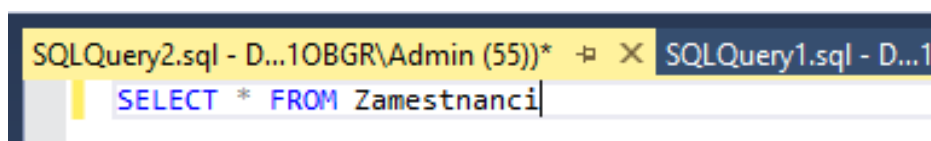
```
SQLQuery2.sql - D...1OBGR\Admin (55))* -p X SQLQuery1.sql - D...1OBGR\Admin (54)  
INSERT INTO Zamestnanci (ZamestnanecID, KrestniJmeno, Prijmeni)  
VALUES (8, 'Valerie', 'Novotná'),  
(9, 'Tomáš', 'Bernát')
```

Obr. 31. Vkládání záznamů do vybraných sloupců tabulky

Aktualizaci vložených dat provádíme pomocí příkazu UPDATE.

3.3.4 Zobrazení vložených dat

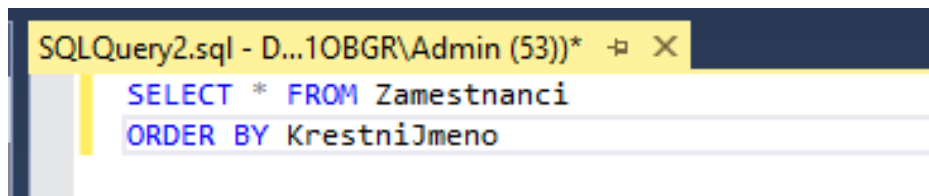
Data zobrazíme pomocí příkazu SELECT. Můžeme zobrazit všechna data v tabulce nebo jen vybrané sloupce. Výčet sloupců nahrazuje znak.



```
SQLQuery2.sql - D...1OBGR\Admin (55))* -p X SQLQuery1.sql - D...1OBGR\Admin (54)  
SELECT * FROM Zamestnanci
```

Obr. 32. Zobrazení dat

Zobrazená data můžeme seřadit pomocí klauzule ORDER BY. Řadit můžeme sestupně (DESC) nebo vzestupně (ASC) dle libovolného sloupce, bez ohledu na to, zda je daný sloupec zobrazen [15]. Pro základní vzestupné třídění je klauzule ASC automatická a zpravidla se neuvádí.

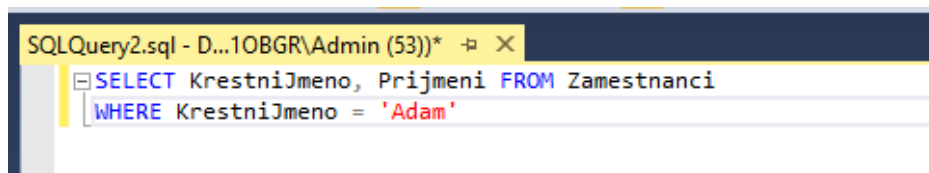


```
SQLQuery2.sql - D...10BGR\Admin (53))* -+ X
SELECT * FROM Zamestnanci
ORDER BY KrestniJmeno
```

Obr. 33. Seřazení zobrazených dat

3.3.5 Zobrazení vložených dat s podmínkou

Pro zobrazení dat můžeme použít několik podmínek, dle kterých se vložená data budou zobrazovat. Podmínky určujeme pomocí klauzule WHERE.



```
SQLQuery2.sql - D...10BGR\Admin (53))* -+ X
SELECT KrestniJmeno, Prijmeni FROM Zamestnanci
WHERE KrestniJmeno = 'Adam'
```

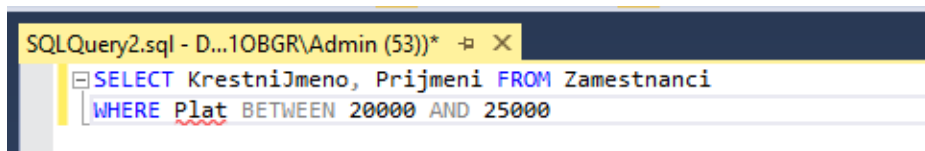
Obr. 34. Zobrazení vložených dat s podmínkou

Je možné zobrazit například zaměstnance s určitým jménem, bydlištěm nebo s určitou výší platu. V podmínce je možné využít několik operátorů.

Tab. 6. Využití operátorů v podmínce WHERE [15]

Operátor	Popis
=	Rovná se
>	Je větší než
<	Je menší než
>=	Je větší nebo rovno
<=	Je menší nebo rovno
!= nebo <>	Nerovná se
!>	Ne více než
!<	Ne méně než
OR	Alespoň jedna podmínka je splněná
AND	Všechny podmínky jsou splněny

NOT	Podmínka není splněna
IS NOT NULL	Hodnota není nulová
IS NULL	Hodnota je nulová
BETWEEN	V rozmezí
NOT BETWEEN	Mimo rozmezí



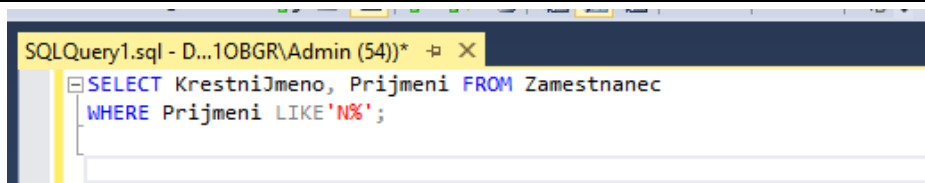
```
SQLQuery2.sql - D:\10BGR\Admin (53)*
SELECT KrestniJmeno, Prijmeni FROM Zamestnanci
WHERE Plat BETWEEN 20000 AND 25000
```

Obr. 35. Zobrazení vložených v určitém rozpětí

Dalšími druhy operátorů jsou LIKE a NOT LIKE, které se využívají pro porovnání řetězců v širším měřítku [15]. Umožňují například vypsání všech zaměstnanců, jejich příjmení začíná na určité písmeno. Rozlišujeme speciální znaky pro vyhledávání podřetězců.

