

Koncepce jazyka SQL



Tato kapitola obsahuje základní informace o jazyku SQL, na kterých budeme stavět ve zbývající části knihy. Jazyk SQL se prosadil jako univerzální jazyk relačních databází a podporují jej téměř všechny systémy řízení báze dat (SRBD), které se v současnosti používají. Není pochyb, že značné rozšíření tohoto jazyka je důsledkem času a úsilí věnovaného vývoji jeho funkcí a standardů. Díky tomu se jazyk SQL vyznačuje vysokou přenositelností mezi různými produkty relační systémy řízení báze dat (RSRBD).

Seznamování s jazykem SQL vám půjde mnohem lépe, pokud budete při čtení aktivně používat SQL a relační databázi a zkoušet příklady uvedené v této knize. Ještě více se naučíte, když budete experimentovat s vlastními úpravami příkladů. Databázi video obchodu, která je základem příkladů v celé knize, můžete implementovat v libovolné relační databázi. Pokyny ke stažení příkazů SQL, které definují tabulky a načítají do nich ukázková data, jsou uvedeny na konci kapitoly 1.

Co je SQL

SQL (*Structured Query Language*) je standardní jazyk, který se používá při komunikaci s relačními databázemi. Název můžete vyslovovat buď jako jednotlivá písmena S-Q-L, nebo jako slovo „sequel“ [síkvl]. *Dotaz* (query) je požadavek, který se odesílá databázi. Na základě tohoto požadavku databáze žadateli zpětně poskytne určitou odpověď. SQL je nejrozšířenější jazyk, který umožňuje tvořit databázové dotazy. SQL se řadí mezi *neprocedurální* neboli *deklarativní* jazyky. To znamená, že počítači sdělíte, jaké výsledky požadujete. Přitom nemusíte definovat, jak tyto výsledky získat. Chcete-li například zjistit průměr sloupce s čísly, stačí o tento údaj požádat pomocí funkce AVG. Není nutné počítat, kolik čísel sloupec obsahuje, a potom dělit jejich součet počtem hodnot – o tyto operace se automaticky postará modul jazyka SQL v SRBD. Podrobný popis funkcí jazyka SQL naleznete v kapitole 4.

Je důležité pochopit, že SQL *nepatří* mezi procedurální jazyky, jako jsou C, Pascal, Basic, FORTRAN, COBOL či Ada. *Procedurální* jazyk pracuje s posloupnostmi příkazů, které se spouštějí v určeném pořadí. Součástí procedurálních jazyků jsou také příkazy, které mohou změnit pořadí provádění příkazů větvením na jinou část procedury, nebo cyklicky procházet určitou sadu příkazů v proceduře. Mnoho dodavatelů RSRBD nabízí procedurální rozšíření základního jazyka SQL – příkladem je PL/SQL (Procedural Language/SQL) společnosti Oracle nebo Transact-SQL v databázích Microsoft. Uvědomte si však, že tato rozšíření SQL lze označit za nové jazyky. Samotný jazyk SQL, který tvoří jejich podmnožinu, zůstává neprocedurální. Jazyk SQL se také odlišuje od objektově orientovaných programovacích jazyků, jako jsou Java a C++. Jednoduše řečeno, jazyk SQL slouží ke správě a údržbě relačních databází. Tento jazyk není vhodný k obecnému programování aplikací, například systémů na příjem objednávek a zpracování mezd.

SQL se často používá v kombinaci s výše uvedenými procedurálními a objektově orientovanými jazyky, kdy zajišťuje ukládání a načítání dat. Další úkoly programu, jako je prezentace dat na webové stránce či reakce na uživatelský vstup z klávesnice a myši, se přitom zajišťují pomocí příkazů v příslušném obecnějším programovacím jazyku. V případě požadavku na interakci s databází vytvoří program pomocí příkazů procedurálního jazyka příkaz jazyka SQL, odešle jej ke zpracování do RSŘBD, přijme výsledky z RSŘBD a vhodným způsobem je zpracuje.

