

Učební pomůcka v angličtině pro Microsoft SQL Server 2014

Patrik Čordáš

**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: **Patrik Čordáš**
Osobní číslo: **A13146**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Učební pomůcka v angličtině pro Microsoft SQL Server 2014**
Téma anglicky: **Lecture Notes in English for the Microsoft SQL Server 2014**

Zásady pro vypracování:

1. Seznamte se s vývojovým prostředím MS SQL Server 2016.
2. Uveďte nové funkce aktuální edice.
3. Popište jednotlivé edice MS SQL Server 2016 a rozdíly mezi nimi.
4. Zpracujte podklady pro výuku v anglickém jazyce, zahrnující přednášky i cvičení, v rozsahu 14 týdnů.
5. V podkladech vytvořte vzorové příklady v anglickém jazyce i s řešením.
6. Vytvořte 60 testových otázek v anglickém jazyce vhodných k importu do systému MOODLE.

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:

24. února 2017

Termín odevzdání bakalářské práce:

24. května 2017

Ve Zlíně dne 24. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení:

Název bakalářské/diplomové práce:

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 s příjmem – 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

29.5.2017



.....
podpis diplomanta

ABSTRAKT

Cieľom tejto bakalárskej práce je vytvoriť učebnú pomôcku v angličtine pre výuku Microsoft SQL Server 2014. Práca sa v teoretickej časti zaoberá problematikou databázových systémov a v stručnosti opisuje jazyk SQL. Následne podrobne opisuje vývojové prostredie Microsoft SQL server 2016 , najnovšie funkcie, jednotlivé edície a ich rozdiely. Praktická časť sa venuje práci vo vývojovom prostredí Microsoft SQL Server 2014 vo verzii Express. Súčasťou tejto bakalárskej práce sú aj výukové podklady v anglickom jazyku, ktoré zahŕňujú prednášky a cvičenia v rozsahu jedného semestra a 60 testovacích otázok vhodných na import do systému MOODLE.

Kľúčová slova: databázy , SQL ,MS SQL 2016 , MS SQL 2014 Express, T-SQL

ABSTRACT

The aim of this bachelor thesis is to create educational materials for Microsoft SQL Server 2014. Theoretical part of this thesis deals with introduction of database systems and quickly describe SQL language. Next detailed describe development environment of Microsoft SQL Server 2016, newest functions, editions and differences between them. Practical part of this thesis deals with development environment of Microsoft SQL Server 2014 in version Express. Another part of this thesis are educational materials in English language which including presentations for education in length one term. Also include 60 testing questions ready to import into system MOODLE.

Keywords: databases , SQL, MS SQL 2016, MS SQL 2014 Express, T-SQL

Ďakujem svojmu vedúcemu bakalárskej práce Ing. Petrovi Šilhavému, Ph.D., za pomoc pri písaní práce a poskytnuté materiály.

OBSAH

| | |
|--|-----------|
| ÚVOD | 10 |
| I TEORETICKÁ ČÁST | 11 |
| 1 DATABÁZA | 12 |
| 1.1 HISTÓRIA DATABÁZ..... | 12 |
| 1.2 DATABÁZOVÉ MODELY | 12 |
| 1.2.1 Hierarchický model | 13 |
| 1.2.2 Sieťový model | 13 |
| 1.2.3 Relačný model..... | 14 |
| 1.2.4 Objektový model | 14 |
| 1.2.5 Objektovo-relačný model..... | 14 |
| 1.3 ENTITNE – RELAČNÉ MODELOVANIE..... | 15 |
| 1.3.1 Entita | 15 |
| 1.3.2 Relácia..... | 15 |
| 1.3.3 Atribút | 15 |
| 1.3.3.1 Jednoduché a zložené atribúty | 15 |
| 1.3.3.2 Atribúty s jednou alebo viacerými hodnotami..... | 16 |
| 1.3.3.3 Odvodené atribúty..... | 16 |
| 1.3.4 Kľúč relácie..... | 16 |
| 1.3.5 Obmedzenie multiplicity v relácii | 16 |
| 1.3.5.1 Relácia 1:1 | 16 |
| 1.3.5.2 Relácia 1:N | 17 |
| 1.3.5.3 Relácia N:N..... | 17 |
| 1.4 NORMALIZÁCIA | 17 |
| 1.4.1 Prvá normálna forma (1NF) | 17 |
| 1.4.2 Druhá normálna forma (2NF) | 17 |
| 1.4.3 Tretia normálna forma (3NF)..... | 17 |
| 2 SQL | 19 |
| 2.1 HISTÓRIA SQL | 19 |
| 2.2 DDL..... | 19 |
| 2.3 DML | 19 |
| 2.4 DCL..... | 20 |
| 3 MICROSOFT SQL SERVER | 21 |
| 3.1 NOVÉ FUNKCIE V MICROSOFT SQL SERVER 2016..... | 21 |
| 3.1.1 Row Level Security | 21 |
| 3.1.2 Live Query Statistics | 21 |
| 3.1.3 Polybase | 22 |
| 3.1.4 Podpora JSON..... | 22 |
| 3.1.5 Dynamic Data Masking..... | 22 |
| 3.1.6 Temporal Tables..... | 23 |
| 3.1.7 Stretch Database..... | 23 |
| 3.2 EDÍCIE MICROSOFT SQL SERVER 2016..... | 23 |
| 3.3 PREHEAD FUNKCII SQL SERVER 2016 ENTERPRISE | 24 |
| 3.3.1 Operačná analýza v reálnom čase | 24 |
| 3.3.2 Vysoká dostupnosť a obnova po havárii | 24 |

| | | |
|-----------|---|-----------|
| 3.3.3 | Zabezpečenie dát | 24 |
| 3.3.4 | Dátové sklady a výkon | 24 |
| 3.3.5 | Práca s dátami | 24 |
| 3.3.6 | Nástroje | 24 |
| 3.4 | PREHEAD FUNKCII SQL SERVER 2016 STANDARD | 24 |
| 3.4.1 | Operačná analýza | 24 |
| 3.4.2 | Zabezpečenie dát | 25 |
| 3.4.3 | Zotavenie po havárii..... | 25 |
| 3.4.4 | Práca s dátami | 25 |
| 3.4.5 | Nástroje | 25 |
| 3.5 | PREHEAD FUNKCII SQL SERVER 2016 EXPRESS..... | 25 |
| 3.5.1 | SQL Server 2016 Express LocalDB | 25 |
| 3.5.2 | SQL Server 2016 Express s pokročilými službami..... | 25 |
| 3.5.3 | SQL Server 2016 Express | 25 |
| 3.6 | PREHEAD FUNKCII SQL SERVER 2016 DEVELOPER..... | 26 |
| 3.6.1 | Operačná analýza v reálnom čase | 26 |
| 3.6.2 | Vysoká dostupnosť a obnova po havárii | 26 |
| 3.6.3 | Zabezpečenie dát | 26 |
| 3.6.4 | Dátové sklady a výkon | 26 |
| 3.6.5 | Práca s dátami | 26 |
| 3.6.6 | Nástroje | 26 |
| 3.7 | POROVNANIE VERZII MICROSOFT SQL SERVER 2008-2016 | 26 |
| 3.7.1 | Funkcie zamerané na zvýšenie výkonu | 27 |
| 3.7.2 | Funkcie zamerané na zvýšenie dostupnosti | 27 |
| 3.7.3 | Funkcie zamerané na zvýšenie zabezpečenia..... | 28 |
| 3.7.4 | Funkcie zamerané na zvýšenie integrity s cloud prostredím | 28 |
| 3.7.5 | Funkcie zamerané na zvýšenie možnosti správy databázy | 29 |
| 3.7.6 | Funkcie zamerané na zlepšenie analýzy..... | 29 |
| II | PRAKTICKÁ ČASŤ | 31 |
| 4 | MICROSOFT SQL SERVER 2014 | 32 |
| 4.1 | INŠTALÁCIA MICROSOFT SQL SERVER 2014 EXPRESS..... | 32 |
| 5 | PRÁCA S MICROSOFT SQL SERVER 2014..... | 42 |
| 5.1 | TVORBA OBJEKTOV | 42 |
| 5.1.1 | Tvorba databáze | 42 |
| 5.1.2 | Tvorba tabuľky..... | 42 |
| 5.2 | PRÁCA S TABUĽKOU | 43 |
| 5.2.1 | SELECT | 43 |
| 5.2.2 | WHERE..... | 43 |
| 5.2.3 | GROUP BY..... | 44 |
| 5.2.4 | HAVING | 44 |
| 5.2.5 | ORDER BY | 44 |
| 5.2.6 | INSERT..... | 45 |
| 5.2.7 | UPDATE | 45 |
| 5.2.8 | DELETE..... | 45 |
| 5.2.9 | DROP TABLE | 46 |
| 5.2.10 | INNER JOIN | 46 |
| 5.2.11 | OUTER JOIN | 46 |

| | | |
|--|---------------------------------------|----|
| 5.2.11.1 | LEFT OUTER JOIN | 46 |
| 5.2.11.2 | RIGHT OUTER JOIN..... | 47 |
| 5.2.11.3 | FULL OUTER JOIN..... | 47 |
| 5.2.12 | CROSS JOIN | 48 |
| 5.2.13 | UNION..... | 48 |
| 5.2.14 | INTERSECT | 49 |
| 5.2.15 | EXCEPT..... | 49 |
| 5.3 | AGREGÁČNÉ FUNKCIE | 50 |
| 5.3.1 | AVG..... | 50 |
| 5.3.2 | MIN, MAX..... | 50 |
| 5.3.3 | SUM..... | 50 |
| 5.3.4 | COUNT..... | 50 |
| 5.4 | VSTAVANÉ FUNKCIE..... | 51 |
| 5.4.1 | Matematické funkcie..... | 51 |
| 5.4.2 | Textové funkcie..... | 51 |
| 5.4.3 | Dátumové funkcie | 52 |
| 5.5 | DÁTOVÉ TYPY | 52 |
| 5.5.1 | Celočíselné dátové typy | 52 |
| 5.5.2 | Desatinné dátové typy | 53 |
| 5.5.3 | Dátumové dátové typy | 53 |
| 5.5.4 | Znakové dátové typy | 53 |
| 5.5.5 | Binárne dátové typy | 53 |
| 5.6 | PROGRAMOVANIE SQL | 54 |
| 5.6.1 | Vytvorenie premennej..... | 54 |
| 5.6.2 | Vytváranie funkcie | 54 |
| 5.6.3 | Vytváranie podmienky | 54 |
| 5.6.4 | Vytvorenie rozhodovacej funkcie | 54 |
| 5.6.5 | Vytvorenie cyklu | 55 |
| 5.6.6 | Trigger..... | 55 |
| 5.6.7 | Procedúry | 56 |
| ZÁVĚR | 57 | |
| SEZNAM POUŽITÉ LITERATURY..... | 58 | |
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK..... | 61 | |
| SEZNAM OBRÁZKŮ | 62 | |
| SEZNAM TABULEK..... | 63 | |
| SEZNAM PŘÍLOH..... | 64 | |

ÚVOD

Databázové systémy sú najrozšírenejší spôsob uchovávaní a spracovania dát v dnešnej dobe. Umožňujú rýchlo spracovať veľké množstvo dát. Ak je databáza uložená na serveri s prístupom na internet umožňuje okamžitý bezpečný prístup k dátam z ktoréhokoľvek miesta pripojeného na internetovú sieť.

Každý databázový systém je postavený na určitom databázovom modeli, ktorý určuje štruktúru a väzby medzi dátami v databáze. Vďaka tejto jednotnej logike v spojení s dotazovacím jazykom SQL je možné vykonávať úplnú kontrolu nad databázou.

Preto si táto bakalárska práca berie za cieľ v teoretickej časti jednoduchou cestou predstaviť čitateľom históriu a logiku databázových systémov. Následne oboznámiť s najnovším vývojovým prostredím Microsoft SQL Server 2016 od spoločnosti Microsoft. A ukázať jednotlivé možnosti a služby, ktoré toto vývojové prostredie ponúka.

V teoretickej časti sa nachádzajú základné informácie potrebné pre prácu s vývojovým prostredím Microsoft SQL Server 2014 vo verzii Express. Postupne od úplných základov ukazujú ako krok po kroku nainštalovať vývojové prostredie, vytvoriť jednoduchú databázu a naplniť ju dátami tak aby sa dalo s nimi ďalej pracovať.

Súčasťou praktickej časti sú aj prezentácie napísané v anglickom jazyku vhodné na výuku anglicky hovoriacich študentov na predmete zameranom na databázové systémy. Prezentácie zobrazujú prevažnú väčšinu informácií obsiahnutých v bakalárskej práci a na jednoduchých príkladoch ktoré na seba nadväzujú vysvetľujú logiku a fungovanie jednotlivých príkazov jazyka SQL. Taktiež zahŕňa 60 testovacích úloh ktoré sú vhodné na import do školského systému MOODLE.

I. TEORETICKÁ ČÁST

1 DATABÁZA

Databáza je množina usporiadaných dát, ktoré sú navzájom poprepájane a uložené na pamäťovom disku. Dáta sú usporiadané tak, aby sa s nimi dalo efektívne pracovať, tj. zobraziť, pridať, aktualizovať apod. [1]

1.1 História databáz

Za úplne prvých predchodcov dnešných databáz sa môžu považovať papierové kartotéky. Dáta v týchto kartotékach boli uložené podľa rôznych kritérií na základe ktorých umožňovali pridávať a zatriedovať nové dáta. Manipuláciu s dátami zabezpečoval človek.[2]

Prvý väčší posun v spracovaní dát nastal v roku 1890, kedy sa po prvý krát použili dierované štítky na sčítanie ľudu v USA. Dierované štítky predstavovali pamäťové médium, ktoré bolo následne spracované pomocou elektromechanického stroja.[2]

S príchodom počítačov vznikla nutnosť vyššieho programovacieho jazyka pre spracovanie dát. Nový programovací jazyk by mal na rozdiel od strojového kódu spĺňať nasledujúce kritéria:

- Umožniť jednoduché zostavenie programu za minimálny čas
- Možnosť prevodu programu na nové typy počítačov
- Zápis programu by mal byť v jazyku blízkom angličtine
- Ponúkať úplnú dokumentáciu programu

Preto v roku 1960 na konferencii CODASYL bola publikovaná prvá verzia programovacieho jazyka COBOL.[2]

1.2 Databázové modely

V roku 1970 sa po prvý krát objavujú pojmy ako schéma databázy, subschéma, jazyk pre definíciu schém a je popísaná celá architektúra sieťového databázového systému. Zároveň sa vyvíjajú hierarchické a relačné databázy.[2]

V roku 1974 vzniká prvá verzia dotazovacieho jazyka SQL a neskôr objektovo orientované databázy, ktorých základy boli postavené na objektovo orientovaných jazykoch.