Tab. 7. Speciální znaky pro vyhledávání

Operátor	Popis
[AD]%	Vyhledá slova, která začínají na A nebo D.
[A-D]%	Vyhledá slova, která začínají na A, B, C nebo D.
[4-8]	Hledá v intervalu 4 – 8 (4, 5, 6, 7, 8).
[^4-8]	Hledá vše mimo interval 4 – 8 (0, 1, 2, 3, 9).



```
SQLQuery1.sql - D:\10BGR\Admin (54)*
SELECT KrestniJmeno, Prijmeni FROM Zamestnanec
WHERE Prijmeni LIKE 'N%';
```

Obr. 36. Použití speciálních znaků pro vyhledávání

Výslednou řadu vypsáných záznamů můžeme omezit jen na jejich určitý počet pomocí klauzule TOP [28]. Tato klauzule vrací prvních několik záznamů nebo prvních několik procent záznamů. Pokud chceme vypsát prvních několik procent záznamů, použijeme klauzuli PERCENT. Klauzule WITH TIES [30] je založena na procentuální hodnotě a slouží k zobrazení všech řádků, pokud je procentuální hranice určena na řádek uvnitř skupiny řádků s identickými hodnotami. Obě klauzule je nutné dát do hranatých závorek.

```
SQLQuery1.sql - D...1OBGR\Admin (54))* -p X
SELECT TOP 5 * FROM Zamestnanec
SELECT TOP 5 * FROM Zamestnanec [PERCENT]
SELECT TOP 5 * FROM Zamestnanec [WITH TIES]
```

Obr. 37. Omezení počtu záznamů ve výsledné řadě

3.3.6 Spojování tabulek

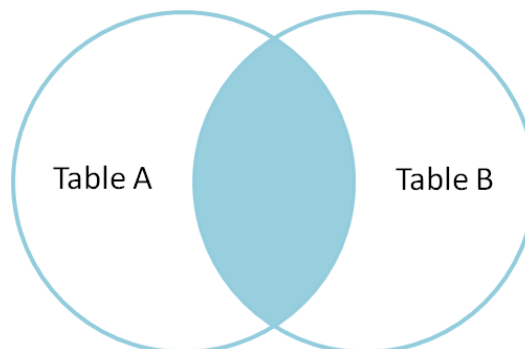
Tabulky spojujeme pomocí příkazu JOIN [20]. Slouží ke spojení výsledku dotazu z více tabulek. Rozlišujeme tři základní druhy spojování tabulek: křížové spojení, vnitřní spojení a vnější spojení. Vnější spojení dále můžeme rozdělit na levé a pravé.

Křížové spojení

Jedná se o nejjednodušší typ spojení. Uskutečníme ho pomocí příkazu CROSS JOIN [20]. Znamená spojení každého záznamu z první tabulky s každým záznamem z tabulky druhé. Provádí se tzv. kartézský součin.

Vnitřní spojení

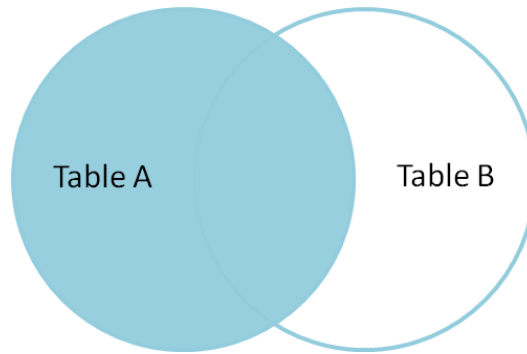
V praxi je to nejvyužívanější typ spojení. Používáme příkaz INNER JOIN [20], který z obou spojovaných tabulek vrátí takové záznamy, ve kterých je obsažena stejná hodnota.



Obr. 38. Vnitřní spojení

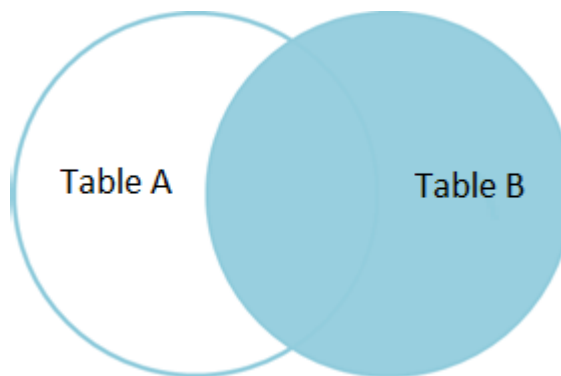
Vnější spojení

Můžeme spojit všechny řádky z jedné tabulky nebo z obou tabulek. Rozlišujeme tři varianty. Příkaz LEFT OUTER JOIN [20] znamená, že každý záznam z první tabulky se musí spojit s některým záznamem z tabulky druhé. Pokud není v tabulce nalezen vyhovující záznam, spojí se se záznamem NULL.



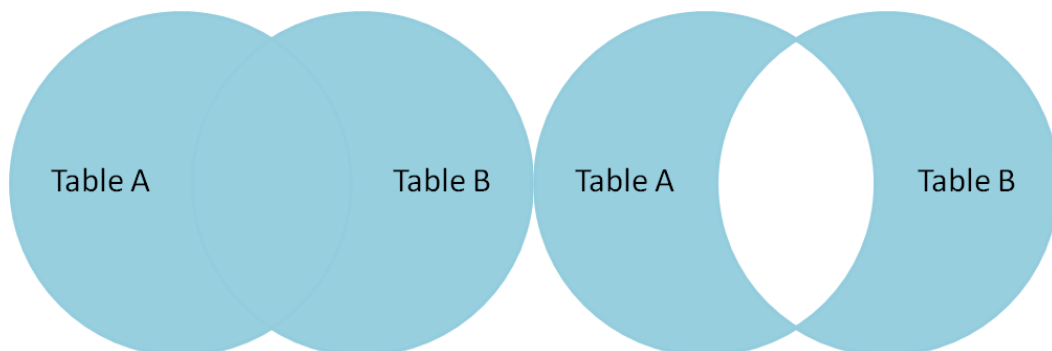
Obr. 39. Vnější spojení *LEFT OUTER JOIN*

Příkaz *RIGHT OUTER JOIN* [20] znamená, že každý záznam z druhé tabulky se musí spojit s některým záznamem z tabulky první. Pokud není v tabulce nalezen vyhovující záznam, spojí se se záznamem *NULL*.



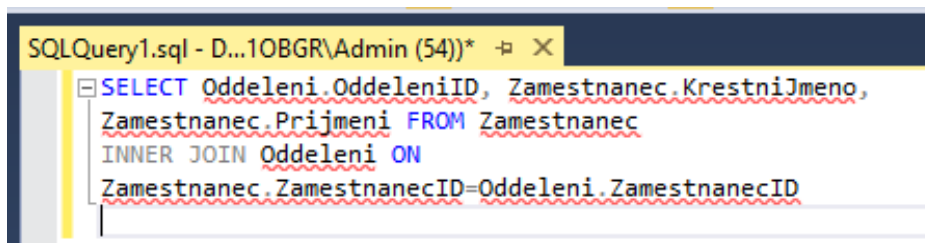
Obr. 40. Vnější spojení *RIGHT OUTER JOIN*

Příkaz *FULL OUTER JOIN* [20] znamená, že každý záznam z každé tabulky musí být spojen s některým záznamem z jiné tabulky. Pokud není v tabulce nalezen vyhovující záznam, spojí se se záznamem *NULL*.