Připojení k databázi

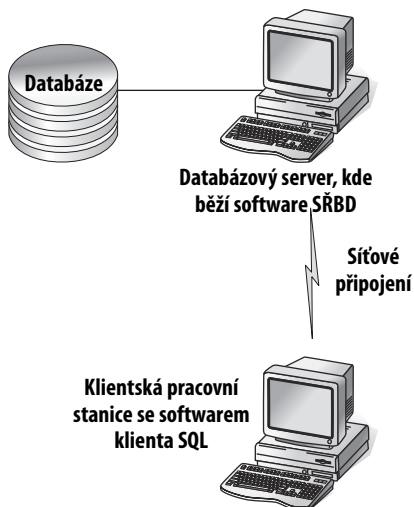
Používáte-li jazyk SQL v osobním počítači, kde je nainstalována samostatná kopie SŘBD (např. Microsoft Access nebo Oracle Personal Edition), jsou všechny databázové komponenty spuštěny v jednom počítačovém systému. Toto uspořádání se však nehodí pro databáze, které je nutné sdílet mezi více uživateli. Mnohem častěji se proto databáze implementují v uspořádání klient/server, jak je znázorněno na obrázku 2.1.

V uspořádání klient/server:

- Software SŘBD funguje na *serveru*, což je sdílený počítačový systém. Pro účely této definice lze počítačový systém typu mainframe považovat za velký server.
- Soubory, které fyzicky tvoří databázi, jsou uloženy na discích připojených k databázovému serveru.
- Uživatelé přistupují k databázi z pracovních stanic, které se označují jako *klienti*. Klient musí mít síťové připojení k databázi. Může se jednat o soukromou síť zapojenou v domácnosti či kanceláři, nebo o veřejnou síť, jako je Internet.
- V klientské pracovní stanici je spuštěn software poskytovaný dodavatelem SŘBD. Tento software umožňuje uživatelům zadávat příkazy SQL, odesílat je do SŘBD ke zpracování a zobrazovat výsledky vrácené z SŘBD. Tento software se obecně nazývá *klient SQL*.

Nic vám samozřejmě nebrání v tom, abyste si software klienta SQL nainstalovali do stejného počítače jako vlastní SŘBD. Vývojáři aplikací, kteří pracují s SŘBD typu MySQL, Microsoft SQL Server a Oracle, to v praxi dělají často, protože je pohodlné mít celé prostředí v jediném počítačovém systému (například v notebooku). Ve fázi, kdy je nutné zajistit přístup více uživatelů, je však praktičtější a efektivnější umístit jedinou kopii SŘBD na sdílený server a do pracovních stanic jednotlivých uživatelů instalovat pouze klient SQL.

Klienti SQL se podle uživatelského rozhraní v klientské pracovní stanici dělí na tři základní typy: na klienty příkazového řádku, grafické a webové klienty. Rozhraní *příkazového řádku* je založeno výhradně na textovém vstupu a výstupu. Příkazy se zadávají z klávesnice a odpovědi na příkazy se zobrazují jako textové zprávy. Hlavní výhodou rozhraní příkazového řádku spočívá v tom, že mohou fungovat téměř v libovolném operačním systému. Jako příklad klienta SQL pro příkazový řádek je na obrázku 2.2 znázorněn klient Oracle SQL*Plus (jeden z klientů SQL společnosti Oracle), který je spuštěn v okně konzoly systému Microsoft Windows. *Grafické uživatelské rozhraní (GUI)* lze spustit v systému založeném na oknech, jako je X Window System, Mac OS nebo Microsoft Windows. Data a možnosti příkazů se zobrazují pomocí grafických prvků typu ikon, tlačítek a dialogových oken. Na obrázku 2.3 vidíte klienta Oracle SQL*Plus, který je spuštěn jako grafická aplikace v systému Microsoft Windows. Činnost *webového* rozhraní zajišťuje databázový server a k interakci s uživatelem databáze slouží webový prohlížeč v klientské pracovní stanici. Webový klient SQL technicky vzato vůbec nepatří mezi klientské aplikace, protože v klientské pracovní stanici není spuštěn žádný software od dodavatele SŘBD. Téměř vždy se však používají kompo-