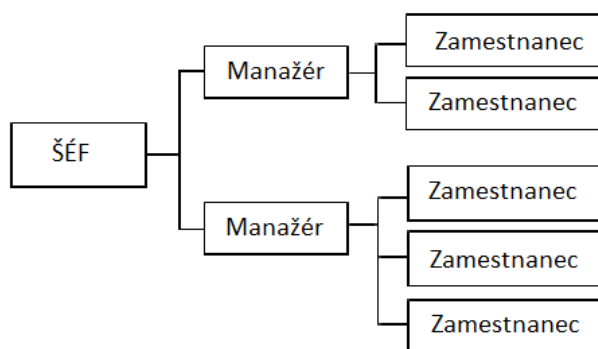
Následne vzniká spojením relačných a objektovo orientovaných systémov objektovo-relačná technológia.[2]

1.2.1 Hierarchický model

Dáta sú usporiadané do stromovej štruktúry , kde vzťah medzi dátami je typu rodič – potomok. Pohyb v štruktúre je možný len v smere rodič – potomok , potomok – rodič alebo medzi potomkami do strany. [3]

Základné charakteristické vlastnosti hierarchického modelu sú :

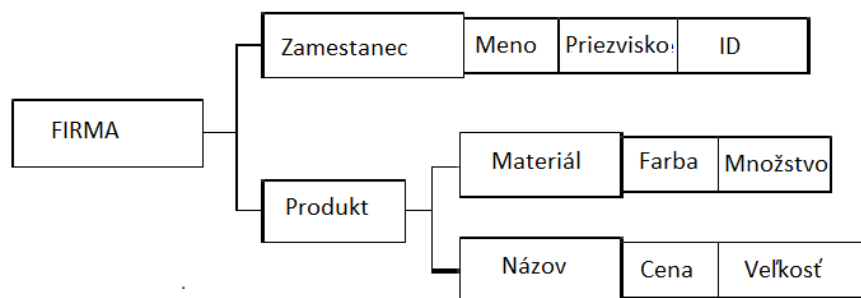
- Má jeden koreň , ktorý nemá žiadneho rodiča
- Každá ďalšia vetva okrem koreňa ma práve jedného rodiča
- Každý rodič môže byť rodičom pre 1 až N potomkov
- Vetva ktorá nie je rodičom pre inú vetvu sa nazýva list. [3]



Obrázok 1 – Hierarchický model

1.2.2 Sieťový model

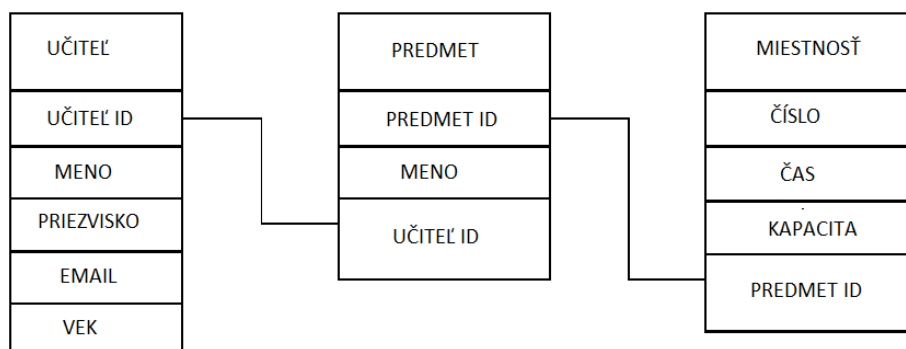
Sieťový model nadväzuje na hierarchický model, pričom je doplnený o mnohonásobné vzťahy (sety). Sety umožňujú prepojenie rôznych záznamov a následne priamy prístup k záznamom bez nutnosti ďalšieho vyhľadávania. K vyhľadaniu záznamu sa používa kľúč.[3]



Obrázok 2 – Siet'ový model

1.2.3 Relačný model

Relačný model je postavený na jednoduchšej štruktúre tvorenej tabuľkami, ktoré sa skladajú z riadkov a stĺpcov. Každý riadok obsahuje unikátny identifikačný kľuč. Relačný model umožňuje užívateľom vytvárať dotazy a tak vybrať len informácie ktoré v daný okamžik potrebuje.[3][4]



Obrázok 3- Relačný model

1.2.4 Objektový model

Predstavuje kombináciu objektovo orientovaného programovania a schopnosti databáz. Objektový model využíva objektovo orientované aspekty ako sú triedy s objektami a metódami, integritné obmedzenia, abstraktné dátové typy, násobnú dedičnosť apod. Výsledný model aplikácie a databázy sa v mnohých smeroch zhoduje a je pohodlnejší pre následnú údržbu. [4]

1.2.5 Objektovo-relačný model

Prestavuje kombináciu objektového a relačného modelu kedy zachováva dáta v tabuľkách ale obohacuje položky o objektovo orientované aspekty. Príkladom je abstraktný dátový

typ, ktorý vzniká kombináciou základných dátových typov. Takéto abstraktné dátové typy sa môžu následne používať na indexáciu alebo ukladanie nových dát na základe obsahu. [4]

1.3 Entitne – relačné modelovanie

1.3.1 Entita

Entita je objekt ktorý ma jedinečne pomenovanie a vlastnosti. Entita môže by reálny ale aj abstraktný objekt, ktorý by malo by možné jednoznačne identifikovať. Vlastnosti pomocou ktorý dokážeme identifikovať entitu nazývame atribúty. Slabá entita je taká entita, ktorá svojim primárnym kľúčom čiastočne alebo úplne závisí na existencii inej entity. Silná entita je taká entita, ktorej primárny kľúč nezávisí na existencii inej entity.[5][25]

1.3.2 Relácia

Je množina spojení medzi entitami, ktoré je možné jednoznačne identifikovať. Každá relácia by mala byť pomenovaná pomocú názvu ktorý popisuje jej funkciu. Relácie rozdeľujeme na stupne podľa toho koľko entít sa na relácii zúčastňuje. Relácia v ktorej vystupuje len jedna entita sa nazýva rekurzívna relácia. Relácia ktorá sa vyskytuje medzi dvoma entitami sa nazýva binárna. Relácie v ktorých sa vyskytujú viac ako dve entity sa súhrne nazývajú zložite relácie.[5]

1.3.3 Atribút

Atribút predstavuje vlastnosť entity alebo relácie. Atribúty môžeme klasifikovať na rôzne skupiny:[25]

1.3.3.1 Jednoduché a zložené atribúty

Jednoduché atribúty, nazývane aj atomické atribúty, nie je možné viac rozdeliť. Jednoduchý atribút môže byť napríklad farba ktorou opíšeme objekt. Zložený atribút sa skladá z viacerých komponentov. Príkladom pre zložený atribút môže byť napríklad moje meno Patrik Čordáš, ktoré je možné rozdeliť na dva rôzne atribúty a to meno Patrik a priezvisko Čordáš.[5][25]

1.3.3.2 Atribúty s jednou alebo viacerými hodnotami

Atribúty s jednou hodnotu bývajú väčšinou atribúty ktoré sú priradené k jednej konkrétnej entite. Atribút s viacerými hodnotami je atribút, ktorý na jeden výskyt entity obsahuje viacero hodnôt.[5][25]

1.3.3.3 Odvodené atribúty

Odvodené atribúty sú také atribúty, ktorých hodnotu je možné odvodiť z súvisiacich atribútov alebo skupiny súvisiacich atribútov. Najjednoduchší príklad odvodeného atribútu môže byť vek, ktorý sa dá odvodiť z iného atribútu a to je dátum narodenia.[5][25]

1.3.4 Kľúč relácie

Kľúč predstavuje jedinečný identifikátor, ktorý umožňuje identifikovať záznam pomocou stĺpca alebo kombinácie stĺpcov.

Super kľúč je stĺpec alebo skupina stĺpcov ktoré jednoznačne identifikujú záznam v relácii. Super kľúč môže obsahovať prebytočné stĺpce preto sa snažíme minimalizovať počet stĺpcov potrebných pre identifikáciu záznamu, takýto super kľúč nazývame potom kandidátny kľúč. Primárny kľúč je kandidátny kľúč, ktorý bol vybraný aby identifikoval záznam. Každá tabuľka má primárny kľúč na jednoznačnú identifikáciu záznamu. Ako alternatívny kľúč sa môže používať kandidátny kľúč. Cudzí kľúč predstavuje stĺpec alebo skupinu stĺpcov ktoré sa zhodujú s kandidátnym kľúčom niektorej inej tabuľky.[5]

1.3.5 Obmedzenie multiplicity v relácii

Multiplicita, hlavný typ obmedzenia v relácii, obmedzuje počet výskytov entít ktoré súvisia s inou entitou pomocou príslušných relácii. Najčastejší stupeň relácie je binárna relácia, ktorá môže mať multiplicitu jedna k jednej (1:1), jedna ku viaceru (1:N) a viaceru ku viaceru (N:N).[5]

1.3.5.1 Relácia 1:1

Jedna k jednej predstavuje reláciu kedy práve jeden záznam v tabuľke zodpovedá práve jednému záznamu v druhej tabuľke. Používa sa na sprehľadnenie veľkých tabuliek. [5]

1.3.5.2 Relácia 1:N

Jedna k viacerým predstavuje reláciu kedy práve jeden záznam v tabuľke je priradený k viacerým záznamom v druhej tabuľke. [5]

1.3.5.3 Relácia N:N

Relácia viac k viacero predstavuje reláciu kedy každý záznam z jednej tabuľky je možné priradiť ku každému záznamu z druhej tabuľky a zároveň každý záznam z druhej tabuľky je možné priradiť ku každému záznamu z prvej tabuľky. [5]

1.4 Normalizácia

Je spôsob ktorý sa používa na vytvorenie tabuliek s minimálnou redundanciou. Normalizácia ponúka dva spôsoby pri návrhu databázy. Návrh databáze z dolu hore a návrh spoločne s Entitne-relačným modelovaním. Normalizácia sa používa na testovanie tabuliek aby sa zistilo dodržanie alebo porušenie pravidiel určených pre normálnu formu. Základne normálne formy sú prvá normálna forma (1NF), druhá normálna forma (2NF) a tretia normálna forma (3NF). Všetky tieto normy sú založené na vzťahoch medzi stĺpcami v tabuľke. [5]

1.4.1 Prvá normálna forma (1NF)

Prvá normálna forma je najdôležitejšia normálna forma nutná na vytvorenie vhodných tabuliek pre relačnú databázu. Tabuľka ktorá spĺňa pravidla 1NF je taká tabuľka v ktorej každý priesečník stĺpca a záznamu obsahuje práve jednu hodnotu. [5]

1.4.2 Druhá normálna forma (2NF)

Druhá normálna forma sa zaoberá len tabuľkami ktoré obsahujú zložený primárny kľúč. To sú tabuľky ktorých primárny kľúč tvorí dva alebo viacero stĺpcov v tabuľke. Tabuľka ktorá spĺňa pravidla 2NF je taká tabuľka, ktorá zároveň spĺňa pravidla 1NF a v ktorej sú hodnoty všetkých stĺpcov ktoré nie sú súčasťou primárneho kľúča, determinované všetkými hodnotami stĺpcov ktoré tvoria primárny kľúč.[5]

1.4.3 Tretia normálna forma (3NF)

Tabuľka ktorá spĺňa pravidla 3NF je taká tabuľka ktorá spĺňa pravidla 1NF a 2NF. A zároveň v ktorej všetky hodnoty v stĺpcoch, ktoré nepatria k primárnym kľúčom sú de-

terminované len stĺpcami primárneho kľúča a nie sú determinované žiadnymi inými stĺpcami. [5]

2 SQL

SQL z anglického Structured Query Language v preklade štruktúrovaný dotazovací jazyk. Slúži ako nástroj pre správu , organizáciu a manipuláciu dát v databázových systémoch.[6]

2.1 História SQL

SQL je nástupca jazyka SEQUEL, ktorý vznikol v roku 1974 v spojitosti s relačným dátovým modelom SystemR. Vznik SQL bol podmienený potrebou vyvinúť štandardizovaný dotazovací jazyk, pomocou ktorého by bolo možné programovať relačné databázy. Preto v roku 1986 bola prijatá organizáciou ANSI (American National Standards Institute) norma SQL86 . Nasledujúci rok normu SQL86 prevzala aj organizácia ISO (International Standards Organization). V roku 1989 boli doplnené integritné obmedzenia a vyšla norma SQL89. V roku 1992 vzniká norma SQL2, kde boli pridané napríklad nové dátové typy ako DATE, TIME a BIT alebo operátori UNION JOIN a NATURAL JOIN. Norma SQL3 vydaná v roku 1999 reaguje na potrebu doplniť objektové prvky. Taktiež doplnila regulárne výrazy , rekurzívne dotazy, trigger apod. V roku 2003 boli pridané XML funkcie , funkcia window a automaticky generované hodnoty. V roku 2006 je definovaný spôsob ako vkladať a ukladať XML dáta v SQL databáze. Ďalšia norma z roku 2008 pridáva trigger INSTEAD OF a povoľuje použitie ORDER BY mimo definície. V roku 2011 bola pridaná definícia časových údajov a spôsob ich manipulácie. V poslednej norme z roku 2016 boli pridané polymorfické tabuľky a JSON.[6][7]

2.2 DDL

DDL (Data Definition Language) je časť jazyka SQL, ktorý slúži na definovanie dátových štruktúr. Patria sem nasledujúce príkazy :

- založenie štruktúry databázy: CREATE TABLE, CREATE VIEW a CREATE INDEX
- predefinovanie štruktúry databázy: ALTER TABLE, DROP TABLE, DROP VIEW a DROP INDEX [6][8]

2.3 DML

DML (Data Manipulation Language) je časť jazyka SQL, ktorá slúži na manipuláciu dát v databáze. Patria sem nasledujúce príkazy :