Obr. 41. Vnější spojení *FULL OUTER JOIN*

Používá se v případě, že máme první tabulku, kde najdeme seznam zaměstnanců (ZamestnanecID, KrestniJmeno, Prijmeni, OddeleniID) a druhou tabulku, která obsahuje oddělení (OddeleniID, Nazev, TelefonniCislo, ZamestnanecID). Pro vypsání informací o oddělení a jména zaměstnance, který na daném oddělení pracuje, je potřeba využít příkaz JOIN.

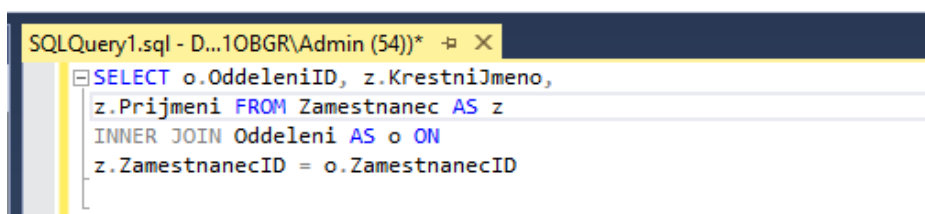


```
SQLQuery1.sql - D...1OBGR\Admin (54))*
SELECT Oddeleni.OddeleniID, Zamestnanec.KrestniJmeno,
Zamestnanec.Prijmeni FROM Zamestnanec
INNER JOIN Oddeleni ON
Zamestnanec.ZamestnanecID=Oddeleni.ZamestnanecID
```

Obr. 42. Spojování tabulek

ALIAS

Alias [31] je symbolické pojmenování tabulky nebo sloupce pomocí klíčového slova AS, které není povinné. Jedná se o složený řetězec z písmen, čísel a podtržitek. Slouží pro pojmenování dočasného sloupce. Například pokud chceme vypsát plat zaměstnanců v cizí měně, můžeme tento sloupec pojmenovat pomocí aliasu. Pokud v dotazu alias nepoužijeme, ve výstupu se zobrazí název sloupce, jako je v tabulce, nebo název znázorňující aritmetickou operaci.



```
SQLQuery1.sql - D...1OBGR\Admin (54))*
SELECT o.OddeleniID, z.KrestniJmeno,
z.Prijmeni FROM Zamestnanec AS z
INNER JOIN Oddeleni AS o ON
z.ZamestnanecID = o.ZamestnanecID
```

Obr. 43. Použití aliasů

3.3.7 Agregáční funkce

Agregáční funkce slouží pro slučování dat z několika řádků do jednoho výsledku [6]. Dochází k využití různých matematických operací. Využití funkcí je možné například pro získání informací o nejnižším, nejvyšším či průměrném platu.

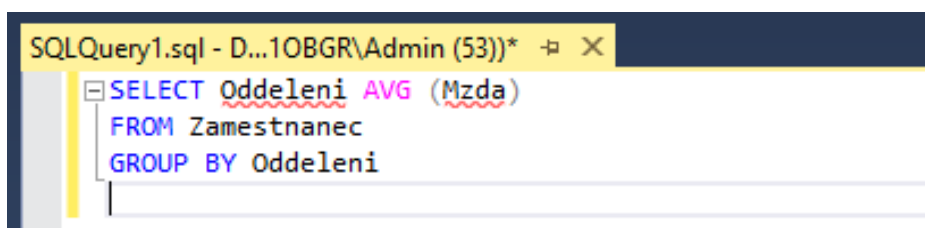
Tab. 8. Agregáční funkce

Funkce	Popis
SUM	Součet numerických hodnot ve sloupci.

MIN	Minimální hodnota ve sloupci.
MAX	Maximální hodnota ve sloupci.
COUNT	Počet hodnot ve sloupci.
AVG	Aritmetický průměr numerických hodnot ve sloupci.

Rozlišujeme agregaci bez klíče a s klíčem. Agregace bez klíče se používá v případě, že výchozí tabulka obsahuje úplná data a není nutné rozšiřovat jednotlivé řádky. Výsledkem bude tabulka, která obsahuje právě jeden řádek.

K slučování dat je možné rozšířit příkaz SELECT o klauzuli GROUP BY případně HAVING. Jedná se o agregaci s klíčem. Klauzule GROUP BY [21] slučuje záznamy, které mají identické hodnoty do jednoho záznamu. Klauzule HAVING [22] je podmínka pro skupiny. Určuje, které záznamy budou zobrazeny. Seskupovat můžeme dle hodnot libovolných sloupců.



```
SQLQuery1.sql - D...10BGR\Admin (53))*
SELECT Oddeleni AVG (Mzda)
FROM Zamestnanec
GROUP BY Oddeleni
```

Obr. 44. Agregací funkce AVG

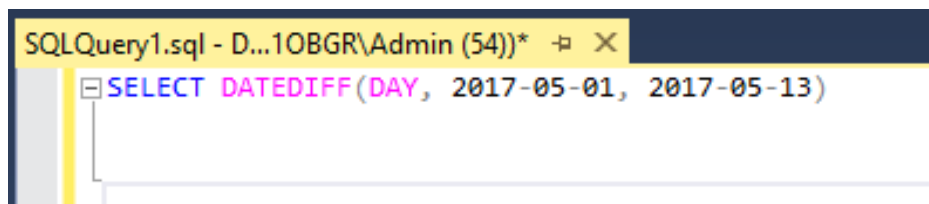
3.3.8 Práce s datem a časem

Pro práci s datem a časem v pracovním prostředí MS SQL Server používáme následující funkce. Funkce vždy spojujeme s příkazem SELECT.

Tab. 9. Operátory pro práci s datem a časem

Funkce	Popis
GETDATE()	Vrátí aktuální datum a čas.
YEAR (sloupec)	Vrátí z uložené procedury pouze údaj o roku.
MONTH (sloupec)	Vrátí z uložené procedury pouze údaj o měsíci.

DAY (sloupec)	Vrátí z uložené procedury pouze údaj o dni.
DATEPART (co, datum)	Vrací část z data (dny, minuty, sekundy).
DATEADD (typ, x, sloupec)	Vrátí hodnotu sloupce zvýšenou o x jednotek. Jako typ se používá SECOND, MINUTE, HOUR, DAY, MONTH, YEAR.
DATEDIFF (typ, x, y)	Vrací počet jednotek mezi dvěma daty (x, y). Jako typ se používá SECOND, MINUTE, HOUR, DAY, MONTH, YEAR.