Obrázek 2.1: Připojení klienta SQL k databázi

nenty vyvíjené dodavatelem SRBD, které se na pozadí stahují do webového prohlížeče a umožňují grafickou prezentaci webového formuláře pro zadávání příkazů SQL a zobrazování výsledků. Na obrázku 2.4 je zobrazen klient Oracle iSQL*Plus spuštěný v prohlížeči Mozilla Firefox.

```

C:\Documents and Settings\Andy>sqlplus

Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Andy>sqlplus

SQL*Plus: Release 10.1.0.2.0 - Production on Wed Jan 12 22:24:48 2005
Copyright (c) 1982, 2004, Oracle. All rights reserved.

Enter user-name: sql_demystified
Enter password:

Connected to:
Personal Oracle Database 10g Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> SELECT MPAA_RATING_CODE FROM MPAA_RATING;

MPAA_
-----
G
NC-17
NR
PG
PG-13
R

6 rows selected.

SQL>

```

Obrázek 2.2: Klient Oracle SQL*Plus spuštěný v okně konzoly systému Microsoft Windows

Obrázek 2.3: Klient Oracle SQL*Plus spuštěný jako grafická aplikace systému Microsoft Windows

Následující tabulka obsahuje seznam klientského softwaru SQL od různých dodavatelů SRBD. V této knize nemáme dostatek místa, abychom mohli uvést podrobnosti o všech rozšířených klientech SQL. Chcete-li tedy získat informace o instalaci a použití klientů SQL, které jsou k dispozici pro váš SRBD, prostudujte si dokumentaci od dodavatele svého systému.

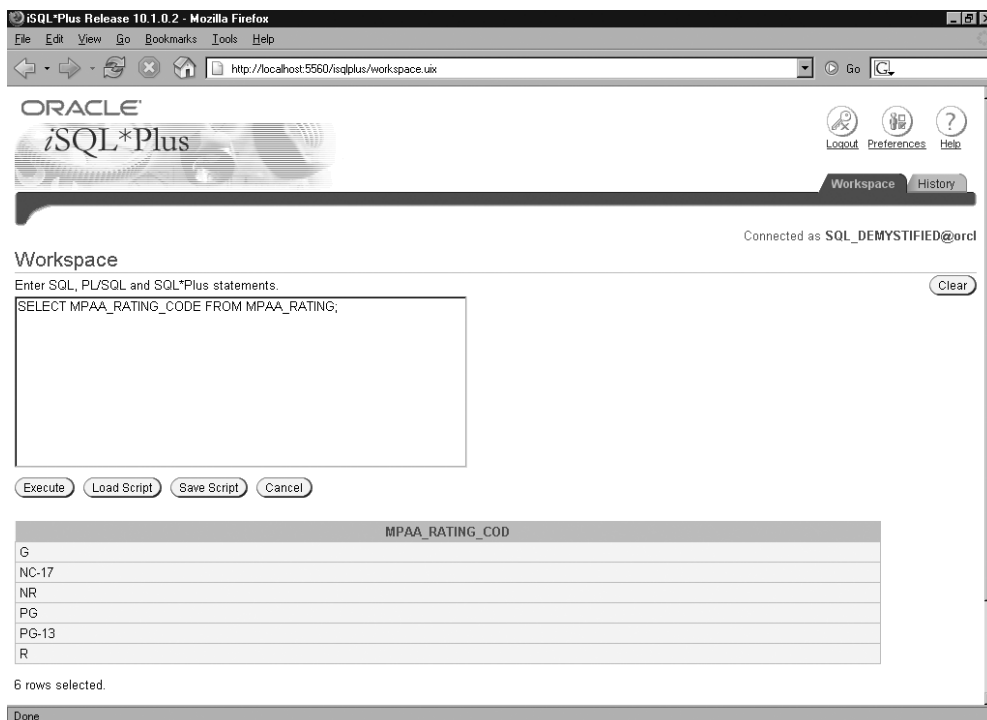
Dodavatel	SRBD	Klient SQL	Popis
Microsoft	Access	Žádné	Microsoft Access je databáze pro osobní použití, Klient SQL je proto integrální součástí SRBD a všechny komponenty se spouští místně v pracovní stanici uživatele.
Microsoft	SQL Server	iSQL	Klient SQL se spouští jako aplikace příkazového řádku v okně konzoly systému Microsoft Windows.
Microsoft	SQL Server	Query Analyzer (ve verzi 2005 je to program SQL Server Management Studio)	Klient SQL se spouští jako grafická aplikace systému Microsoft Windows.
MySQL	MySQL	MySQL	Klient SQL se spouští jako aplikace příkazového řádku v mnoha různých operačních systémech, mj. Microsoft Windows, Linux, Mac OS X a různých implementacích systému Unix.
Oracle	Oracle	iSQL*Plus	Webový klient SQL, který je podporován ve verzi Oracle 9i a novějších verzích.
Oracle	Oracle	SQL*Plus	Klient SQL lze spustit jako grafickou aplikaci systému Microsoft Windows, nebo jako aplikaci příkazového řádku v mnoha různých operačních systémech, mj. Microsoft Windows, Linux, Mac OS X a různých implementacích systému Unix.