- dotazovací příkaz: SELECT
- filtrovanie výstupu : WHERE
- vytváranie skupiny riadkov : GROUP BY
- filtrovanie skupiny riadkov : HAVING
- usporiadanie výstupu : ORDER BY
- vkladanie dát do vytvorenej tabuľky : INSERT
- modifikácia riadkov tabuľky : UPDATE
- odstránenie riadkov tabuľky : DELETE
- vymazanie celej tabuľky a všetkých riadkov : DROP TABLE [6][8]

2.4 DCL

DCL (Data Control Language) je časť jazyka SQL, ktorá slúži na zabezpečenie dát v databáze. Patria sem nasledujúce príkazy :

- pridelovanie prístupových práv k danej databáze : GRANT
- odobranie už pridelených práv : REVOKE [8]

3 MICROSOFT SQL SERVER

Microsoft SQL Server predstavuje relačný databázový systém vyvinutý spoločnosťou Microsoft. Jeho primárnou funkciou je ukladanie a načítanie dát podľa požiadaviek užívateľa. Umožňuje taktiež vykonávať operácie na základe požiadaviek iných aplikácii bežiacich na vzdialenom serveri. [18]

3.1 Nové funkcie v Microsoft SQL Server 2016

Vo verzii Microsoft SQL Server 2016 prichádza množstvo nových funkcií, ktoré sa prevažne zameriavajú na bezpečnosť dát uložených v databázach a integritu dát uložených na cloudových serveroch s dátami ktoré sa nachádzajú priamo na serveri.[9]

3.1.1 Row Level Security

Zabezpečenie na úrovni riadku slúži na obmedzenie prístupu k vybraným riadkom v databázovej tabuľke na základe toho aký užívateľ daný dotaz spustil. Vďaka tomu, že dáta sú zabezpečené už na úrovni databáze, užívateľ sa ani nedozvie, že nejaké záznamy boli odfiltrované. To isté platí aj pri pokuse o aktualizáciu alebo odstránenie záznamu, ktorý je na základe obmedzení odfiltrovaný. [9][15]

3.1.2 Live Query Statistics

V predchádzajúcich verziách Microsoft SQL Server boli len dve možnosti na zobrazenie exekučného plánu dotazu. Prvá možnosť bola pripojiť skutočný použitý plán k výsledku dotazu. Druhá možnosť ponúkala ešte pred spustením dotazu zobrazit' odhadovaný plán. V najnovšej verzii pribudla tretia možnosť Live Query Statistic čo znamená, že je možné sledovať exekučný plán aktuálne spusteného dotazu. Vďaka tejto možnosti je možné sledovať ako spracovanie dotazu plynie z jedného operátora na ďalší.

Vďaka Live Query Statistic môžeme sledovať :

- celkový priebeh dotazu
- ktoré operátory sú už spracované
- počet spracovaných riadkov
- koľko času ubehlo od spustenia dotazu
- priebeh spracovania jednotlivých operátorov [10][17]

3.1.3 Polybase

PolyBase umožňuje přístup k datům které sú uložené na cloude ako je Azure Blob Storage. Následne je možné dáta čítať rovnakým spôsobom ako sa čítajú tabuľky uložené na lokálnom serveri. Taktiež umožňuje pomocou JOIN kombinovať dáta uložené na cloude s dátami ktoré sú na lokálnom úložisku. [9][11]

3.1.4 Podpora JSON

Podpora JSON prichádza s funkciami FOR JSON, OPENJSON, ISJSON, JSON_VALUE a JSON_QUERY.

- FOR JSON slúži na transformáciu výstupu do formátu JSON.
- OPENJSON slúži na prevod JSON do tabuľky.
- ISJSON slúži na overenie validity JSON.
- JSON_VALUE vracia skalárnu hodnotu.
- JSON_QUERY vracia z JSON fragment čo predstavuje pole alebo objekt. [9][12]

3.1.5 Dynamic Data Masking

V preklade maskovanie dynamických dát, umožňuje zakrývať dáta alebo ich podstatnú časť tak aby neboli viditeľne pre neoprávnených užívateľov. Najčastejšie sa jedná o súkromné dáta ako sú rodné číslo , číslo kreditnej karty , dátum narodenia alebo priezvisko. Použitím Dynamic Data Masking zostávajú dáta v databáze nezmenené. Mení sa len hodnota ktorá bude zobrazená vo výslednom dotaze pre neoprávneného užívateľa. V definícii maskovania je možné ako funkciu použiť 3 možnosti : default , email alebo partial.

Default spôsobí úplne zamaskovanie dát a to v závislosti na použitom dátovom type.

- Textové dátové typy ako char, varchar, text sa zamaskujú pomocou reťazca „xxxx“.
- Číselné dátové typy ako bit, decimal, int, float sa zamaskujú pomocou znaku „0“.
- Dátum a čas sa zamaskuje ako „01.01.200 00:00:00.0000000“.
- Binárny typ sa zamaskuje ako jeden byte v ASCII je to „0“.

Možnosť email slúži na zamaskovanie emailovej adresy. Dáta sú maskované tak, že neoprávnený užívateľ vidí len prvé písmeno z emailovej adresy. Ostatné znaky sú nahradené znakom „x“ a adresa vždy končí reťazcom „.com“.

Možnosť partial slúži na čiastočné zamaskovanie dát. V tejto možnosti je možné nastaviť koľko znakov z ľavej a pravej časti reťazca ostane odhalených a taktiež aký znak sa použije na zamaskovanie ostatných znakov. [10][16]

3.1.6 Temporal Tables

Táto funkcia umožní nastaviť tabuľkám sledovanie histórie, potom už len stačí použiť príkaz SELECT a príslušný čas alebo interval, ktorý nás zaujíma. Ďalej nám umožní porovnávať 2 a viac časových okamžikov, filtrovať zmeny konkrétneho záznamu alebo môže slúžiť na zálohu databáze na úrovni riadka. [13]

3.1.7 Stretch Database

Táto funkcia umožňuje presun málo využívaných dát, ku ktorým je ale potrebné zachovávať prístup, na cloudové úložisko. Po presune nie je nutné nejak upravovať aplikácie pretože prístup k týmto dátam je rovnaký ako keby sa nachádzali na lokálnom SQL serveri. Jediný rozdiel je v odozve na dotazy, ktorá je v prípade dát uložených na cloude o čosi väčšia ako pri dátach uložených na lokálnom SQL serveri. Pred presunutím je nutné dáta presunúť do samostatnej tabuľky a mať vytvorený účet MS Azure. Miesto kam sa dáta následne presunú je Azure SQL databáza. Sprievodca Stretch Database túto databázu sám vytvorí a automaticky nastaví, tieto nastavenie je možné neskôr zmeniť alebo celý proces vrátiť a presunúť dáta späť na lokálny SQL server.[9][14]

3.2 Edície Microsoft SQL Server 2016

Spoločnosť Microsoft ponúka 4 základne edície Microsoft SQL Server 2016. Tieto edície sa zameriavajú na rôzne skupiny užívateľov. Najnižšia edícia Express je úplne bezplatná a slúži prevažne na malé dátové aplikácie. Pre stredne veľké projekty je určená verzia Standard, ponúka veľké množstvo programovacích funkcií, inovácie v zabezpečení a vysoký výkon. Najvyššiu triedu zastupuje verzia Enterprise ktorá ponúka úplne všetky funkcie a vymoženosti. Maximálne zabezpečenie a možnosť neobmedzené virtualizácie v spojení s maximálnym výkonom aj pre tie najnáročnejšie projekty. Edícia Developer je bezplatná verzia, ktorá ponúka všetky možnosti z edície Enterprise pre vývojárske a testovacie účely. [19]

3.3 Prehľad funkcií SQL Server 2016 Enterprise

3.3.1 Operačná analýza v reálnom čase

Umožňuje vykonávať analýzu priamo na transakčných dátach pomocou funkcií columnstore a rowstore. [20]

3.3.2 Vysoká dostupnosť a obnova po havárii

Pomocou technológie Always On umožňuje ľahké vyrovnanie záťaži sekundárnych inštalácií a rýchlejšie prevzatie služieb po havárii. Taktiež ponúka možnosť vytvoriť asynchrónne repliky pomocou služby Azure Virtual Machines. [20]

3.3.3 Zabezpečenie dát

Zabezpečenie dát pomocou technológií Always Encrypted , Row Level Security, Dynamic Data Masking. [20]

3.3.4 Dátové sklady a výkon

Umožňuje škálovať až 640 logických procesorov s celkovou adresovateľnou pamäťou 12 TB. Pomocou architektúry MPP (Massive Parallel Processing) ponúka možnosť integrácie s nerelačnými zdrojmi ako je Hadoop. [20]

3.3.5 Práca s dátami

Podpora analýzy ,čítania a ukladania dokumentov JSON. Funkcia PolyBase pre prácu s dátami na cloude. Uchovávanie historických dát pomocou technológie Stretch Database. [20]

3.3.6 Nástroje

Azure Active Directory , SQL Server Management Studio, vývojárske nástroje integrované do Visual Studio. [20]

3.4 Prehľad funkcií SQL Server 2016 Standard

3.4.1 Operačná analýza

Hybridné spracovanie transakcií a analýza v reálnom čase pomocou funkcií columnstore a rowstore. [21]

3.4.2 Zabezpečenie dát

Zabezpečenie dát pomocou funkcií Row Level security , Dynamic Data Masking a Always Encrypted. [21]

3.4.3 Zotavenie po havárii

Zaistenie dát pred stratou pomocou synchronnej repliky s dvoma uzlami. A služba Azure Virtual Machines pre ukladanie asynchronnej repliky pre hybridne scenár zotavenia po zlyhaní. [21]

3.4.4 Práca s dátami

Podpora práce s JSON dokumentami. Možnosť pracovať s dátami ktoré sú uložené na cloude pomocou funkcie PolyBase. A ukladanie historických dát pomocou Stretch Database. [21]

3.4.5 Nástroje

Azure Active Directory , SQL Server Management Studio a integrácia s aplikáciou Visual Studio. [21]

3.5 Prehľad funkcií SQL Server 2016 Express

Edícia SQL Server 2016 Express vychádza v 3 rôznych verziách:

3.5.1 SQL Server 2016 Express LocalDB

Ponúka funkcie edície SQL Server Express so všetkými programovacími funkciami v užívateľskom režime bez možnosti vzdialeného spravovania. Ponúka rýchlu inštaláciu bez nutnosti konfigurácie. [22]

3.5.2 SQL Server 2016 Express s pokročilými službami

Základný databázový server s funkciou fulltextového vyhľadávania a službou Reporting Services. [22]

3.5.3 SQL Server 2016 Express

Základný databázový server bez funkcie fulltextového vyhľadávania a bez služby Reporting Services. [22]

3.6 Prehľad funkcií SQL Server 2016 Developer

3.6.1 Operačná analýza v reálnom čase

Umožňuje vykonávať analýzu priamo na transakčných dátach pomocou funkcií columnstore a rowstore. [23]

3.6.2 Vysoká dostupnosť a obnova po havárii

Pomocou technológie Always On umožňuje ľahké vyrovnanie záťaží sekundárnych inštalácií a rýchlejšie prevzatie služieb po havárii. Taktiež ponúka možnosť vytvoriť asynchrónne repliky pomocou služby Azure Virtual Machines. [23]

3.6.3 Zabezpečenie dát

Zabezpečenie dát pomocou technológií Always Encrypted, Row Level Security, Dynamic Data Masking. [23]

3.6.4 Dátové sklady a výkon

Umožňuje škálovať až 640 logických procesorov s celkovou adresovateľnou pamäťou 12 TB. Pomocou architektúry MPP (Massive Parallel Processing) ponúka možnosť integrácie s nerelačnými zdrojmi ako je Hadoop. [23]

3.6.5 Práca s dátami

Podpora analýzy, čítania a ukladania dokumentov JSON. Funkcia PolyBase pre prácu s dátami na cloude. Uchovávanie historických dát pomocou technológie Stretch Database. [23]

3.6.6 Nástroje

Azure Active Directory, SQL Server Management Studio, vývojárske nástroje integrované do Visual Studio. [23]

3.7 Porovnanie verzii Microsoft SQL Server 2008-2016

Spoločnosť Microsoft prináša pravidelne nové verzie produktu Microsoft SQL Server. Zameriava sa hlavne na vylepšenia v oblasti výkonu, bezpečnosti, správy a analýzy. Pridáva nové funkcie a prvky. [24]

3.7.1 Funkcie zamerané na zvýšenie výkonu

Výkon je jeden z kľúčových aspektov každej aplikácie. Preto sa v každej verzii Microsoft SQL Server objavujú nové funkcie zamerané na zvýšenie výkonu. Funkcia *In-memory OLTP* ponúka spracovanie transakcií v reálnom čase, *In-memory columnstore* organizuje a komprimuje dáta, *Buffer pool extension to SSD* umožňuje rozšírenie vyrovnávacej pamäte na SSD disk a *Resource Governor* umožňuje riadenie priority zaťaženia prostriedkov. V prehľadnej tabuľke je možné odsledovať ktorá funkcia bola pridaná v ktorej verzii Microsoft SQL Server. [24]

| Výkon | | | | |
|------------------------------|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| In-memory OLTP | Áno | Áno | Nie | Nie |
| In-memory columnstore | Áno | Áno | Áno | Nie |
| Buffer pool extension to SSD | Áno | Áno | Nie | Nie |
| Resource Governor | Áno | Áno | Áno | Áno |

Tabuľka 1 Funkcie zamerané na výkon

3.7.2 Funkcie zamerané na zvýšenie dostupnosti

Dostupnosť je jeden z kľúčových aspektov pre užívateľa. Preto sa v každej verzii Microsoft SQL Server objavujú nové funkcie zamerané na zvýšenie dostupnosti služieb. Funkcia *Always ON* umožňuje rýchlu dostupnosť poskytovaných služieb a funkcia *Enhanced virtualization support and live migration* umožňuje zobrazenie presunú dát vo vizualizácii. V prehľadnej tabuľke je možné odsledovať ktorá funkcia bola pridaná v ktorej verzii Microsoft SQL Server. [24]

| Dostupnosť | | | | |
|--|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| Always On | Áno | Áno | Áno | Nie |
| Enhanced virtualization support and live migration | Áno | Áno | Áno | Nie |