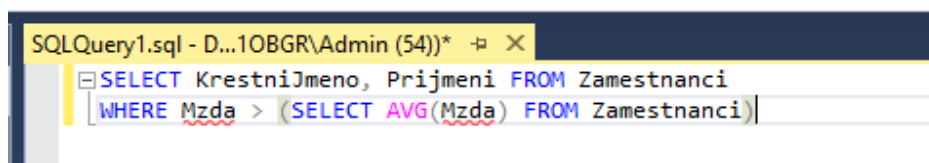
```
SQLQuery1.sql - D...1OBGR\Admin (54)*
SELECT DATEDIFF(DAY, 2017-05-01, 2017-05-13)
```

Obr. 45. Vypsání počtu dnů mezi dvěma daty

3.3.9 Vnořené dotazy

Jedná se o příkazy, které obsahují další příkaz SELECT. Slouží v případě, kdy na základě jedné informace zjišťujeme informace další [23]. Například pro zjištění všech zaměstnanců, kteří mají nadprůměrný plat.

Pomocí klasického SQL dotazu nejprve zjistíme, jaká je výše průměrného platu a následně použijeme příkaz, který vypíše seznam všech zaměstnanců, kteří jej mají nadprůměrný. Tyto dva kroky můžeme spojit v jeden pomocí jednoduchého vnořené dotazu.



```
SQLQuery1.sql - D...1OBGR\Admin (54)*
SELECT KrestniJmeno, Prijmeni FROM Zamestnanci
WHERE Mzda > (SELECT AVG(Mzda) FROM Zamestnanci)
```

Obr. 46. Jednoduchý vnořený dotaz

Operátory

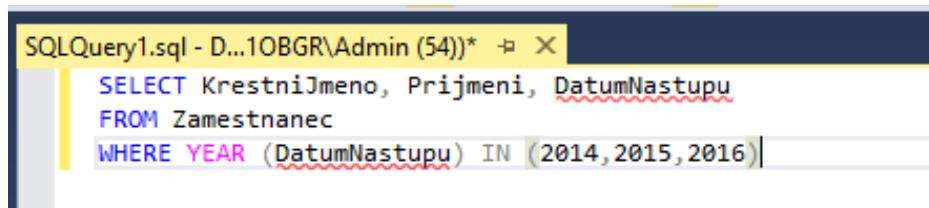
Vnořené dotazy umožňují rozšíření o další operátory, které se aplikují z levé strany na sloupec a ze strany pravé na vnořený dotaz vracejí více hodnot.

Tab. 10. Operátory [23]

Operátor	Popis
IN	Porovnává, zda se hodnota v levém sloupci vyskytuje mezi hodnotami vrácených vnořeným dotazem.
ANY/SOME	Určují, zda se hodnota vztahuje na alespoň jednu z hodnot vrácených vnořeným dotazem.
ALL	Aplikuje relaci na všechny hodnoty vrácené vnořeným dotazem.

Použití operátorů

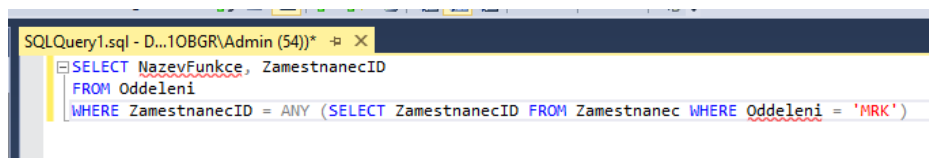
Pomocí operátoru IN můžeme v databázi vypsat zaměstnance, kteří nastoupili v letech 2014 – 2016.



```
SQLQuery1.sql - D...1OBGR\Admin (54))*
SELECT KrestniJmeno, Prijmeni, DatumNastupu
FROM Zamestnanec
WHERE YEAR (DatumNastupu) IN (2014, 2015, 2016)
```

Obr. 47. Použití operátoru IN

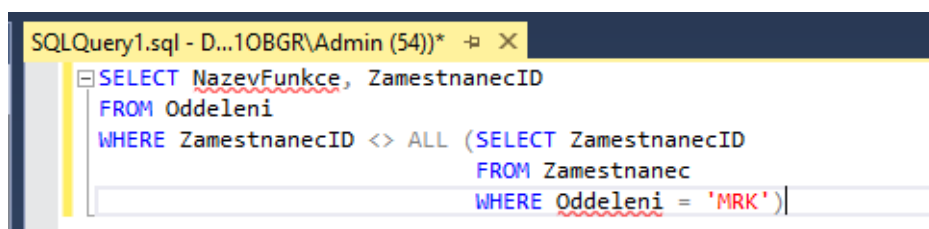
Chceme-li vypsat seznam funkcí, které vykonává libovolný zaměstnanec na určitém oddělení použijeme operátor ANY.



```
SQLQuery1.sql - D...1OBGR\Admin (54))*
SELECT NazevFunkce, ZamestnanecID
FROM Oddeleni
WHERE ZamestnanecID = ANY (SELECT ZamestnanecID FROM Zamestnanec WHERE Oddeleni = 'MRK')
```

Obr. 48. Použití operátoru ANY

Pro vypsání seznamu funkcí, který vykonává libovolný zaměstnanec, který nepracuje na daném oddělení (například externí zaměstnanec), použijeme operátor ALL.



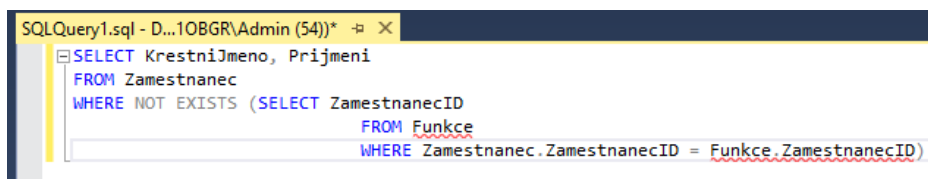
```
SQLQuery1.sql - D...1OBGR\Admin (54))*
SELECT NazevFunkce, ZamestnanecID
FROM Oddeleni
WHERE ZamestnanecID <> ALL (SELECT ZamestnanecID
FROM Zamestnanec
WHERE Oddeleni = 'MRK')
```

Obr. 49. Použití operátoru ALL

Operátor EXISTS

Operátor vrací pravdivost hodnoty [24], na základě hodnot vrácených vnořených dotazem. Operátor vrátí hodnotu TRUE, pokud vnořený dotaz vrátí alespoň jeden řádek. Tento dotaz se vyhodnocuje pro každý řádek samostatně.