Dodavatel	SŘBD	Klient SQL	Popis
Oracle	Oracle	SQL Worksheet	Klient SQL napsaný v jazyku Java. Je k dispozici v databázových systémech Oracle 8i a 9i, ale v systému Oracle 10g jej nahrazuje klient iSQL*Plus.
Sybase	Sybase	iSQL	Klient SQL se spouští jako aplikace příkazového řádku v okně konzoly systému Microsoft Windows. Podoba s databází Microsoft SQL Server není náhodná – nejstarší verze databáze Microsoft SQL Server byly založeny na SŘBD Sybase.

Stručná historie jazyka SQL

Koncem 70. let vyvinula skupina výzkumníků ve společnosti IBM experimentální relační databázi s názvem System/R, která byla založena na práci Dr. E. F. Cotta. Součástí databáze System/R byl jazyk označovaný SEQUEL (Structured English Query Language), který umožňoval manipulaci s daty a jejich načítání. Později se zjistilo, že slovo „SEQUEL“ je ochranná známka britské společnosti Hawker-Siddeley Aircraft Company. Akronym „SEQUEL“ byl proto zkrácen na „SQL“.

Společnost IBM sice přišla s první implementací SQL, ale na trhu ji předstihly dva jiné produkty, které zahrnovaly dotazovací jazyky s odlišnými názvy. Prvenství mezi komerčními relačními databázemi si tedy vydobily Oracle společnosti Relational Software a INGRES od firmy Relational Technology. Společnost IBM v roce 1982 vydala databázi SQL/DS, jejíž dotazovací jazyk byl přejmenován na SQL (Structured Query Language). V 80. letech sice bylo velmi módní hovořit o *strukturovaném programování*, ale slovo „structured“ v názvu SQL se strukturovaným programo-



Obrázek 2.4: Klient Oracle iSQL*Plus spuštěný v prohlížeči Mozilla Firefox

váním nijak nesouvisí, protože SQL není procedurální programovací jazyk. Je však docela dobře možné, že marketingový rozruch kolem strukturovaného programování přispěl k tomu, že se název „SQL“ prosadil v konkurenci s názvy, které používali jiní tehdejší dodavatelé dotazovacích jazyků.

Standardizační komise jazyka SQL vznikly při institutu ANSI (American National Standards Institute) roku 1986 a organizaci ISO (International Organization for Standardization) v roce 1987. Komise obou organizací našťásti spolupracovaly na vývoji jednotného a obecně platného standardu SQL. O dva roky později byla publikována první specifikace standardu, známá pod označením SQL-89. O tři roky později byl standard rozšířen ve verzi SQL-92, která měla přibližně 600 stránek. Třetí generace standardu se nazývá SQL-99 nebo také SQL3. Většina produktů RSŘBD je založena na standardu SQL-92 (který se nyní označuje jako SQL2). Standard SQL3 obsahuje mnoho objektových funkcí, aby mohl jazyk SQL pracovat s objektově-relačními databázemi. Kromě toho zahrnuje jazyková rozšíření, díky nimž je jazyk SQL výpočetně kompletní (nově jsou k dispozici smyčky, větvení a konstrukty typu CASE). Poslední generace s názvem SQL:2003 přináší funkce související s jazykem XML a další rozšíření. Pouze několik málo dodavatelů implementovalo do svých produktů významné části standardů SQL3 a SQL:2003. Dodavatelé SŘBD sice zaměstnávají týmy expertů, kteří se věnují souladu se standardy, ale většina programátorů v SQL tyto nové standardy příliš nezná. Je to způsobeno zejména tím, že standardy nejsou volně dostupné. Standard SQL:2003 si můžete zakoupit u organizace ISO (www.iso.org) nebo institutu ANSI (webstore.ansi.org). Pokud máte omezený rozpočet, můžete si stáhnout pokročilý koncept standardu z webu společnosti Whitemarsh Information Systems Corporation (www.wiscorp.com/SQLStandards.html).