Tabuľka 2 Funkcie zamerané na dostupnosť služieb

3.7.3 Funkcie zamerané na zvýšenie zabezpečenia

Zabezpečenie databázy je dôležitý aspekt každej väčšej firmy. Preto Microsoft SQL Server ponúka hneď niekoľko funkcií zameraných len na bezpečnosť databázy. Funkcia *Transparent data encryption* ponúka šifrovanie priamo na úrovni databázy, *Backup encryption support* umožňuje šifrovanie zálohovaných dát, *Fine-grained auditing* ponúka štruktúrované riadenie prístupu a *Separation of duties* umožňuje dynamické vylučovanie zodpovednosti. [24]

| Zabezpečenie | | | | |
|-----------------------------|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| Transparent data encryption | Áno | Áno | Áno | Áno |
| Backup encryption support | Áno | Áno | Nie | Nie |
| Fine-grained auditing | Áno | Áno | Áno | Áno |
| Separation of duties | Áno | Áno | Áno | Nie |

Tabuľka 3 Funkcie zamerané na zabezpečenie databázy

3.7.4 Funkcie zamerané na zvýšenie integrity s cloud prostredím

Integrita s cloudom umožňuje jednoduchý spôsob ako presunúť malo používané prostriedky v databáze na virtuálny disk a uvoľniť prostriedky pre iné potrebnejšie operácie a dáta. Funkcia *Backup to Azure* umožňuje zálohovať kritické dáta na cloud Azure, *Disaster recovery to Microsoft* pomáha so zotavením po havárii a *Optimized virtual machine images in Azure gallery* optimalizuje obrazy záloh na serveri Azure. [24]

| Cloud | | | | |
|---|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| Backup to Azure | Áno | Áno | Áno | Nie |
| Disaster recovery to Microsoft | Áno | Áno | Nie | Nie |
| Optimized virtual machine images in Azure gallery | Áno | Áno | Áno | Nie |

Tabuľka 4 Funkcie zamerané na integritu s cloudom

3.7.5 Funkcie zamerané na zvýšenie možnosti správy databázy

Funkcie pre správu databázy umožňujú optimalizovať niektoré procesy a urýchliť chod a správu databázy. *Distributed replay* umožňuje prístup k aktualizáciám, *Policy-based management* umožňuje riadenie jednej alebo viacerých inštancií a *Enhanced programmability* pomáha optimalizovať procesy v databáze. [24]

| Správa | | | | |
|--------------------------|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| Distributed replay | Áno | Áno | Áno | Nie |
| Policy-based management | Áno | Áno | Áno | Áno |
| Enhanced programmability | Áno | Áno | Áno | Áno |

Tabuľka 5 Funkcie zamerané na správu databázy

3.7.6 Funkcie zamerané na zlepšenie analýzy

PowerPivot for Excel predstavuje dostupnosť služby pre program Microsoft Excel, *Integration services managed as a server* umožňuje čistiť a spravovať dáta na serveri. *Hadoop connector via Apache Sqoop* predstavuje možnosť prepojenia *Hadoop connector* cez *Apache Sqoop*. *Master data services* umožňuje analýzu hlavných dátových služieb. *Data quality services* kontroluje kvalitu služieb prevádzkovaných na serveri. [24]

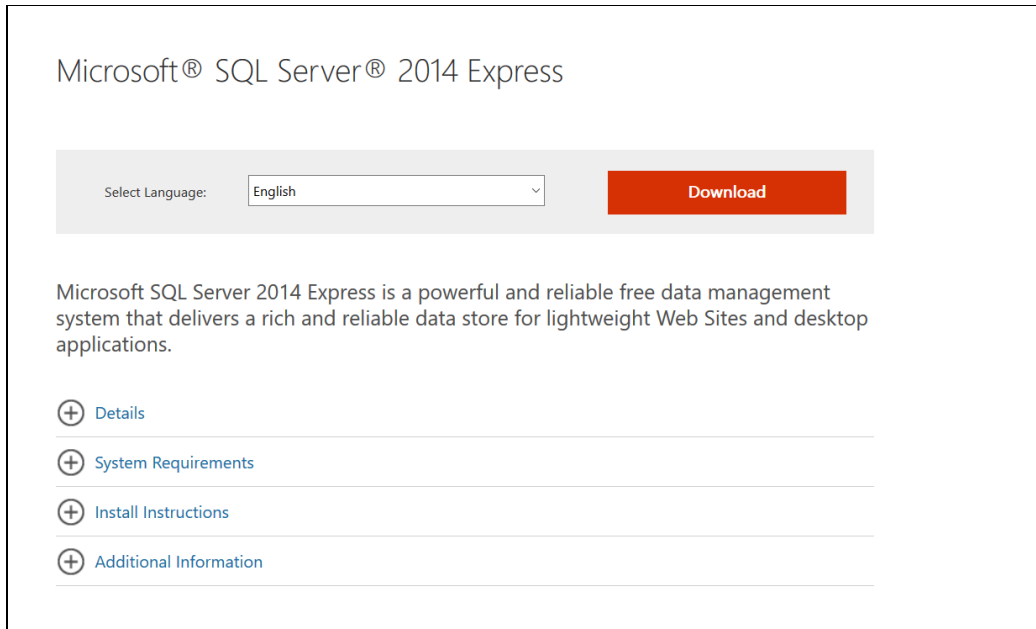
| Analýza | | | | |
|--|------|------|------|------|
| Funkcia | 2016 | 2014 | 2012 | 2008 |
| PowerPivot for Excel | Áno | Áno | Áno | Áno |
| Integration services managed as a server | Áno | Áno | Áno | Nie |
| Hadoop connector via Apache Sqoop | Áno | Áno | Áno | Áno |
| Tabular BI semantic model | Áno | Áno | Áno | Áno |
| Master data services | Áno | Áno | Áno | Áno |
| Data quality services | Áno | Áno | Áno | Nie |

Tabuľka 6 Funkcie zamerané na zlepšenie analýzy

II. PRAKTICKÁ ČÁST

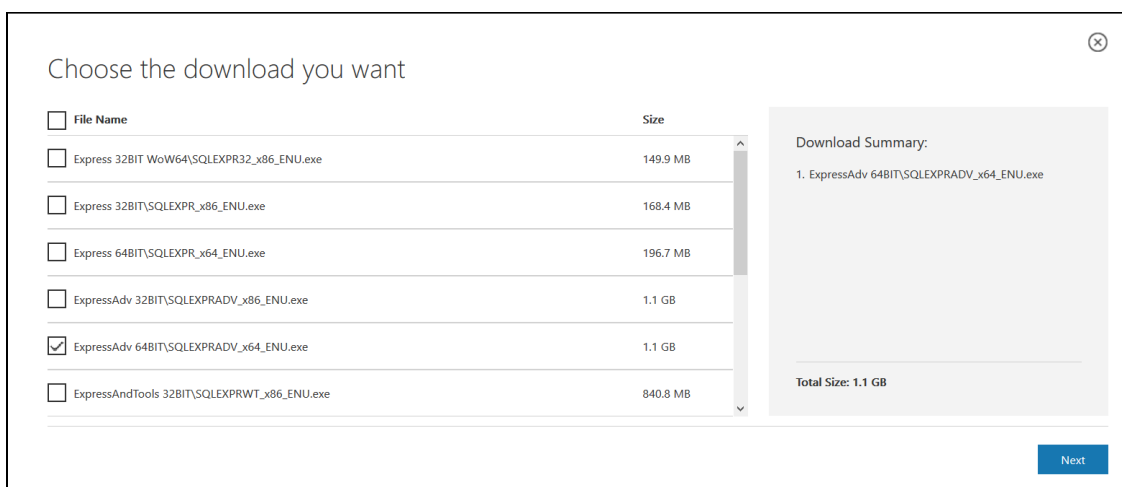
4 MICROSOFT SQL SERVER 2014

4.1 Inštalácia Microsoft SQL Server 2014 Express



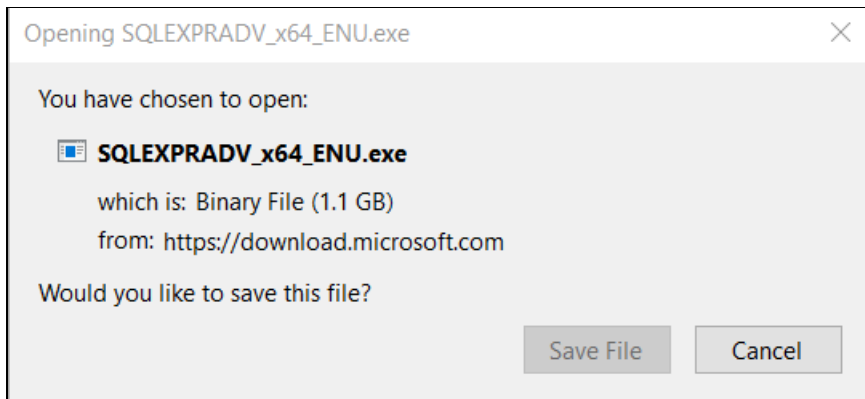
Obrázok 4 Voľba jazyka a detaily inštalácie

Verzia SQL Server 2014 Express je zadarmo dostupná priamo na webovej stránke Microsoftu : <https://www.microsoft.com/en-US/download/details.aspx?id=42299> . Po kliknutí na črevné tlačidlo *Download* pokračujeme na výber verzie.



Obrázok 5 Voľba 32 a 64 bitovej verzie

Vyberáme verziu Microsoft SQL Server Express with Services, ktorá je pod označením *ExpressAdv 64BIT* pre 64bitové systémy alebo *ExpressAdv 32BIT* pre 32bitové systémy. Označíme box s verziou ktorá vyhovuje nášmu systému a pokračujeme kliknutím na tlačidlo *Next*.



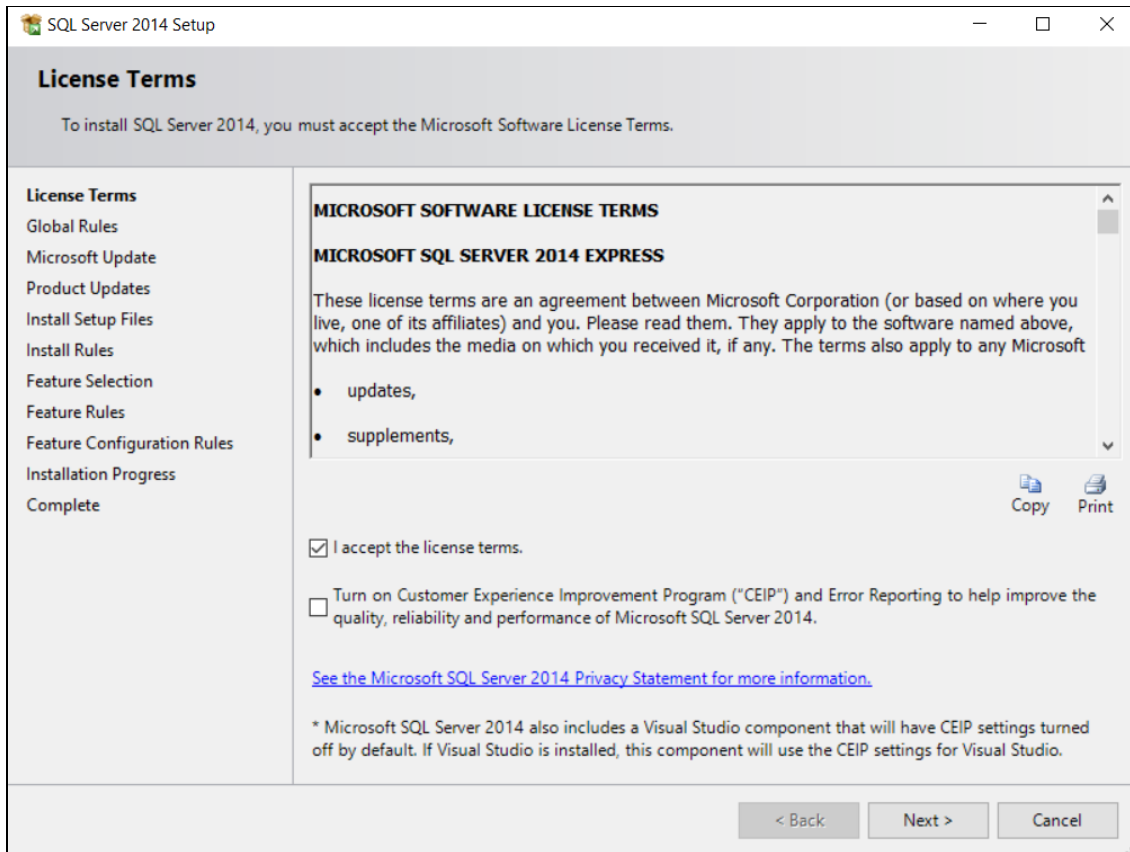
Obrázok 6 Uloženie inštalačného súboru

Skontrolujeme či verzia zodpovedá naším požiadavkám a pomocou tlačidla *Save File* uložíme inštalačný súbor na disk. Po stiahnutí spustíme inštalačný súbor a pokračujeme v inštalácii pomocou sprievodcu pre inštaláciu SQL Server.



