


Vysoká škola báňská – Technická univerzita Ostrava
Fakulta strojní, Katedra automatizační techniky a řízení

4

Informační systémy

2008/2009




Radim Farana

1

Obsah

- Jazyk SQL,
 - datové typy,
 - klauzule SELECT,
 - WHERE,
 - a ORDER BY.
- Doporučená literatura:
Gruber, M. *Mistrovství v SQL. Svazek 1.*
Praha : SoftPress s.r.o., 2004. ISBN 80-86497-62-3




Informační systémy

2

Jazyk SQL

- Historie začíná 1974 uveřejněním prvních prací dr. Codda.
- Snaha o vytvoření přirozeného jazyka pro dotazování (Angličtina).
- První použitelná verze IBM (SEQUEL).
- Postupně se přidávaly další firmy (Oracle, SyBase).
- Postupně vznikaly standardy SQL-86, SQL-92 označovaná SQL-2.



Informační systémy

3

Základní pojmy

● Použití

- Jako jazyk pro manipulaci s daty v relačních databázových systémech
- Jako součást hostitelského jazyka
 - Oracle – procedurální jazyk PL/SQL
 - MS SQL Server – transakční T-SQL

● Rozdělení

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)



Jazyk SQL - DDL

● Data Definition Language

- Definuje struktury a objekty v databázi
- Tabulky, pohledy, indexy apod.
- Příkazy

CREATE TABLE	CREATE VIEW	CREATE DATABASE
DROP TABLE	DROP VIEW	DROP DATABASE
ALTER TABLE	ALTER VIEW	ALTER DATABASE



Jazyk SQL - DML

● Data Manipulation Language (DML)

- Umožňuje manipulaci s daty
- Vkládání
- Výběr
- Aktualizace

SELECT	UPDATE	INSERT	DELETE
--------	--------	--------	--------



Jazyk SQL - DCL

- Data Control Language
 - Příkazy pro řízení provozu a údržbu databáze
 - Příkazy pro přidělování práv uživatelům a skupinám

CREATE USER	DROP USER
-------------	-----------

- Transaction Control Language (TCL)
 - Příkazy pro práci s transakcemi

SET TRANSACTION	COMMIT	ROLLBACK
-----------------	--------	----------



Datové typy

Typ dat:	Popis:
SMALLINT	Celá čísla v délce až šesti číslic (včetně znaménka). Rozpětí povolených hodnot je od -32,768 do 32,767. Velikost: 2 Byte.
INTEGER	Celá čísla v délce až 11 číslic (včetně znaménka). Rozpětí povolených hodnot je interval od -2,147,483,648 až do 2,147,483,647. Velikost: 4 Byte.
DECIMAL(x,y)	Čísla s pevnou řádovou čárkou se znaménkem celkem s x číslicemi a y desetinnými místy, x je implicitně 38 a y se implicitně rovná 6. Velikost: proměnná.
NUMERIC(x,y)	Obdobně jako DECIMAL(x,y). Velikost: proměnná.
FLOAT(x)	Čísla v pohyblivé řádové čárce. Rozpětí čísel sahá od $2.22507385850720160 \cdot 10^{-308}$ až k $1.79769313486231560 \cdot 10^{308}$. Velikost: 8 Byte.
CHAR(n)	Znakové řetězce v délce n znaků. Povolené rozpětí pro n je interval od 1 do 32767. Bez určení n je implicitně rovno 1. Velikost: n znaků.
CHARACTER(n)	
VARCHAR(n)	Umožňuje používat řetězce proměnné délky. Dovoluje ukládat řetězce různé délky v jednotlivých řádcích. Stanovuje se horní maximálně možná délka řetězce. Jinak podobně jako u CHAR(n). Velikost: n znaků.
VARYING CHARACTER	
LONG VARCHAR	Řetězce libovolného počtu znaků. Maximální velikost je omezena maximálně dovolenou velikostí databázového souboru.
DATE	Datum ve formátu dle nastavení: yyyy/mm/dd. Velikost: 4 Byte.
TIMESTAMP	Stejně jako datum plus hodina, minuta, vteřina a zlomek vteřiny (na 6 míst): yyyy/mm/dd hh-mm-ss.ffff. Velikost: 8 Byte.
TIME	Časový okamžik dne. Ve formátu hodina/minuta/vteřina. Velikost: 4 Byte.
DOUBLE	Obdobně jako FLOAT. Velikost: 8 Byte.
REAL	Reálná čísla s vědeckotechnickým formátem. Rozpětí čísel sahá od $1.175494351 \cdot 10^{-38}$ až do $3.402823466 \cdot 10^{38}$. Velikost: 4 Byte.
BINARY(n)	Binární data celkové délky n Byte. Implicitně je size = 1 Byte. Max. délka je 32767 Byte. Velikost: n Byte.
LONG BINARY	Binární datový typ libovolné délky. Omezení jako u LONG VARCHAR.