Význam standardů spočívá v tom, že umožňují *přenositelnost*, tzn. snadnější spouštění softwaru na jiných platformách. Přenositelnost kódu v jazyku SQL mezi produkty RSŘBD od různých dodavatelů byla poměrně špatná, dokud dodavatelé nezačali dodržovat publikované standardy. Téměř všichni dodavatelé však přidali do jazyka SQL své vlastní prvky. Za prvé chtěli své produkty odlišit, a za druhé byli pod tlakem trhu, aby implementovali funkce dříve, než vznikly příslušné standardy. Jako příklad lze uvést podporu datových typů DATE a TIMESTAMP. Data mají ve zpracování obchodních údajů mimořádný význam. První produkty RSŘBD však vyvíjeli počítačová experti a akademici, nikoli specialisté na podnikové výpočetní systémy. Proto potřebu těchto datových typů nepředpokládali. První verze jazyka SQL proto nezahrnovaly žádnou speciální podporu dat. Když se začaly objevovat první komerční produkty, dodavatelé reagovali na tlak svých největších zákazníků a urychleně podporu dat doplňovali. Každý z dodavatelů přitom bohužel postupoval jinak. Kdykoli přenášíte příkazy SQL do produktu jiného dodavatele, musíte si být vědomi rozdílů mezi dialekty SQL. Jazyk SQL je dostatečně kompatibilní a kód lze mezi produkty různých dodavatelů snadno přenášet, ale kompletní databázové systémy při přenosu zpravidla vyžadují jisté úpravy.

Syntaktické konvence jazyka SQL

V této části se seznámíte s obecnými syntaktickými konvencemi, pomocí nichž se tvoří příkazy SQL. Nezapomeňte však, že existuje mnoho rozšíření a odchylek v závislosti na jednotlivých dodavatelích. Pro jednoduchost budeme termínem *implementace* označovat jednotlivé verze jazyka SQL od příslušného dodavatele (tj. Oracle 9i, Oracle 10g, Microsoft SQL Server 7, Microsoft SQL Server 2000 i Microsoft SQL Server 2005 zahrnují různé *implementace* jazyka SQL).

Syntaktické konvence jazyka SQL si přiblížíme pomocí jednoduchého příkladu. Tento příkaz vypíše hodnoty atributů ID filmu a Název filmu pro každý film ve video obchodě, který má hodnocení PG:

```
SELECT ID_FILMU, NAZEV_FILMU
FROM FILMY
WHERE KOD_HODNOCENI_MPAA = 'PG';
```

Uplatňují se tyto základní konvence:

- Každý příkaz začíná klíčovými slovy (zpravidla jedním). Téměř vždy se jedná o anglické dějové sloveso. Výše uvedený příkaz začíná klíčovým slovem SELECT, které je podrobně popsáno v kapitole 4.
- Každý příkaz končí oddělovačem, což je většinou středník (;). Některé implementace dovolují nastavit jiný znak oddělovače. Určité implementace (např. Oracle) nespustí příkaz SQL bez koncového oddělovače, zatímco v jiných implementacích není koncový oddělovač povinný.
- Struktura příkazů se podobá anglickým větám. Jazykové prvky se přitom oddělují jednou nebo více mezerami. *Jazykový prvek* odpovídá slovu v anglické větě. Jedná se o libovolnou povolenou součást příkazu: klíčové slovo (SELECT, FROM, WHERE), název databázového objektu (FILMY, ID_FILMU, NAZEV_FILMU), operátor (=) nebo konstantu (,PG').
- Příkazy se zapisují volným způsobem, tzn. neexistují přísná pravidla ohledně umístění jazykových prvků na řádku nebo rozdělení příkazu na více řádků. Obvykle však není vhodné přerušovat jazykový prvek koncem řádku. Následující příkaz se logicky shoduje s příkazem, který jsme uvedli na začátku tohoto tématu, avšak je méně srozumitelný:

```
SELECT ID_FILMU,NAZEV_FILMU FROM FILMY WHERE
KOD_HODNOCENI_MPAA='PG';
```