Například, chceme-li zkontrolovat, zda u každého zaměstnance máme vyplněnou funkci, kterou vykonává. Vypíšeme tedy seznam zaměstnanců, u nichž nemáme zadanou funkci.

A screenshot of a SQL query editor window titled 'SQLQuery1.sql - D...1OBGR\Admin (54)*'. The query text is:

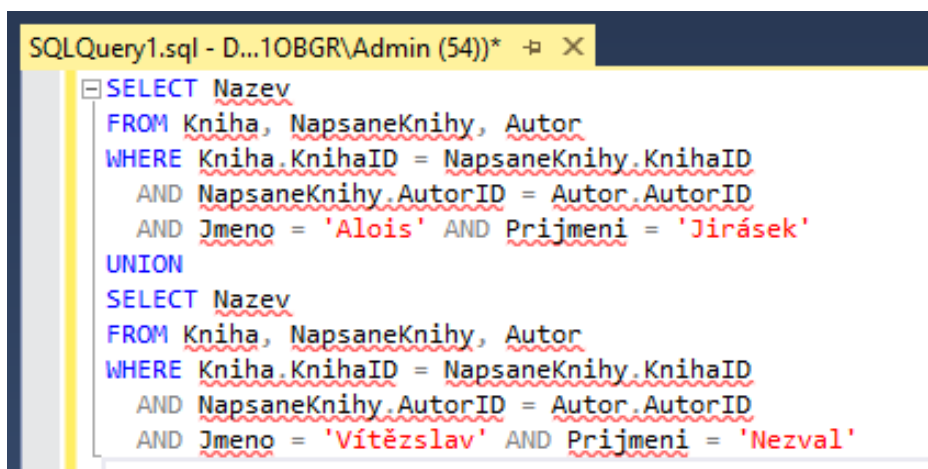
```
SELECT KrestniJmeno, Prijmeni
FROM Zamestnanec
WHERE NOT EXISTS (SELECT ZamestnanecID
                  FROM Funkce
                  WHERE Zamestnanec.ZamestnanecID = Funkce.ZamestnanecID)
```

Obr. 50. Použití operátoru EXISTS

3.3.10 Skládání dotazů

S výstupními daty je možné nadále pracovat [25], provést sjednocení, průnik nebo rozdíl. Pro skládání dotazů používáme příkazy UNION, INTERSECT a MINUS. Při skládání dotazů je nutné dodržet počet sloupců v jednotlivých dotazech, sloupce musí mít shodný datový typ. Třídít výstupy ze složených dotazů je možné pomocí příkazu ORDER BY, které se používá až na konci dotazu.

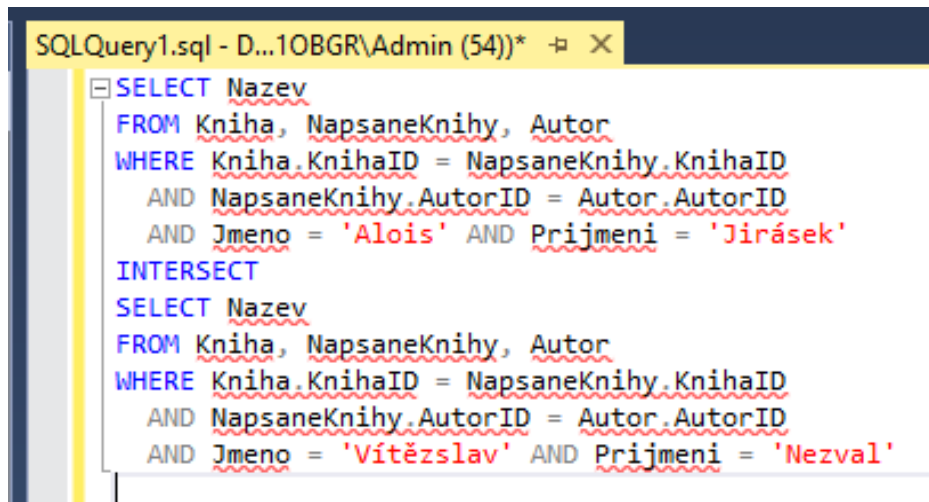
Příkaz UNION [25] slouží pro sjednocení výstupů, za kterým následuje další příkaz SELECT. Používá se například v případě databáze knihovny, že chceme vypsát díla dvou autorů.

A screenshot of a SQL query editor window titled 'SQLQuery1.sql - D...1OBGR\Admin (54)*'. The query text is:

```
SELECT Nazev
FROM Kniha, NapsaneKnihy, Autor
WHERE Kniha.KnihaID = NapsaneKnihy.KnihaID
      AND NapsaneKnihy.AutorID = Autor.AutorID
      AND Jmeno = 'Alois' AND Prijmeni = 'Jirásek'
UNION
SELECT Nazev
FROM Kniha, NapsaneKnihy, Autor
WHERE Kniha.KnihaID = NapsaneKnihy.KnihaID
      AND NapsaneKnihy.AutorID = Autor.AutorID
      AND Jmeno = 'Vítězslav' AND Prijmeni = 'Nezval'
```

Obr. 51. Sjednocení dotazů

Pro průnik dotazů slouží příkaz INTERSECT [25], který má větší využití v praxi. Při použití databáze knihovny, můžeme vypsát všechna díla od Aloise Jiráka a zároveň od Vítězslava Nezvala.



```
SQLQuery1.sql - D...1OBGR\Admin (54)* -# X
SELECT Nazev
FROM Kniha, NapsaneKnihy, Autor
WHERE Kniha.KnihaID = NapsaneKnihy.KnihaID
AND NapsaneKnihy.AutorID = Autor.AutorID
AND Jmeno = 'Alois' AND Prijmeni = 'Jirásek'
INTERSECT
SELECT Nazev
FROM Kniha, NapsaneKnihy, Autor
WHERE Kniha.KnihaID = NapsaneKnihy.KnihaID
AND NapsaneKnihy.AutorID = Autor.AutorID
AND Jmeno = 'Vítězslav' AND Prijmeni = 'Nezval'
```

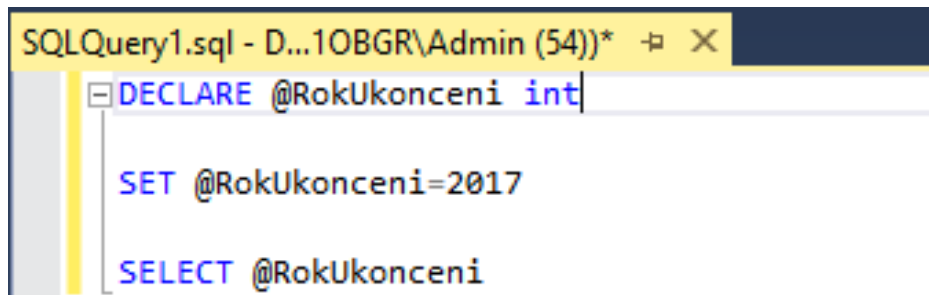
Obr. 52. Průnik dotazů

Klíčové slovo MINUS [25] zajišťuje rozdíl dotazů. V případě databáze knihovny, můžeme rozdíl dotazů použít pro vypsání díla jednoho autora, která zároveň nenapsal druhý autor. Jedná se o díla, která mají více autorů.