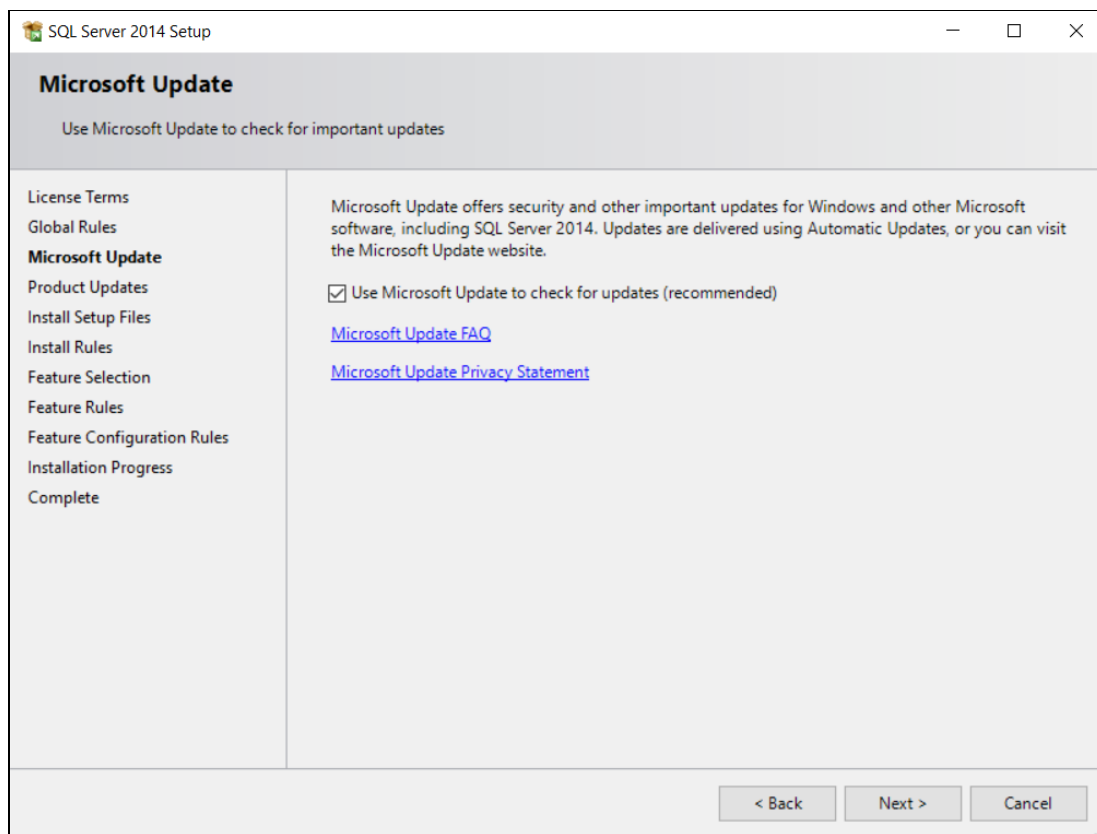
Obrázok 7 Výber inštalácie Microsoft SQL Server 2014

V prvom kroku vyberieme prvú možnosť : *New SQL Server stand-alone installation*.



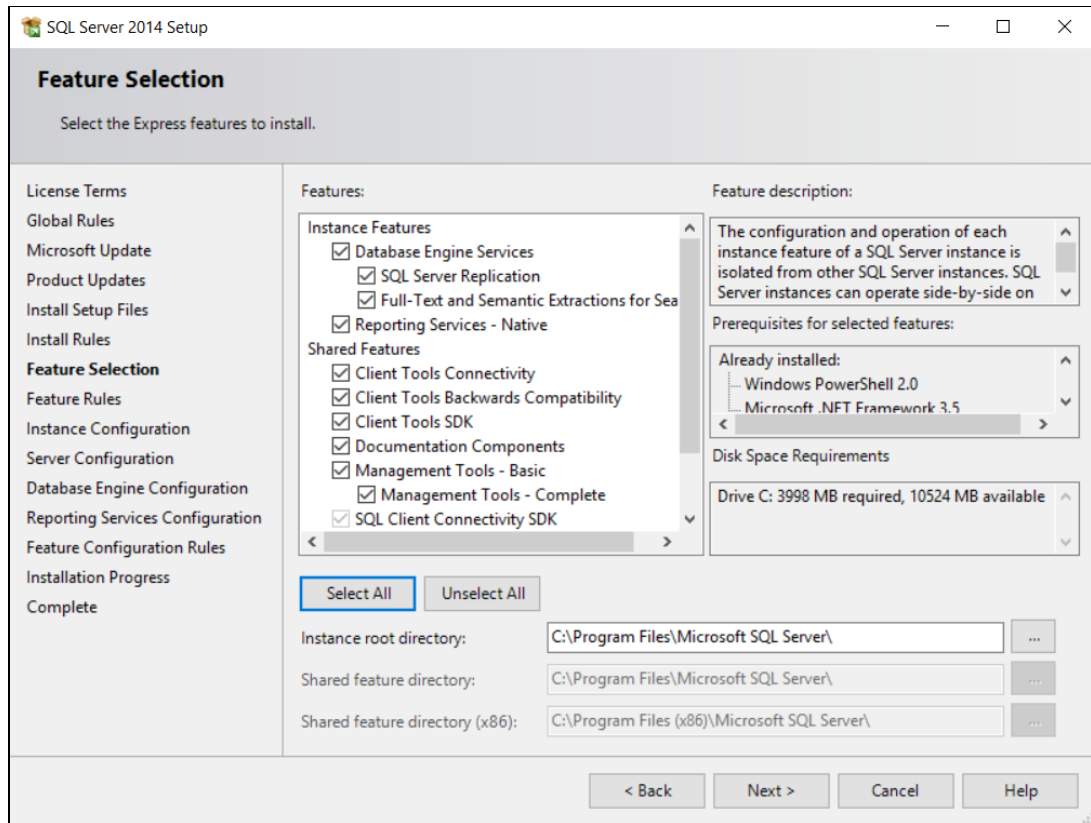
Obrázok 8 Odsúhlasenie licenčných podmienok.

V ďalšom kroku odsúhlasíme licenčné podmienky pre používanie softwaru Microsoft SQL server 2014 Express. Zaklikneme okno : *I accept the license terms* a pokračujeme kliknutím na tlačidlo *Next*.



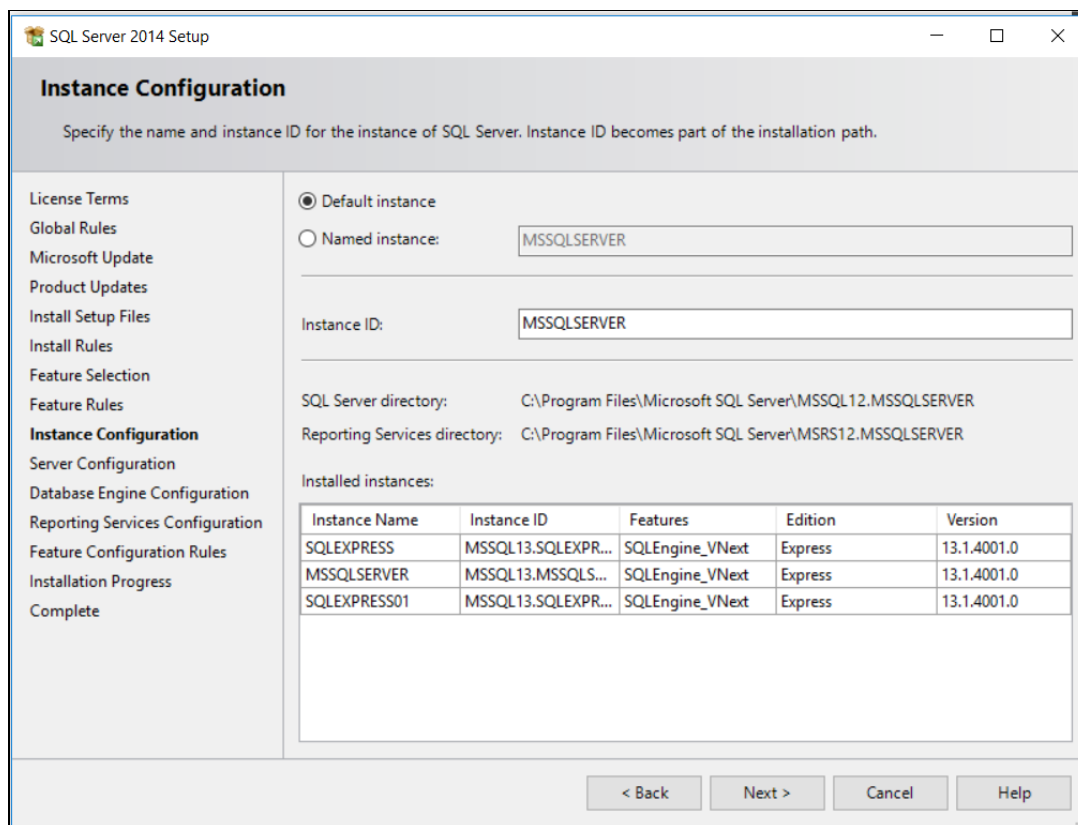
Obrázok 9 Povolenie automatických aktualizácií

Potvrdíme automatické sťahovanie aktualizácií, ktoré nám zaisti vyššiu bezpečnosť a stabilitu programu a pokračujeme v inštalácii pomocou tlačidla *Next*.



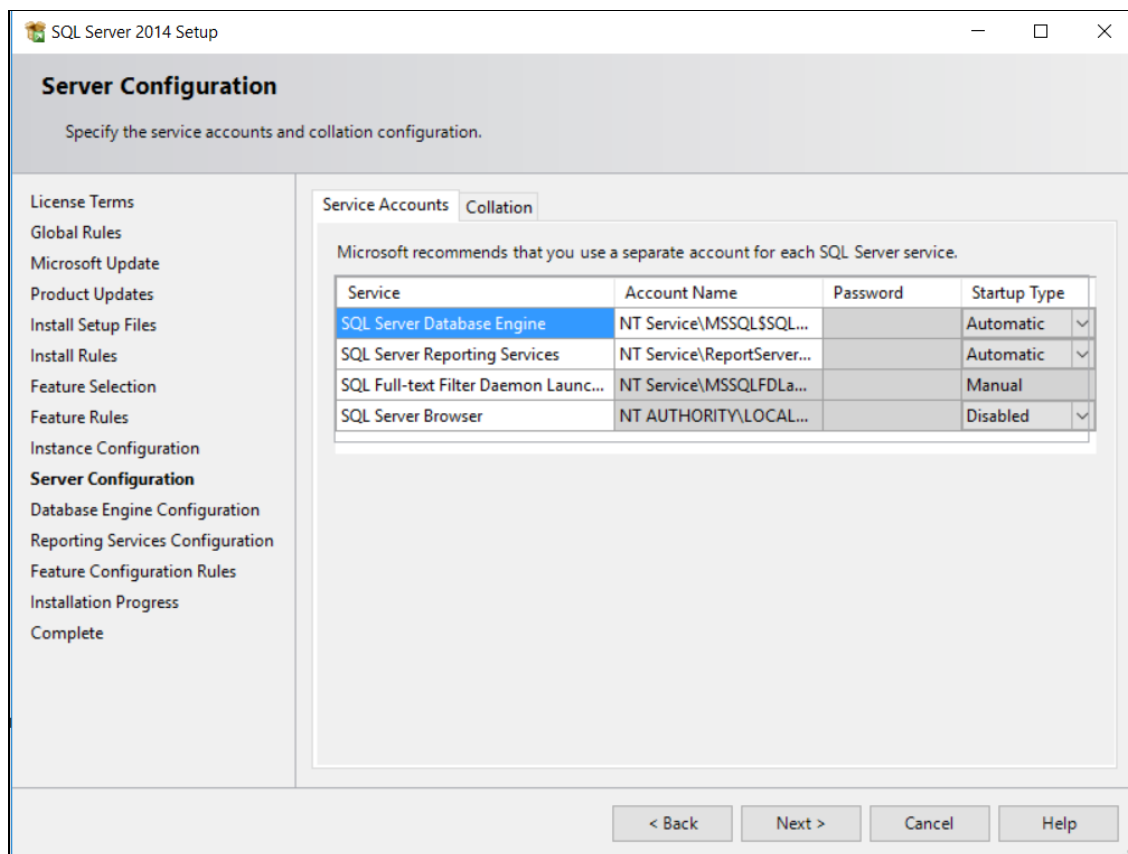
Obrázok 10 Výber doplnkov k inštalácii

V nasledujúcom kroku vyberieme doplnky ktoré chceme nainštalovať. Zaklikneme tlačidlo *Select All* a vyberiem tak všetky dostupne doplnky.



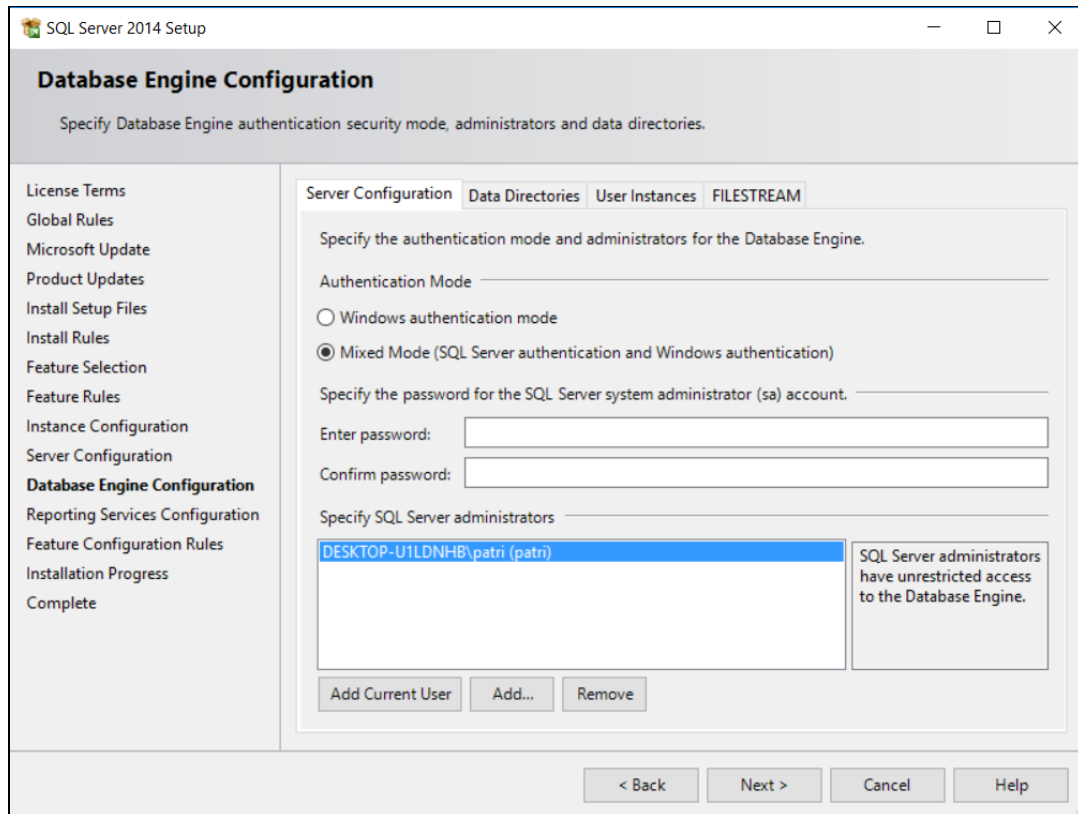
Obrázok 11 Konfigurácia inštancie.