- Příkazy tvoří řadu klauzulí, které obvykle musí následovat v určitém pořadí (mnohé klauzule jsou však volitelné). Uvedený příklad zahrnuje tři klauzule. Každá z nich začíná klíčovým slovem (SELECT, FROM, WHERE).
- Prvky jazyka SQL lze zapisovat velkými písmeny, malými písmeny nebo pomocí jejich kombinace. Ve většině implementací se však podle standardů ANSI/ISO všechna malá písmena před zpracováním automaticky převádějí na velká. To neznamená, že byste nemohli zadávat malými písmeny data. Požadavek na zápis velkými písmeny se týká příkazů a názvů databázových objektů (tabulek, sloupců atd.). Jako výjimky je nutné uvést produkty Microsoft SQL Server a Sybase. Jejich databáze mohou fungovat v „režimu rozlišování velkých a malých písmen“, kdy se názvy objektů zadané s různou velikostí písmen považují za odlišné. V databázi MySQL je rozlišování velikosti písmen v názvech objektů závislé na tom, zda se touto vlastností vyznačuje hostitelský operační systém.
- Položky seznamu se oddělují čárkami. Náš příklad obsahuje seznam dvou názvů sloupců oddělených čárkami (ID_FILMU, NAZEV_FILMU). Všechny mezery za čárkami jsou nepovinné – mezery lze úplně vynechat, nebo jich můžete zadat libovolný počet.
- Znakové řetězce zahrnuté v příkazech SQL je nutné uzavřít do apostrofů (některé implementace jazyka SQL také dovolují použít dvojité uvozovky). Číselné konstanty se do apostrofů nikdy neuzavírají. Pokud chcete uvést symbol apostrofu jako součást znakového řetězce, stačí zadat dva apostrofy po sobě. Chcete-li například v databázi najít film Sophie's Choice, zadejte klauzuli WHERE takto:

```
WHERE NAZEV_FILMU = 'Sophie''s Choice'
```

- Názvy databázových objektů mohou obsahovat výhradně písmena, číslice a znak podtržítka. Podtržítka se obvykle používají jako oddělovače slov, aby byl název srozumitelnější. Jak

jsme již uvedli, některé implementace umožňují pracovat s názvy, které kombinují velká a malá písmena (např. DruheJmenoOsoby – tento styl se označuje jako „velbloudí notace“). Chcete-li však umožnit přenos kódu SQL do jiných implementací, není tento postup vhodný. Po přenosu totiž vzniknou názvy typu „DRUHEJMENOOSOBY“, které se špatně čtou.

- Každá implementace jazyka SQL obsahuje definovanou sadu vyhrazených slov. Tato slova mají v procesoru dotazů SQL v SRBD speciální význam, a proto je nelze použít v jiném kontextu, např. jako název databázového objektu. Toto omezení zajišťuje, aby nedošlo k nesprávné interpretaci příkazů SQL v SRBD. Asi tušíte, že seznam vyhrazených slov se značně liší v závislosti na implementaci jazyka SQL. Proto je rozumné prostudovat si dokumentaci k implementaci jazyka, kterou používáte.
- Jednořádkový komentář začíná dvěma pomlčkami za sebou (--). Dvě pomlčky se mohou nacházet na začátku řádku – pak slouží jako komentář celý řádek. Jsou-li na jiném místě řádku, považuje se za komentář zbytek řádku. Například:

```
-- Toto je jednořádkový komentář v jazyku SQL.
```

- Víceřádkový komentář začíná kombinací lomítka a hvězdičky (/*) a pokračuje, dokud procesor nenalezne opačnou kombinaci (*/). Dbejte na to, abyste komentáře správně ukončili. Pokud na ukončovací řetězec zapomenete, může SRBD považovat několik pečlivě napsaných řádků kódu za komentář. Uvedme si příklad víceřádkového komentáře:

```
/* Toto je víceřádkový komentář.  
   Pokračuje, dokud se neobjeví ukončovací  
   kombinace znaků. */
```