3.3.11 Proměnné

MS SQL Server umožňuje vytvářet proměnné [4]. Rozlišujeme tři typy. Lokální proměnné, což jsou systémově nebo uživatelsky definovaná data. Kurzorové proměnné, které zajišťují zpětnou kompatibilitu s objekty ze starších verzí systému. Tabulkové proměnné umožňují ukládání dat ve formě tabulky.

Proměnnou vytvoříme pomocí klauzule DECLARE za níž následuje @Nazev a datový typ. Pokud proměnné nepřidáme žádnou hodnotu, automaticky je hodnota proměnné NULL. Přidat hodnotu můžeme při deklaraci proměnné nebo použitím klíčového slova SET. Proměnnou voláme příkazem SELECT.



```
SQLQuery1.sql - D...1OBGR\Admin (54))* -# X  
DECLARE @RokUkonceni int  
SET @RokUkonceni=2017  
SELECT @RokUkonceni
```

Obr. 53. Proměnné

Podmínky

Podmínkový blok IF, slouží pro jednoduché rozhodování, zda je tvrzení pravdivé nebo nepravdivé. Klíčová slova BEGIN a END [4] používáme k ohraničení bloku. Jako rozhodovací blok můžeme použít klíčové slovo CASE [4]. Používáme jej pro ošetření více stavů proměnné. Vrací pouze jednu hodnotu. Příkaz WHILE [4] používáme pro opakované provádění příkazů. Cyklus probíhá až do splnění podmínky. Příkaz můžeme doplnit klíčovými slovy BREAK a CONTINUE pro řízení logiky uvnitř cyklu [4]. Klíčové slovo BREAK slouží k vyskočení ze smyčky WHILE, zatímco CONTINUE slouží k restartování cyklu. Všechny další příkazy, které následují po klíčovém slovu CONTINUE jsou ignorovány.

3.3.12 Uložené procedury

Procedury [32] jsou části kódu, které uložíme a můžeme je později znovu použít. Je možné použití parametrů a následná modifikace kódu, což je výhoda. Je možné je volat z jiné aplikace a slouží k rychlejšímu zpracování.

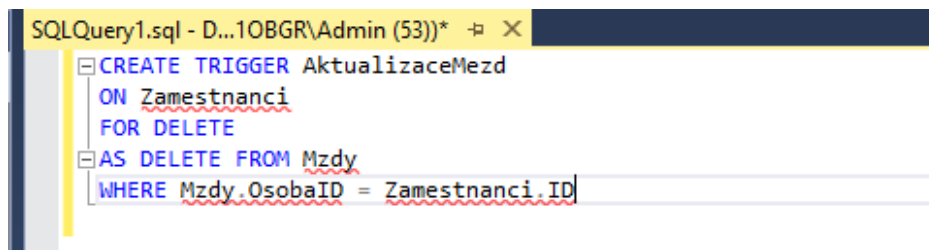
Proceduru vytvoříme pomocí příkazu CREATE PROCEDURE. Po vytvoření se procedura automaticky uloží do záložky Programmability. Pro opakované spuštění procedury použijeme příkaz EXECUTE nebo zkráceně EXEC. Pokud chceme provést nějaké změny již v uložené proceduře, použijeme příkaz ALTER PROCEDURE. Proceduru smažeme pomocí příkazu DROP PROCEDURE.

Procedury mohou obsahovat parametry, které můžeme při zavolání měnit. Parametrů může být několik a jsou zpravidla odděleny čárkou. Jedná se například o parametr pro dané jméno. Parametr může mít zadanou výchozí hodnotu [32], která slouží k tomu, aby se předcházelo chybám při volání procedur bez parametru. Volaná procedura poté proběhne správně, jakoby byl parametr s hodnotou NULL. Parametr může mít zadanou i výstupní hodnotu, která může být jen jedna.

V procedurách můžeme používat komentáře. Rozlišujeme komentář řádkový a blokový. Řádkový komentář se používá pro stručný text, který je dočasně zakomponován do jednoho řádku. Začíná znaky --. Blokový komentář se používá pro rozsáhlejší texty a dočasné zakomponování rozsáhlé části kódu. Komentář je mezi znaky /* a */. Druhy komentářů lze libovolně kombinovat.

3.3.13 Trigger

Trigger neboli spoušť [26], slouží k automatickému spuštění příkazu. Používá se například, pokud z firmy odchází zaměstnanec, rušíme jeho záznamy v tabulce SeznamZamestnancu. Je nutné také zrušit relevantní záznamy v dalších tabulkách, například v tabulce Mzdy požadujeme odstranit záznam, který náležel příslušnému pracovníkovi. Tyto operace jsou prováděny automaticky pomocí příkazu.

The image shows a screenshot of a SQL query editor window. The title bar reads "SQLQuery1.sql - D...1OBGR\Admin (53))* -p X". The main area contains the following SQL code:

```
CREATE TRIGGER AktualizaceMezd
ON Zamestnanci
FOR DELETE
AS DELETE FROM Mzdy
WHERE Mzdy.OsobaID = Zamestnanci.ID
```

The code is color-coded: "CREATE TRIGGER" is blue, "AktualizaceMezd" is black, "ON Zamestnanci" is blue, "FOR DELETE" is blue, "AS DELETE FROM Mzdy" is blue, and "WHERE Mzdy.OsobaID = Zamestnanci.ID" is blue. The table names "Zamestnanci" and "Mzdy" are underlined with red dashed lines.

Obr. 54. Trigger

4 PODKLADY PRO VÝUKU

Informace obsažené v teoretické části práce jsou stručně shrnuty do prezentací, které budou sloužit jako podklady pro přednášejícího. Obsah praktické části práce je využit jako podkladový materiál pro cvičení. Jsou zde popsány příkazy SQL a jejich využití. Každá prezentace je doplněna úkolem, pro ověření pochopení učiva v daném týdnu.

Dalšími podklady pro výuku je 120 testových otázek, které obsahují problematiku teoretické i praktické části probírané v práci. Otázky jsou vhodné pro import do Moodle a byly předány vedoucímu bakalářské práce.