Konfiguráciu inštancie necháme na základných hodnotách preto zaklikneme možnosť *Default instance*. Pokračujeme ďalej tlačidlom *Next*.



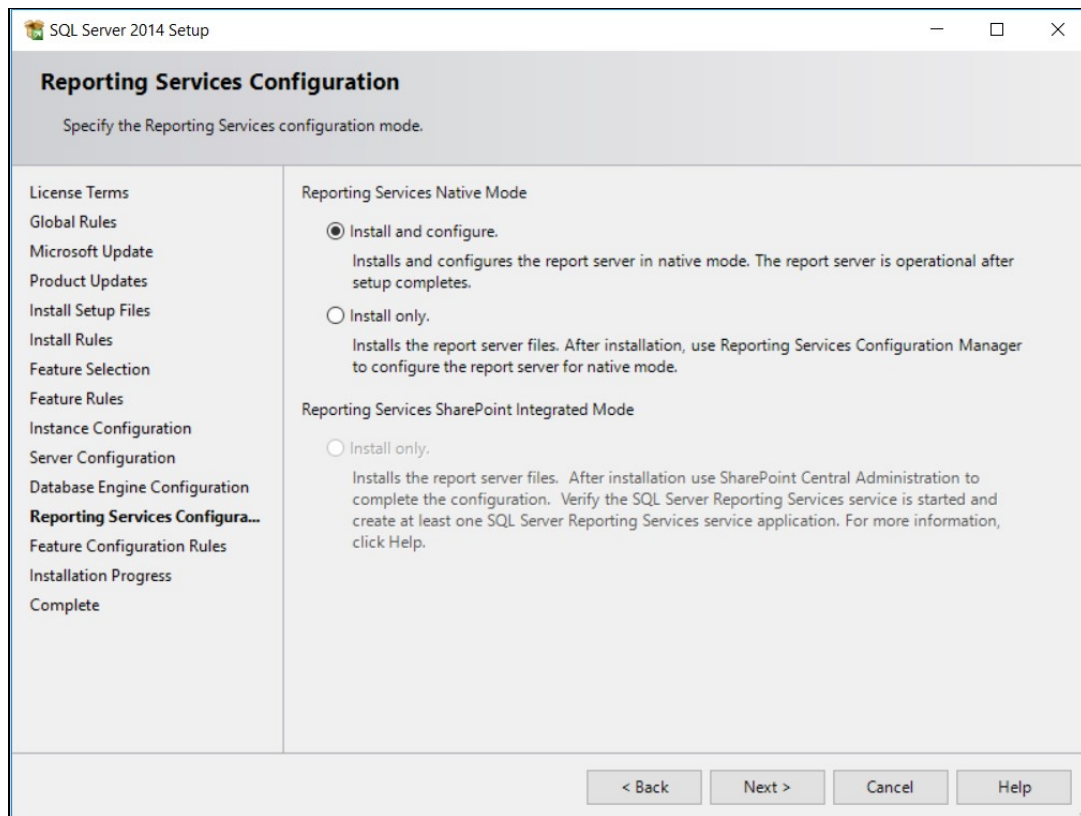
Obrázok 12 Konfigurácia servera

V nastaveniach servera nič nemeníme a necháme všetko v predvolených nastaveniach a pokračujeme ďalej tlačidlom *Next*.



Obrázok 13 Autorizácia pomocou hesla

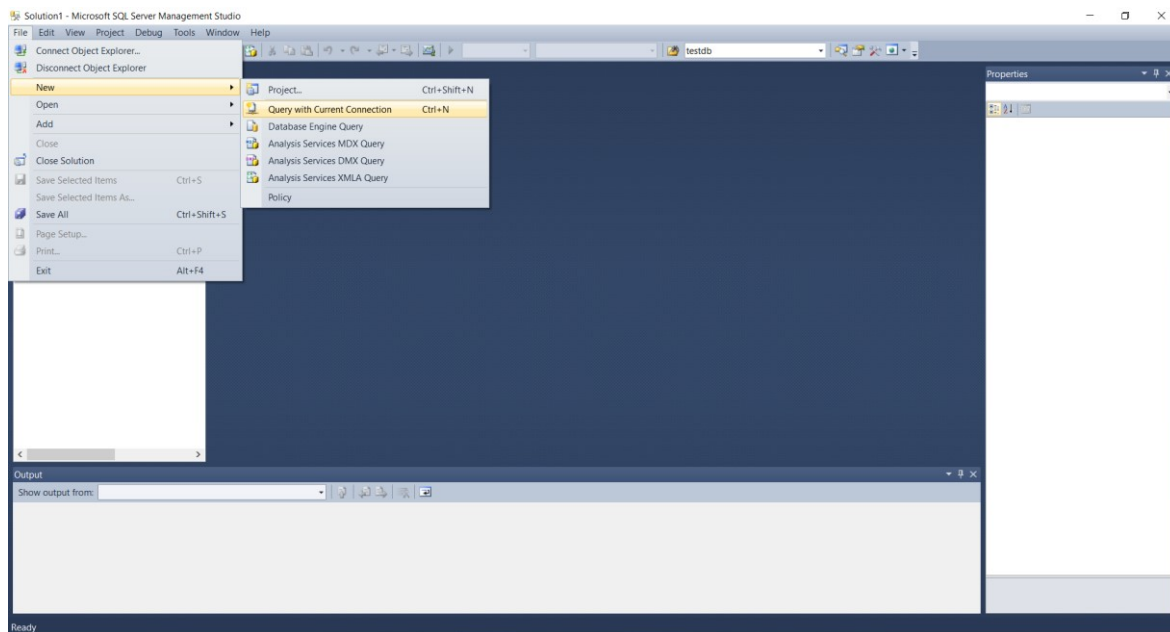
V nasledujúcom kroku nastavíme overovanie užívateľa. Pre naše potreby vyberieme možnosť *Mixed Mode*. Zadáme heslo a opätovným zadaním potvrdíme zvolené heslo. Pridáme administrátora tak že zvolíme aktuálne prihláseného užívateľa na počítači. A pokračujeme ďalej pomocou tlačidla *Next*.



Obrázok 14 Posledný krok inštalácie

V poslednom kroku vyberieme možnosť *Install only* a tlačidlom *Next* spustíme inštaláciu. Samotná inštalácia potrvá niekoľko minút. Po jej dokončení dostaneme oznámenie o úspešnej inštalácii a môžeme začať pracovať.

5 PRÁCA S MICROSOFT SQL SERVER 2014



Obrázok 15 Rozhranie Microsoft SQL 2014 Express

5.1 Tvorba objektov

5.1.1 Tvorba databáze

Pri tvorbe databázy je nutné uviesť meno databáze ostatné možnosti deklarácie tabuľky sú voliteľné. Kľúčový príkaz na tvorbu objektu je CREATE a nasleduje voľba objektu čo v prípade tvorby databázy je DATABASE.

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification] ...

create_specification ::=
    [DEFAULT] CHARACTER SET [=] charset_name
    | [DEFAULT] COLLATE [=] collation_name
```

5.1.2 Tvorba tabuľky

Pri tvorbe tabuľky je nutné uviesť názov tabuľky a postupne definovať názvy a dátové typy stĺpcov ktoré sa majú v tabuľke nachádzať. Taktiež je možné pridať obmedzenia k jednotlivým stĺpcom alebo k celej tabuľke. Tieto obmedzenia sú voliteľné

```

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [partition_options]

create_definition ::=
    col_name column_definition
  | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
    [index_option] ...
  | {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
    [index_option] ...

column_definition ::=
    data_type [NOT NULL | NULL] [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
    [COMMENT 'string']

data_type ::=
    pozri "Dátové typy"

index_type ::=
    USING {BTREE | HASH}

```

5.2 Práca s tabuľkou

5.2.1 SELECT

Príkaz SELECT je jeden zo základných príkazov v jazyku SQL. Slúži na zobrazenie dát z jednej alebo viacerých tabuliek. Pričom sa používa spolu s príkazom FROM, ktorý definuje z ktorých tabuliek sa budú dáta zobrazovať. [mistrovstvi]

```

SELECT [ALL | DISTINCT | DISTINCTROW ]
    select_expr
    [FROM tabulky
    [WHERE where_condition]
    [GROUP BY {col_name | expr | POSITION}[ASC | DESC], ... [WITH ROLLUP]]
    [HAVING where_condition]
    [ORDER BY {col_name | expr | POSITION} [ASC | DESC], ...]

```

5.2.2 WHERE

Príkaz WHERE slúži na filtrovanie riadkov podľa zvolenej vyhľadávacej podmienky. Vyhľadávacie podmienky môžeme rozdeliť na 5 základných typov :

- porovnanie – porovnáva hodnotu dvoch výrazov
- rozsah – zisťuje či sa hodnota výrazu nachádza vo zvolenom rozsahu
- množina – zisťuje či sa hodnota výrazu rovná niektorej hodnote z množiny
- porovnanie vzorov – zisťuje či zadaný reťazec obsahuje určitý vzorec
- NULL – zisťuje či stĺpec obsahuje hodnotu NULL [mistrovstvi]

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
  select_expr
  [FROM tabulky
   [WHERE where_condition]
   [GROUP BY {col_name | expr | POSITION}[ASC | DESC], ... [WITH
ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | POSITION} [ASC | DESC], ...]
```

5.2.3 GROUP BY

Príkaz GROUP BY umožňuje zoskupovať údaje podľa určenej podmienkovej hodnoty. Dotaz SELECT, ktorý obsahuje príkaz GROUP BY potom nazývame zoskupený dotaz. Takýto zoskupený dotaz potom pre každú podmienku vytvorí jednoriadkový súhrn. [data-baze]

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
  select_expr
  [FROM tabulky
   [WHERE where_condition]
   [GROUP BY {col_name | expr | POSITION}[ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | POSITION} [ASC | DESC], ...]
```

5.2.4 HAVING

Príkaz HAVING sa používa s príkazom GROUP BY pre obmedzenie skupín, ktoré sa majú objaviť vo výslednej tabuľke. Na rozdiel od príkazu WHERE, ktorý filtruje jednotlivé riadky vstupujúce do tabuľky konečných výsledkov, príkaz HAVING filtruje skupiny ktoré vstupujú do tabuľky konečných výsledkov. [mistrovstvi]

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
  select_expr
  [FROM tabulky
   [WHERE where_condition]
   [GROUP BY {col_name | expr | POSITION}[ASC | DESC], ... [WITH ROLLUP]]
   [HAVING where_condition]
  [ORDER BY {col_name | expr | POSITION} [ASC | DESC], ...]
```

5.2.5 ORDER BY

Príkaz ORDER BY umožňuje zoradenie výsledku dotazu podľa ktoréhokol'vek stĺpca alebo kombinácie stĺpcov bez ohľadu nato či sa dané stĺpce nachádzajú vo výsledku. Riadky je možné zoradiť zostupne alebo vzostupne pomocou podmienok DESC a ASC. V prípadoch keď stĺpec obsahuje viacero rovnakých riadkov je môžeme určiť primárny a sekundárny kľúč radenia.[mistrovstvi]

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
  select_expr
  [FROM tabulky
  [WHERE where_condition]
  [GROUP BY {col_name | expr | POSITION}[ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | POSITION} [ASC | DESC], ...]
```

5.2.6 INSERT

Príkaz INSERT slúži na vkladanie nových riadkov do tabuľky. Pri vkladaní nového riadku musíme uviesť názov tabuľky do ktorej budeme daný riadok vkladat' a následne hodnoty ktoré budeme vkladat' do jednotlivých stĺpcov. [mistrovstvi]

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name [(col_name, ...)]
  {VALUES | VALUE} ({expr | DEFAULT}, ...), (...), ...
```

5.2.7 UPDATE

Príkaz UPDATE slúži na modifikáciu existujúcich dát v tabuľke. Pri modifikácii je potrebné uviesť tabuľku v ktorej budeme dáta modifikovať. Následne je nutné pomocou príkazu SET určiť ktoré stĺpce v danej tabuľke budeme modifikovať. Ak nechceme modifikovať všetky riadky môžeme použiť príkaz WHERE a určiť podmienku ktoré riadky budú modifikované.

[mistrovstvi]

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
  SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...
  [WHERE where_condition]
  [ORDER BY ...]
```

5.2.8 DELETE

Príkaz DELETE slúži na mazanie riadkov tabuľky. Princíp fungovania je takmer totožný s príkazom UPDATE ale na rozdiel od príkazu UPDATE dáta nemení ale zmaže ich. Rovnako ako pri UPDATE je možné použiť príkaz WHERE a určiť podmienku, ktoré riadky budú zmazane. [mistrovstvi]

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
  [WHERE where_condition]
  [ORDER BY ...]
```

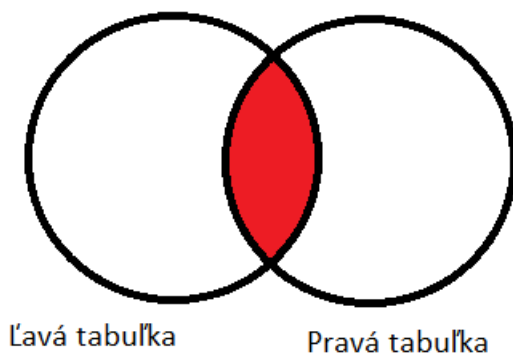
5.2.9 DROP TABLE

Príkaz DROP TABLE slúži na odstránenie celej štruktúry tabuľky spolu s dátami, ktoré obsahovala. Na vykonanie príkazu stačí zadať samotný príkaz DROP TABLE a názov tabuľky ktorú chceme odstrániť. [databaze]

```
DROP [TEMPORARY] TABLE [IF EXISTS]
tbl_name [, tbl_name] ...
```

5.2.10 INNER JOIN

INNER JOIN predstavuje vnútorné spojenie, ktoré sa vykoná na základe jedného alebo viacerých spoločných atribútov. Výsledné spojenie bude obsahovať len dáta ktoré majú v spojovacích atribútoch úplnú zhodu. Výsledok tohto spojenia je totožný s príkazom WHERE.