Kategorie příkazů jazyka SQL

Příkazy SQL se v závislosti na své funkci dělí do kategorií. Podle názoru některých odborníků jsou tyto kategorie buď samostatnými jazyky, nebo jejich částmi. Všechny kategorie jazyka SQL však mají stejnou základní syntaxi a pravidla. Proto je můžeme považovat za kategorie příkazů v rámci jediného jazyka. Jedná se o následující kategorie, které jsou dále popsány:

- jazyk DDL (Data Definition Language),
- jazyk DQL (Data Query Language),
- jazyk DML (Data Manipulation Language),
- jazyk DCL (Data Control Language),
- příkazy řízení transakcí.

Jazyk DDL (Data Definition Language)

Jazyk DDL zahrnuje příkazy SQL, které umožňují uživatelům databáze vytvářet databázové objekty (např. tabulky, pohledy a indexy) a upravovat jejich strukturu. Za součást jazyka DDL se považují příkazy SQL, které obsahují klíčová slova CREATE, ALTER a DROP. Je důležité si uvědomit, že příkazy jazyka DDL mají vliv na kontejnery, které uchovávají data v databázi, nikoli na vlastní data. Proto existují příkazy DDL pro vytvoření, odstranění a úpravy tabulek, ale žádný z těchto příkazů neumožňuje vytvářet či měnit řádky dat v těchto tabulkách. Popis příkazů jazyka DDL naleznete v kapitole 3.

Jazyk DQL (Data Query Language)

Jazyk DQL obsahuje příkazy SQL, které načítají data z databáze. Ačkoli se jedná o velmi důležitou součást jazyka SQL, jsou příkazy jazyka DQL založeny pouze na jediném klíčovém slově: SELECT. Jazykem DQL se budeme zabývat v kapitolách 4, 5 a 6. Někteří dodavatelé a autoři pracují s hierarchií příkazů SQL, kde jazyky DQL a DML patří do stejné skupiny.

Jazyk DML (Data Manipulation Language)

Součástí jazyka DML jsou příkazy SQL, které umožňují uživatelům přidávat data do databáze (v podobě řádků nebo tabulek), odebírat data z databáze a měnit stávající data. Do jazyka DML se zahrnují příkazy SQL s klíčovými slovy INSERT, UPDATE a DELETE. S jazykem DML se seznámíte v kapitole 7.

Jazyk DCL (Data Control Language)

Do jazyka DCL patří příkazy SQL, které správcům dovolují řídit přístup k datům v databázi a používat různá systémová oprávnění SRBD, jako je například funkce pro spuštění nebo vypnutí databáze. Jazyk DCL sdružuje příkazy jazyka SQL, kde se vyskytují klíčová slova GRANT a ALTER. Popisu jazyka DCL je věnována kapitola 8.

Příkazy řízení transakcí

Databázová transakce je sada příkazů, kterou databázový uživatel požaduje zpracovat jako nedělitelnou jednotku. To znamená, že transakce musí být kompletně úspěšná nebo neúspěšná. Příkazy řídicí databázové transakce přesně neodpovídají syntaxi příkazů jazyka SQL, ale mají značný vliv na chování těch příkazů SQL, které jsou součástí transakcí. Příkazy řízení transakcí se budeme zabývat v kapitole 9.

Test

Zvolte správné odpovědi na jednotlivé otázky. Na všechny otázky může existovat více správných odpovědí.

1. SQL
 - a. Lze hláskovat jako S-Q-L
 - b. Lze vyslovit jako slovo „sequel“ [síkvl]
 - c. Umožňuje zobrazovat webové stránky
 - d. Dovoluje komunikovat s libovolnou databází
 - e. Slouží ke komunikaci s relačními databázemi
2. SQL je
 - a. Objektově orientovaný jazyk
 - b. Procedurální jazyk
 - c. Neprocedurální jazyk
 - d. Deklarativní jazyk
 - e. Standardní jazyk