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit učební pomůcku pro výuku databázových systémů pro začátečníky. Využit byl program Microsoft SQL Server 2016, což je nejnovější dostupná verze. Systém je dostupný ke stažení na oficiálních webových stránkách společnosti Microsoft ve zkušební verzi.

V první části bakalářské práce byla zpracována teorie databázových systémů. Byly zde uvedeny základní definice a úvod do problematiky databázových systémů. Byl zde představen systém Microsoft SQL Server, jeho vznik a vývoj. Probrány byly základní funkce a rozdíly mezi jednotlivými verzemi. Tato část je zpracována do prezentací, které jsou v příloze této práce a slouží jako podklady pro přednášejícího.

V úvodu praktické části bakalářské práce je popsán postup pro stažení zkušební verze a popis instalace. V další části jsou popsány základní operace, tvorba příkazů a práce s daty v prostředí Microsoft SQL Server 2016. Tato část je také zpracována do prezentací, které jsou vhodné jako podkladový materiál pro výuku na cvičení. Každá prezentace je doplněna úkoly, kterými si student ověří získané znalosti.

Součástí bakalářské práce je soubor 120 testových otázek, kde jsou obsaženy termíny z teoretické i praktické části. Otázky jsou určeny pro závěrečný test z databázových systémů a jsou vhodné pro import do Moodle. Tento soubor testovacích otázek byl odevzdán vedoucímu práce.

SEZNAM POUŽITÉ LITERATURY

- [1] JORGENSEN, Adam. Professional microsoft sql server 2014 administration. 1st edition. Indianapolis, IN: John Wiley and Sons, 2014, pages cm. ISBN 1118859138.
- [2] BEN-GAN, Itzik, Dejan SARKA a Ron TALMAGE. Querying Microsoft SQL Server 2012: exam 70-461 training kit. Sebastopol, Calif.: Microsoft, c2012, xxx, 704 p. ISBN 0735666059.
- [3] KROENKE, David a David J AUER. Databáze. 1. vyd. Brno: Computer Press, 2015, 496 s. ISBN 978-80-251-4352-0.
- [4] MASOOD-AL-FAROOQ, B. A. SQL Server 2014 Development Essentials. UK.: Packt Publishing, 2014. ISBN 978-1-78217-255-0.
- [5] CORONEL, Carlos a Steven MORRIS. Database systems: design, implementation, and management. 11e [edition]. United States: Course Technology Cengage Learning, 2015, xxvii, 751 pages. ISBN 128519618x.
- [6] OPPEL, Andrew J. *Databáze bez předchozích znalostí: [přůvodce pro samouky]*. Brno: Computer Press, 2006, 319 s. ISBN 80-251-1199-7
- [7] SQL Server 2016 [online]. [cit. 2017-03-15]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2016>
- [8] Edice systému SQL Server 2016 [online]. [cit. 2017-03-15]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions>
- [9] POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. Praha: České vysoké učení technické v Praze, 2013, 265 s. Vysokoškolská učebnice. ISBN 978-80-01-05212-9.
- [10] BORONCZYK, Tim. *MySQL okamžitě*. Brno: Computer Press, 2016, 144 s. ISBN 978-80-251-4737-5.
- [11] *Edice SQL Server 2016 SP1 Enterprise* [online]. [cit. 2017-03-30]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-enterprise>
- [12] *Edice SQL Server 2016 SP1 Standard* [online]. [cit. 2017-03-30]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-standard>
- [13] *Edice SQL Server 2016 SP1 Developer* [online]. [cit. 2017-03-30]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-developers>

- [14] *Edice SQL Server 2016 SPI Express* [online]. [cit. 2017-03-30]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-express>
- [15] CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7
- [16] *Hardware and Software Requirements for Installing SQL Server* [online]. [cit. 2017-04-06]. Dostupné z: <https://docs.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server>
- [17] *Databáze: Databázové modely* [online]. [cit. 2017-04-07]. Dostupné z: <http://www.databaze.chytrak.cz/modely.htm>
- [18] PRICE, Jason. *C#: programování databází*. Praha: Grada, 2005, 623 s. Profesional. ISBN 8024709821.
- [19] *MS SQL Server - historie a vývoj* [online]. [cit. 2017-04-12]. Dostupné z: http://www.cecak.cz/fel/dba/referaty/mssql/historie_a_vyvoj_ms_sql
- [20] *A Visual Explanation of SQL Joins* [online]. [cit. 2017-04-21]. Dostupné z: <https://blog.codinghorror.com/a-visual-explanation-of-sql-joins/>
- [21] *Klauzule GROUP BY* [online]. [cit. 2017-04-25]. Dostupné z: <https://support.office.com/cs-cz/article/Klauzule-GROUP-BY-84eeb766-25d2-4aa1-8eea-002bb65ef3a0>
- [22] *Klauzule HAVING* [online]. [cit. 2017-04-25]. Dostupné z: <https://support.office.com/cs-cz/article/Klauzule-HAVING-64c52dba-5cda-45c5-98b5-bd155a89f02f>
- [23] *SQL – vnořené dotazy* [online]. [cit. 2017-04-25]. Dostupné z: <https://www.interval.cz/clanky/sql-vnorene-dotazy/>
- [24] *SQL – pokročilé dotazy* [online]. [cit. 2017-04-25]. Dostupné z: <https://www.interval.cz/clanky/sql-pokrocile-dotazy/>
- [25] *SQL – skládání dotazů* [online]. [cit. 2017-04-29]. Dostupné z: <https://www.interval.cz/clanky/sql-skladani-dotazu/>
- [26] *SQL – jak na triggeru* [online]. [cit. 2017-05-03]. Dostupné z: <https://www.interval.cz/clanky/sql-jak-na-triggeru/>
- [27] *Instalace SQL Serveru 2016* [online]. [cit. 2017-05-08]. Dostupné z: <http://sqldot.com/sql-server-2016-instalace/>

- [28] *Univerzální stránková procedura v MS SQL* [online]. [cit. 2017-05-08]. Dostupné z: <https://www.interval.cz/clanky/univerzalni-strankovaci-procedura-v-ms-sql/>
- [29] *TOP (Transact-SQL)* [online]. [cit. 2017-05-08]. Dostupné z: [https://technet.microsoft.com/cs-cz/library/ms189463\(v=sql.100\).aspx](https://technet.microsoft.com/cs-cz/library/ms189463(v=sql.100).aspx)
- [30] *Stránka vlastností uložené procedury (ADP)* [online]. [cit. 2017-05-08]. Dostupné z: <https://support.office.com/cs-cz/article/Stránka-vlastností-uložené-procedury-ADP-41249760-437e-4061-9b13-04348047aaba>
- [31] *PROGRAMOVÁNÍ V SQL* [online]. [cit. 2017-05-15]. Dostupné z: http://ucimedatabase.cz/wp-content/uploads/2012/02/CZ_programovani_SQL.pdf
- [32] *14. díl - MS-SQL - Uložené procedury* [online]. [cit. 2017-05-15]. Dostupné z: <https://www.itnetwork.cz/ms-sql/ms-sql-ulozene-procedury>
- [33] *SQL Server 2008 - datové typy* [online]. [cit. 2017-05-16]. Dostupné z: <http://programujte.com/clanek/2010022200-sql-server-2008-datove-typy/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SQL	Structured Query Language
DML	Data Manipulation Language
DDL	Data Definition Language
DCL	Data Control Language
TCL	Transaction Control Language
MS	Microsoft