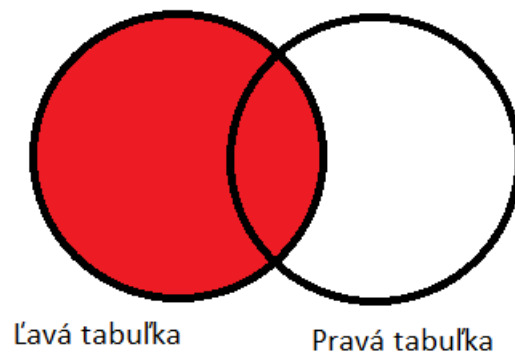
Obrázok 16 Grafické znázornenie funkcie INNER JOIN

5.2.11 OUTER JOIN

OUTER JOIN sa používa na vonkajšie spojenie dát pričom sú do výsledného spojenia zahrnuté aj dáta ktoré nespĺňajú spojovaciu podmienku. Ktoré dáta sa zachovávajú závisí od toho ktorú z variantov OUTER JOIN použijeme.

5.2.11.1 LEFT OUTER JOIN

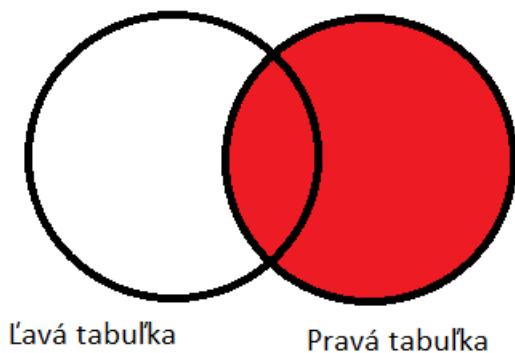
V prípade LEFT OUTER JOIN sa dáta z ľavej tabuľky zachovávajú aj keď nemajú väzbu s dátami v pravej tabuľky.



Obrázok 17 Grafické znázornenie funkcie LEFT OUTER JOIN

5.2.11.2 RIGHT OUTER JOIN

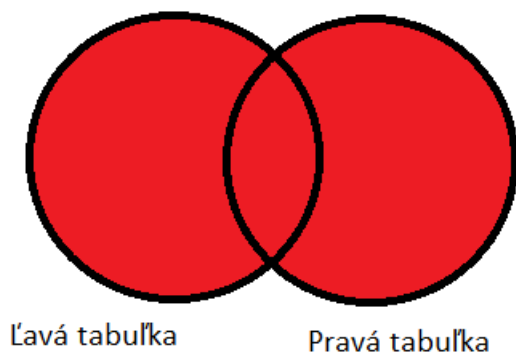
V prípade RIGHT OUTER JOIN sa dáta z pravej tabuľky zachovávajú aj keď nemajú väzbu s dátami v ľavej tabuľke.



Obrázok 18 Grafické znázornenie funkcie RIGHT OUTER JOIN

5.2.11.3 FULL OUTER JOIN

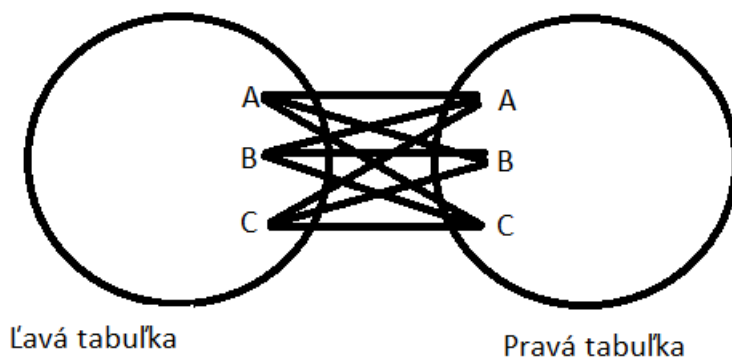
FULL OUTER JOIN predstavuje kombináciu LEFT OUTER JOIN a RIGHT OUTER JOIN. Dáta z ľavej aj pravej tabuľky sa zachovávajú aj keď nemajú žiadnu väzbu medzi sebou.



Obrázok 19 Grafické znázornenie funkcie FULL OUTER JOIN

5.2.12 CROSS JOIN

Príkaz CROSS JOIN sa používa v prípade, keď je nutné priradiť každému riadku prvej tabuľky hodnotu každého riadku druhej tabuľky. Toto spojenie predstavuje kartezsky súčin.

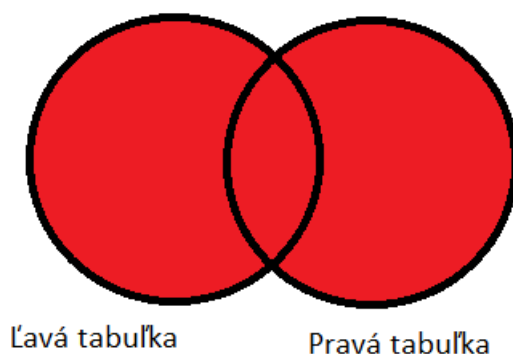


Obrázok 20 Grafické znázornenie funkcie CROSS JOIN

5.2.13 UNION

Union slúži na výpis dvoch alebo viacerých tabuliek do jednej spoločnej tabuľky, tak že automaticky eliminuje duplicitné riadky bez ohľadu odkiaľ riadky pochádzajú. Podmienkou pre úspešne spojenie tabuliek je, že spojované tabuľky musia mať rovnaký počet stĺpcov s rovnakým dátovým typom ale ich názvy sa nemusia zhodovať. Po spojení sa zachová názov stĺpca prvej tabuľky. V prípade, že nechceme pri spájaní dvoch tabuliek eliminovať žiadne riadky je nutné použiť príkaz UNION ALL, ktorý všetky duplicitné riadky zachová.

```
SELECT ...
UNION [ALL | DISTINCT] SELECT ...
[UNION [ALL | DISTINCT] SELECT ...]
```

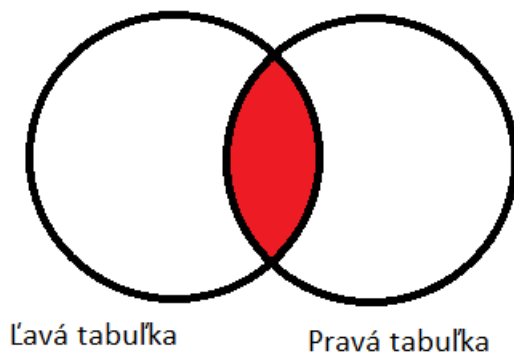



Obrázok 21 Grafické znázornenie funkcie UNION

5.2.14 INTERSECT

Príkaz INTERSECT slúži na vybratie prieniku dvoch alebo viacerých tabuliek. Čo znamená že vráti všetky údaje ktoré sa aspoň raz nachádzajú v každej tabuľke.

```
SELECT ...  
(INTERSECT | EXCEPT | UNION [ALL | DISTINCT]) SELECT ...  
[(INTERSECT | EXCEPT | UNION [ALL | DISTINCT]) SELECT ...]  
[ORDER BY [column [, column ...]]]  
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

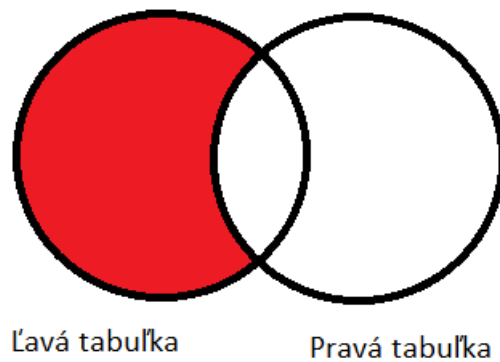


Obrázok 22 Grafické znázornenie funkcie INTERSECT

5.2.15 EXCEPT

Príkaz EXCEPT odstráni z prvej tabuľky údaje ktoré sa zhodujú s údajmi v druhej tabuľke.

```
SELECT ...  
(INTERSECT | EXCEPT | UNION [ALL | DISTINCT]) SELECT ...  
[(INTERSECT | EXCEPT | UNION [ALL | DISTINCT]) SELECT ...]  
[ORDER BY [column [, column ...]]]  
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
```



Obrázok 23 Grafické znázornenie funkcie EXCEPT

5.3 Agregáčné funkcie

Agregáčné funkcie vracajú jednoduchú hodnotu pre jednotlivé stĺpce tabuľky. [mistrovstvi]

5.3.1 AVG

Funkciu AVG je možné použiť len na numerické polia. Vracia priemernú hodnotu všetkých numerických dát obsiahnutých vo vybranom stĺpci. Ak je potrebné pri výpočte priemeru vynechať duplicitné hodnoty v stĺpci je možné použiť funkciu AVG v kombinácii s funkciou DISTINCT. [mistrovstvi]

5.3.2 MIN, MAX

Funkcie MIN a MAX je možné použiť na numerické a aj na nenumerické polia. Funkcie vrátia najmenšiu alebo najväčšiu hodnotu ktorá sa nachádza vo zvolenom stĺpci. [mistrovstvi]

5.3.3 SUM

Funkcia SUM vracia súčet všetkých hodnôt vo vybranom stĺpci. Funkciu je možné použiť len na numerické polia. Pomocou príkazu DISTINCT je možné odstrániť duplicitné hodnoty a tak ovplyvniť výsledok funkcie ak je to potrebné. [mistrovstvi]

5.3.4 COUNT

Funkcia COUNT vracia počet hodnôt vo vybranom stĺpci tak, že najprv eliminuje riadky, ktoré obsahujú hodnotu NULL. Ak chceme spočítať všetky riadky aj tie ktoré obsahujú hodnoty NULL musíme použiť funkciu v nasledujúcom tvare : COUNT(*). [mistrovstvi]

5.4 Vstavané funkcie

5.4.1 Matematické funkcie

$ABS(x)$ – vráti absolútnu hodnotu z čísla x .

$SIN(x)$ – vráti sínus uhla x . Uhol x je zadaný v radiánoch.

$COS(x)$ – vráti kosínus uhla x . Uhol x je zadaný v radiánoch.

$CEIL(x)$ – zaokrúhli číslo x smerom na hor.

$EXP(x)$ – vráti hodnotu e umocnenú na x .

$FLOOR(x)$ – zaokrúhli číslo smerom dole.

$FORMAT(x,n)$ – používa sa na formátovanie čísla x na n desatinných miest.

$LOG(x)$ – vráti prirodzený logaritmus z čísla x .

$LOG10(x)$ – vráti desiatkový logaritmus z čísla x .

$PI()$ – vráti hodnotu čísla π .

$POWER(x,y)$ – vráti hodnotu x umocnenú na y .

$ROUND(x,d)$ – zaokrúhli číslo x na d desatinných miest.

$SQRT(x)$ – vráti druhú odmocninu čísla x . Ak je x záporne vráti *NULL*.

$TAN(x)$ – vráti tangens uhla x v radiánoch.

$RAND()$ – generuje náhodne číslo v intervale od 0-1.

5.4.2 Textové funkcie

$CHAR(x,...)$ – prevedie každý znak x na znak podľa použitej znakovej tabuľky.

$LOWER(str)$ – prevedie reťazec str na malé písmena.

$UPPER(str)$ – prevedie reťazec str na veľké písmena.

$LTRIM(str)$ – zo začiatku reťazca str odstráni nečitateľné znaky a medzery.

$RTRIM(str)$ – z konca reťazca str odstráni nečitateľné znaky a medzery.

$TRIM(str)$ – zo začiatku a konca reťazca str odstráni nečitateľne znaky a medzery.

5.4.3 Dátumové funkcie

ADDDATE(*date, days*) – pripočíta k zadanému dátumu *date* počet dní *days*.

ADTIME(*t1, t2*) – pripočíta k času *t1* čas *t2*.

CURRDATE() – vráti aktuálny dátum vo formáte RRRR-MM-DD ak je v funkcia použitá pre reťazec alebo vo formáte RRRRMMDD ak je v číselnom kontexte.

CURTIME() – vráti aktuálny čas vo formáte HH:MM:SS alebo HHMMSS v závislosti na formáte v ktorom je funkcia použitá.

DATEDIFF(*d1, d2*) – vráti rozdiel v dňoch medzi dátumom *d1* a *d2*.

DATE(*d*) – z dátumu a času *d* vyberie len dátum.

DAYNAME(*d*) – z dátumu *d* určí názov dna

DAYOFWEEK(*d*) – vráti číslo dňa v týždni z dátumu *d*.

DAYOFYEAR(*d*) – z dátumu *d* vráti deň v roku.

HOUR(*t*) – vráti z dátumu a času *d* len čas v hodinách

NOW() – vráti aktuálny dátum a čas.

SYSDATE() – vráti hodnotu aktuálneho systémového dátumu a času.

WEEK(*d*) – z dátumu *d* určí číslo týždňa

5.5 Dátové typy

Dátové typy sa používajú na definovanie atribútov a relácii v tabuľkách. Preto v databázových systémoch rozlišujeme 5 základných dátových typov. Celočíselné, desatinné, dátumové, znakové a binárne. Existuje aj špeciálny dátový typ *NULL*, ktorý reprezentuje nepriradenú hodnotu.

5.5.1 Celočíselné dátové typy

INT , slúži na uloženie celého čísla v intervale od -2147483648 do 2147483647.

BIGINT, slúži na uloženie celého čísla v rozsahu -9223372036854775808 do 9223372036854775807

5.5.2 Desatinné datové typy

FLOAT(n,d), kde n je počet čísel pred desatinou čiarkou a d je počet čísel za desatinnou čiarkou. Slúži na uloženie čísla s desatinnou časťou.

DOUBLE(n,d), kde kde n je počet čísel pred desatinou čiarkou a d je počet čísel za desatinnou čiarkou. Slúži na uloženie čísla s desatinnou časťou s dvojitou presnosťou.

5.5.3 Dátumové datové typy

DATE(), slúži na ukladanie dátumu vo formáte RRRR-MM-DD ,kde RRRR je rok , MM je mesiac a DD predstavuje deň.

DATETIME, slúži na ukladanie kombinácie dátumu a času vo formáte RRRR-MM-DD HH:MM:SS , kde RRRR je rok , MM je mesiac, DD predstavuje deň, HH sú hodiny , MM minúty a SS sú sekundy.

TIME(), slúži na ukladanie času vo formáte HH:MM:SS, kde HH sú hodiny , MM minúty a SS sú sekundy.

5.5.4 Znakové datové typy

CHAR(n), kde n je počet znakov. Používa sa na uloženie reťazca znakov pevnej dĺžky. Maximálny počet znakov je 255.

VARCHAR(n), kde n je počet znakov. Používa sa na uloženie reťazca premenlivej dĺžky. Maximálny počet znakov je 255 ale pri prekročení sa zmení na datový typ TEXT.

TEXT , slúži na uloženie reťazca s maximálnou dĺžkou 65535 znakov.

5.5.5 Binárne datové typy

BINARY(n), kde n je pamäťová náročnosť v rozsahu od 1 po 8000 B. Slúži na ukladanie dát ako sú obrázky, hudobne súbory, dokumenty.

VARBINARY(n), kde n je pamäťová náročnosť v rozsahu od 1 po 8000 B. Slúži na ukladanie dát ako sú obrázky, hudobne súbory, dokumenty.

VARBINARY(n), kde n je pamäťová náročnosť v rozsahu od 1 po 2147483648 B. Slúži na ukladanie dát ako sú obrázky, hudobne súbory, dokumenty.

5.6 Programovanie SQL

SQL umožňuje vytvárať premenné , funkcie , podmienky a cykly.

5.6.1 Vytvorenie premennej

Pri vytváraní premennej môžeme pridať hodnotu premennej tromi rôznymi spôsobmi. Pri deklarácii premennej, použitím SET alebo pomocou SELECT . Ak nepriradíme žiadnu hodnotu SQL priradí k danej premennej automaticky hodnotu *NULL*.