3. Mezi procedurální rozšíření jazyka SQL patří
 - a. Java
 - b. Oracle PL/SQL
 - c. C++
 - d. Microsoft Transact-SQL
 - e. FORTRAN
4. V uspořádání klient/server
 - a. Software SŘBD funguje na serveru
 - b. Software SŘBD funguje na klientovi
 - c. Klientský software SQL je spuštěn na klientovi
 - d. Klientský software SQL lze spustit na serveru
 - e. Databáze je uložena na discích připojených ke klientovi
5. Klient SQL pro příkazový řádek
 - a. Vyžaduje grafický systém s okny
 - b. Lze spustit na mnoha různých klientech
 - c. Vyžaduje na klientovi webový prohlížeč
 - d. Zobrazuje data a parametry příkazů pomocí grafických prvků
 - e. Zobrazuje odpovědi na příkazy formou textových zpráv
6. Grafický klient SQL
 - a. Vyžaduje grafický systém s okny
 - b. Lze spustit na mnoha různých klientech
 - c. Vyžaduje na klientovi webový prohlížeč
 - d. Zobrazuje data a parametry příkazů pomocí grafických prvků
 - e. Zobrazuje odpovědi na příkazy formou textových zpráv
7. Webový klient SQL
 - a. Vyžaduje grafický systém s okny
 - b. Lze spustit na mnoha různých klientech
 - c. Vyžaduje na klientovi webový prohlížeč
 - d. Zobrazuje data a parametry příkazů pomocí grafických prvků
 - e. Zobrazuje odpovědi na příkazy formou textových zpráv
8. Společnost Oracle nabízí klienty SQL
 - a. iSQL
 - b. Query Analyzer
 - c. iSQL*Plus
 - d. SQL*Plus
 - e. SQL Worksheet
9. Společnost Microsoft nabízí klienty SQL
 - a. iSQL
 - b. Query Analyzer
 - c. iSQL*Plus
 - d. SQL*Plus
 - e. SQL Worksheet

10. Jazyk SQL byl vyvinut
 - a. Ve společnosti IBM
 - b. V institutu ANSI
 - c. Roku 1982
 - d. V 70. letech
 - e. Na základě standardů ANSI
11. Ke standardům SQL patří
 - a. SQL-88
 - b. SQL-89
 - c. SQL-92
 - d. SQL-99
 - e. SQL:2003
12. Dodavatelská rozšíření jazyka SQL
 - a. Zlepšují přenositelnost jazyka SQL
 - b. Zhoršují přenositelnost jazyka SQL
 - c. Pomáhají odlišit produkty jednotlivých dodavatelů
 - d. Vycházela z požadavků trhu
 - e. Jsou mezi implementacemi různých dodavatelů kompatibilní
13. Příkazy jazyka SQL
 - a. Začínají klíčovým slovem příkazu
 - b. Končí klíčovým slovem příkazu
 - c. Začínají oddělovačem, jako je středník
 - d. Končí oddělovačem, jako je středník
 - e. Začínají levou závorkou
14. K prvkům jazyka SQL patří
 - a. Klíčová slova
 - b. Názvy databázových objektů
 - c. Operátory
 - d. Omezení
 - e. Konstanty
15. Jako oddělovače prvků jazyka SQL slouží
 - a. Čárky
 - b. Právě jedna mezera
 - c. Jedna nebo více mezer
 - d. Nové řádky
 - e. Podtržítka
16. Názvy databázových objektů mohou obsahovat
 - a. Závorky
 - b. Podtržítka
 - c. Čísla
 - d. Písmena
 - e. Čárky

17. Příkazy jazyka SQL lze rozdělit do následujících kategorií:
 - a. Jazyk DDL (Data Definition Language)
 - b. Jazyk DSL (Data Selection Language)
 - c. Jazyk DRL (Data Replication Language)
 - d. Jazyk DCL (Data Control Language)
 - e. Jazyk DML (Data Manipulation Language)
18. Do jazyka DDL (Data Definition Language) patří následující příkazy:
 - a. SELECT
 - b. INSERT
 - c. CREATE
 - d. ALTER
 - e. DELETE
19. Do jazyka DQL (Data Query Language) patří následující příkazy:
 - a. SELECT
 - b. INSERT
 - c. CREATE
 - d. ALTER
 - e. DELETE
20. Do jazyka DML (Data Manipulation Language) patří následující příkazy:
 - a. SELECT
 - b. INSERT
 - c. CREATE
 - d. ALTER
 - e. DELETE