SEZNAM OBRÁZKŮ

<i>Obr. 1. Komponenty databázového systému</i>	11
<i>Obr. 2. Hierarchický model [17]</i>	12
<i>Obr. 3. Síťový model [17]</i>	12
<i>Obr. 4. Relační model [17]</i>	13
<i>Obr. 5. Relace typu 1:1 [4]</i>	14
<i>Obr. 6. Relace typu 1:N [4]</i>	15
<i>Obr. 7. Relace typu M:N [4]</i>	16
<i>Obr. 8. Výběr složky pro instalaci souborů</i>	27
<i>Obr. 9. Průběh stahování</i>	27
<i>Obr. 10. Instalační nabídka</i>	28
<i>Obr. 11. Licenční podmínky</i>	28
<i>Obr. 12. Výběr modulů</i>	29
<i>Obr. 13. Pojmenování instance</i>	29
<i>Obr. 14. Automatické nebo manuální spouštění služeb</i>	30
<i>Obr. 15. Uživatelské jméno a heslo</i>	30
<i>Obr. 16. Průběh instalace</i>	31
<i>Obr. 17. Report</i>	31
<i>Obr. 18. Instalace Management Studia</i>	32
<i>Obr. 19. Průběh instalace Management Studia</i>	32
<i>Obr. 20. Dokončení instalace Management Studia</i>	33
<i>Obr. 21. Přihlašovací dialog</i>	33
<i>Obr. 22. Pracovní prostředí MS SQL Server 2016</i>	34
<i>Obr. 23. Vytvoření tabulky</i>	34
<i>Obr. 24. Přidání sloupce do tabulky</i>	35
<i>Obr. 25. Odstranění sloupce z tabulky</i>	35
<i>Obr. 26. Změna datového typu sloupce</i>	35
<i>Obr. 27. Změna názvu sloupce</i>	35
<i>Obr. 28. Změna názvu tabulky</i>	35
<i>Obr. 29. Odstranění tabulky</i>	36
<i>Obr. 30. Vkládání záznamů do všech sloupců tabulky</i>	36
<i>Obr. 31. Vkládání záznamů do vybraných sloupců tabulky</i>	36
<i>Obr. 32. Zobrazení dat</i>	36

<i>Obr. 33. Seřazení zobrazených dat</i>	<i>37</i>
<i>Obr. 34. Zobrazení vložených dat s podmínkou.....</i>	<i>37</i>
<i>Obr. 35. Zobrazení vložených v určitém rozpětí</i>	<i>38</i>
<i>Obr. 36. Použití speciálních znaků pro vyhledávání</i>	<i>38</i>
<i>Obr. 37. Omezení počtu záznamů ve výsledné řadě</i>	<i>39</i>
<i>Obr. 38. Vnitřní spojení</i>	<i>39</i>
<i>Obr. 39. Vnější spojení LEFT OUTER JOIN.....</i>	<i>40</i>
<i>Obr. 40. Vnější spojení RIGHT OUTER JOIN</i>	<i>40</i>
<i>Obr. 41. Vnější spojení FULL OUTER JOIN</i>	<i>40</i>
<i>Obr. 42. Spojování tabulek</i>	<i>41</i>
<i>Obr. 43. Použití aliasů</i>	<i>41</i>
<i>Obr. 44. Agregiční funkce AVG</i>	<i>42</i>
<i>Obr. 45. Vypsání počtu dnů mezi dvěma daty.....</i>	<i>43</i>
<i>Obr. 46. Jednoduchý vnořený dotaz</i>	<i>43</i>
<i>Obr. 47. Použití operátoru IN.....</i>	<i>44</i>
<i>Obr. 48. Použití operátoru ANY</i>	<i>44</i>
<i>Obr. 49. Použití operátoru ALL.....</i>	<i>44</i>
<i>Obr. 50. Použití operátoru EXISTS</i>	<i>45</i>
<i>Obr. 51. Sjednocení dotazů.....</i>	<i>45</i>
<i>Obr. 52. Průnik dotazů</i>	<i>46</i>
<i>Obr. 53. Proměnné.....</i>	<i>47</i>
<i>Obr. 54. Trigger.....</i>	<i>48</i>

SEZNAM TABULEK

<i>Tab. 1. Textové datové typy používané v MS SQL Server [10]</i>	18
<i>Tab. 2. Číselné datové typy používané v MS SQL Server [10]</i>	18
<i>Tab. 3. Datové typy pro datum a čas používané v MS SQL Serveru [10]</i>	19
<i>Tab. 4. Požadavky na operační systém [16]</i>	23
<i>Tab. 5. Porovnání funkcí jednotlivých edic [8]</i>	24
<i>Tab. 6. Využití operátory v podmínce WHERE [15]</i>	37
<i>Tab. 7. Speciální znaky pro vyhledávání</i>	38
<i>Tab. 8. Agregáční funkce</i>	41
<i>Tab. 9. Operátory pro práci s datem a časem</i>	42
<i>Tab. 10. Operátory [23]</i>	44

SEZNAM PŘÍLOH

P I SOUBOR VÝUKOVÝCH MATERIÁLŮ NA CD

P II SOUBOR TESTOVÝCH OTÁZEK NA CD

PŘÍLOHA P I: SOUBOR VÝUKOVÝCH MATERIÁLŮ NA CD

Přiložený disk CD obsahuje složku s názvem Přednášky, kde jsou prezentace, které slouží jako podkladový materiál na přednášky a složku z názvem Cvičení, kde jsou prezentace, které slouží jako podkladový materiál na cvičení.

PŘÍLOHA P II: SOUBOR TESTOVÝCH OTÁZEK NA CD

Příložený disk CD obsahuje 120 testových otázek, které jsou vhodné pro import do Moodle. Otázky jsou vhodné pro závěrečný test z předmětu. Testovány jsou znalosti jak z teoretické tak i praktické části. Otázky jsou uzavřené vždy s jednou správnou odpovědí.