```
variable_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial_value]
```

5.6.2 Vytváranie funkcie

Základnú časť funkcie tvoria príkazy BEGIN a END ktoré ohraničujú telo funkcie.

```
CREATE [OR REPLACE] FUNCTION function_name  
[(parameter_name [IN | OUT | IN OUT] type [, ...])]  
RETURN return_datatype  
{IS | AS}  
BEGIN  
    < function_body >  
END [function_name];
```

5.6.3 Vytváranie podmienky

Telo podmienky je tvorené z troch príkazov IF určuje podmienku z ktorej vyplýva pravda alebo nepravda. V prípade, že výsledok podmienky je pravdivý vykoná sa príkaz za THEN. V prípade, že podmienka nie je splnená vykoná sa príkaz za ELSE. Celá podmienka sa ukončí príkazom END IF.

```
IF condition THEN statement  
ELSE statement  
END IF
```

5.6.4 Vytvorenie rozhodovacej funkcie

V prípade, že potrebujeme reagovať na rôzne hodnoty rôznymi príkazmi použijeme príkaz CASE. CASE definuje aký príkaz sa vykoná pre vybranú hodnotu. Ak vybraná hodnota nezodpovedá žiadnej možnosti je definovaný príkaz pre ostatné hodnoty. Celá funkcia je ukončená príkazom END CASE.

```
CASE case_value
  WHEN when_value THEN statement
  WHEN when_value THEN statement
  ELSE statement
END CASE
```

5.6.5 Vytvorenie cyklu

Pre vytvorenie cyklu je nutné zadať pomocou SET hodnotu. Následne pomocou WHILE určiť podmienku pri dosiahnutí akej hodnoty sa cyklus ukončí. Nasleduje príkaz ktorý sa bude opakovane vykonávať a inkrementačný alebo dekrementačný posun. Celý cyklus sa ukončí príkazom END WHILE.

```
SET i=value;
WHILE i > 0 DO
  statement
  SET i = i - 1;
END WHILE;
```

5.6.6 Trigger

Trigger definuje akciu ktorá sa ma vykonať keď v aplikácii dôjde k určitej udalosti. Medzi udalosti ktoré môžu spustiť trigger patri:

- vloženie údajov do tabuľky príkazom INSERT
- zmazanie údajov z tabuľky príkazom DELETE
- aktualizácia údajov v tabuľke príkazom UPDATE

Taktiež je možné definovať či sa trigger spustí pred udalosťou BEFORE , po udalosti AFTER alebo namiesto udalosti INSTEAD OF.

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
  Declaration-statements
BEGIN
  Executable-statements
EXCEPTION
  Exception-handling-statements
END;
```

5.6.7 Procedúry

Procedúra je sada príkazov, ktoré umožňujú vykonávať opakujúcu sa činnosť jednoduchšie a rýchlejšie. Procedúra uložená na serveri je skompilovaná, čo znamená že takto uloženú procedúru spravuje realizačný plán. Vďaka tomu dokáže server spustiť a vykonať operáciu ako keby užívateľ zadával príkaz po príkaze. V prípade použitia procedúry nie je nutné použiť jednotlivé príkazy, postačí použiť vopred nadefinovanú procedúru ktorá obsahuje všetky potrebné príkazy. Takto nadefinované procedúry pomáhajú unifikovať požiadavky od užívateľov. Vďaka tomu prispievajú k bezpečnosti servera keďže je možné nastaviť užívateľa, ktorý ma oprávnenie spustiť danú procedúru. Procedúry môžu vykonávať väčšinu bežných aktivít:

- vyberanie dát pomocou príkazu SELECT
- vkladanie aktualizácii pomocou príkazu UPDATE
- vytváranie , používanie a mazanie dočasných tabuliek pomocou JOIN , UNION
- matematické a štatistické výpočty

```
CREATE
  [DEFINER = { USER | CURRENT_USER }]
  PROCEDURE sp_name ([proc_parameter[,...]])
  routine_body

proc_parameter ::=
  [ IN | OUT | INOUT ] param_name TYPE

TYPE ::=
  TubovoIný dátový SQL typ.

routine_body ::=
  SQL príkaz
```


ZÁVĚR

Cieľom práce bolo vytvoriť učebnú pomôcku pre výuku anglicky hovoriacich študentov na predmet zameraný na databázové systémy bez predošlej znalosti tejto tematiky.

Teoretická časť sa zameriava na modely databázových systémov ich históriu a vývoj. Prehľadne zobrazuje históriu a štruktúru SQL jazyka. Nasledovne predstavuje najavšiu verziu vývojového prostredia Microsoft SQL Server 2016 jej nové funkcie a služby. Porovnáva medzi sebou jednotlivé edície a taktiež predchádzajúce verzie.

Praktická časť sa zaberá prácou vo vývojovom prostredí Microsoft SQL Server 2014 Express. Postupne prechádza cez jednotlivé príkazy a funkcie a slúži ako podklad pre prezentácie určené na cvičenia z databázových systémov.

V prezentáciách sa nachádzajú konkrétne príklady pre podstatnú časť príkazov spomenutých v praktickej časti. Zadania pre samostatnú činnosť s následným príkladom riešenia daného zadania zväčša doplnene o tabuľky alebo krátky komentár vysvetľujúci princíp riešenia.

Praktická časť je taktiež doplnená o 60 testovacích otázok vhodných na overenie znalosti po absolvovaní výuky, ktoré je možné nahrať do školského systému MOODLE.

SEZNAM POUŽITÉ LITERATURY

- [1] Česká terminologická databáze knihovnictví a informační vědy (TDKIV). *Data-báze Národní knihovny ČR* [online]. 2014 [cit. 2017-05-23]. Dostupné z: http://aleph.nkp.cz/F/?func=direct&doc_number=000000089&local_base=KTD
- [2] Databáze. *Historii* [online]. 2010 [cit. 2017-05-23]. Dostupné z: <http://www.databaze.chytrak.cz/historie.htm>
- [3] Databáze. *Databázové modely* [online]. 2010 [cit. 2017-05-23]. Dostupné z: <http://www.databaze.chytrak.cz/modely.htm>
- [4] Relační vs. objektově. *Relační vs. objektově-relační vs. objektové databáze* [online]. [cit. 2017-05-23]. Dostupné z: <http://www.fi.muni.cz/~xbatko/oracle/compare.html>
- [5] CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [6] Databázové a informačné systémy. *Jazyk SQL* [online]. 2005 [cit. 2017-05-23]. Dostupné z: <http://spseke.sk/tutor/prednasky/dbs/sql.html>
- [7] Historie jazyka SQL. *Historie jazyka SQL* [online]. 2005 [cit. 2017-05-23]. Dostupné z: <http://www.rydval.cz/phprs/view.php?cislocianku=2005123125>
- [8] What are the difference between DDL, DML and DCL commands? *What are the difference between DDL, DML and DCL commands?* [online]. 2004 [cit. 2017-05-23]. Dostupné z: http://www.orafaq.com/faq/what_are_the_difference_between_ddl_dml_and_dcl_commands
- [9] Top 7 Features Coming to SQL Server 2016. *Top 7 Features Coming to SQL Server 2016* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <https://redmondmag.com/articles/2015/06/03/features-to-sql-server-2016.aspx>
- [10] Novinky SQL 2016. *Co dává Microsoft SQL Server 2016 vývojářům na hrani?* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.gopas.sk/News/Co-dava-Microsoft-SQL-Server-2016-vyvojarum-na-hrani-.aspx>
- [11] Lernikon. *Ukázka PolyBase: čtení dat z CSV souborů uložených na Azure Blob Storage* [online]. 2015 [cit. 2017-05-23]. Dostupné z:

- <http://www.lernikon.cz/blog/clanek/ukazka-polybase--cteni-dat-z-csv-souboru-ulozenych-na-azure-storage>
- [12] Lernikon. *Nové funkce pro práci s daty ve formátu JSON: ISJSON, JSON_VALUE a JSON_QUERY* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/nove-funkce-pro-praci-s-daty-ve-formatu-json---isjson--json-value-a-json-query>
- [13] Lernikon. *Stroj času pro Vaše dotazy - Temporal Tables* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/stroj-casu-pro-vase-dotazy---temporal-tables>
- [14] Lernikon. *Stretch Database - přesouváme historická data do oblak* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/stretch-database---presouvame-historicka-data-do-oblak>
- [15] Lernikon. *Zabezpečení na úrovni řádku v SQL Serveru 2016 (CTP 2.1)* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/zabezpeceni-na-urovni-radku-v-sql-serveru-2016--ctp-2-1->
- [16] Lernikon. *Maskování dynamických dat v SQL Serveru 2016 (CTP 2.1)* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/maskovani-dynamickych-dat-v-sql-serveru-2016--ctp-2-1->
- [17] Lernikon. *Live Query Statistics v SQL Serveru 2016* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <http://www.lernikon.cz/blog/clanek/live-query-statistics-v-sql-serveru-2016>
- [18] What is SQL Server? - Definition from Techopedia. *SQL Server* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.techopedia.com/definition/1243/sql-server>
- [19] SQL Server 2016 - Editions | Microsoft. *SQL Server 2016* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/en-us/sql-server/sql-server-2016-editions>
- [20] Edice SQL Server 2016 SP1 Enterprise | Microsoft. *Edice SQL Server 2016 SP1 Enterprise* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-enterprise>

- [21] Edice SQL Server 2016 SP1 Standard | Microsoft. *Edice SQL Server 2016 SP1 Standard* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-standard>
- [22] Edice SQL Server 2016 SP1 Express | Microsoft. *Edice SQL Server 2016 SP1 Express* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-express>
- [23] Edice SQL Server 2016 SP1 Developer | Microsoft. *Edice SQL Server 2016 SP1 Developer* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-developers>
- [24] SQL Server 2016 - Comparison | Microsoft. *Compare Microsoft SQL Server versions* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.microsoft.com/en-us/sql-server/sql-server-2016-comparison>
- [25] KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|---------|---------------------------------------|
| Tj, | To je |
| USA | United States of America |
| COBOL | Common business-oriented language |
| SQL | Structured Query Language |
| 1NF | First normal form |
| 2NF | Second normal form |
| 3NF | Third normal form |
| SEQUE | Structured English Query Language |
| ISO | International Standards Organization |
| XML | Extensible Markup Language |
| DDL | Data Definition Language |
| DML | Data Manipulation Language |
| ANSI | American National Standards Institute |
| DCL | Data Control Language |
| JSON | JavaScript Object Notation |
| MPP | Massive Parallel Processing |
| OLTP | Online transaction processing |
| SSD | Solid-state drive |
| CODASYL | Conference on Data Systems Languages |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obrázok 1 – Hierarchický model..... | 13 |
| Obrázok 2 – Sieťový model..... | 14 |
| Obrázok 3- Relačný model | 14 |
| Obrázok 4 Voľba jazyka a detaily inštalácie | 32 |
| Obrázok 5 Voľba 32 a 64 bitovej verzie..... | 32 |
| Obrázok 6 Uloženie inštalačného súboru | 33 |
| Obrázok 7 Výber inštalácie Microsoft SQL Server 2014..... | 34 |
| Obrázok 8 Odsúhlasenie licenčných podmienok..... | 35 |
| Obrázok 9 Povolenie automatických aktualizácií..... | 36 |
| Obrázok 10 Výber doplnkov k inštalácii | 37 |
| Obrázok 11 Konfigurácia inštalácie. | 38 |
| Obrázok 12 Konfigurácia servera | 39 |
| Obrázok 13 Autorizácia pomocou hesla | 40 |
| Obrázok 14 Posledný krok inštalácie..... | 41 |
| Obrázok 15 Rozhranie Microsoft SQL 2014 Express | 42 |
| Obrázok 16 Grafické znázornenie funkcie INNER JOIN | 46 |
| Obrázok 17 Grafické znázornenie funkcie LEFT OUTER JOIN..... | 47 |
| Obrázok 18 Grafické znázornenie funkcie RIGHT OUTER JOIN | 47 |
| Obrázok 19 Grafické znázornenie funkcie FULL OUTER JOIN | 48 |
| Obrázok 20 Grafické znázornenie funkcie CROSS JOIN | 48 |
| Obrázok 21 Grafické znázornenie funkcie UNION | 49 |
| Obrázok 22 Grafické znázornenie funkcie INTERSECT..... | 49 |
| Obrázok 23 Grafické znázornenie funkcie EXCEPT | 50 |

SEZNAM TABULEK

| | |
|---|----|
| Tabuľka 1 Funkcie zamerané na výkon | 27 |
| Tabuľka 2 Funkcie zamerané na dostupnosť služieb..... | 28 |
| Tabuľka 3 Funkcie zamerané na zabezpečenie databázy | 28 |
| Tabuľka 4 Funkcie zamerané na integritu s cloudom..... | 29 |
| Tabuľka 5 Funkcie zamerané na správu databázy | 29 |
| Tabuľka 6 Funkcie zamerané na zlepšenie analýzy..... | 30 |

SEZNAM PŘÍLOH

PRÍLOHA P I: Súbor výukových materiálov v anglickom jazyku na CD

PŘÍLOHA P I: NÁZEV PŘÍLOHY

Priložené CD obsahuje zložku Presentations a Exercises, ktoré slúžia ako výchovné materiály pre predmet vedený v angličtine so zameraním na databázové systémy.