Datové typy SQL Serveru

- binary
- Bigint
- bit
- Char
- datetime
- decimal
- Float
- image
- Int
- Money
- nchar
- Ntext
- nvarchar
- Numeric
- Real
- smalldatetime
- smallint
- smallmoney
- sql_variant
- sysname
- text
- timestamp
- tinyint
- tvarbinary
- varchar
- uniqueidentifier



Vytvoření databáze

- `CREATE {DATABASE | SCHEMA}`
`"<specif_souboru> "`
`[USER "username" [PASSWORD`
`"password"]`
`[PAGE_SIZE [=] int]`
`[LENGTH [=] int [PAGE[S]]]`
`[DEFAULT CHARACTER SET charset]`



Vytvoření tabulky

- `CREATE TABLE tabulka "<jm_souboru>"`
`(<def_sloupace> [, <def_sloupace> |`
`<t_omezení> ...)]`
- `CREATE TABLE osoba`
`(osc Integer,`
`oname CHAR(50),`
`oaddr CHAR(255));`



Změna tabulky

- `ALTER TABLE jméno`
`{ADD <sloupec_def> | ADD`
`<tab_omezení> | DROP sloupec`
`| DROP CONSTRAINT omezení};`
- `ALTER TABLE osoba`
`ADD ostreet CHAR(100),`
`DROP oaddr;`



Odstranění tabulky

- DROP TABLE jméno
- Někdy je potřeba nejprve odstranit definovaná omezení,
- U některých systémů také odstranit všechny záznamy.



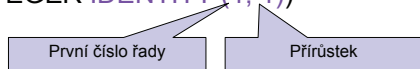
Deklarace omezení

- Vyloučení hodnoty NULL
- (osc INTEGER NOT NULL)
- Unikátnost hodnot
- (osc INTEGER NOT NULL UNIQUE)
- Primární klíč
- (osc INTEGER NOT NULL PRIMARY KEY)



Deklarace omezení

- Omezení IDENTITY (automatické číslo)
- (osc INTEGER IDENTITY (1, 1))



Kontrola sloupcových hodnot

- Omezení oboru hodnot
- (oplat DECIMAL CHECK (oplat>10000))
- Omezení výčtem hodnot
- (oplat DECIMAL CHECK (oplat IN (1000, 2000, 3000)))
- Omezení více sloupců
- (oplat DECIMAL CHECK (oplat>1000 OR ostreet='Moje ulice'))



Pojmenování omezení

- CONSTRAINT jméno CHECK (omezení)
- CONSTRAINT MinMzda CHECK (oplat>10000)
- Je možné je dodatečně odstranit
- ALTER TABLE DROP CONSTRAINT MinMzda



Přiřazení implicitní hodnoty

- (oplat DECIMAL DEFAULT=10000))
- Není-li implicitní hodnota určena, pak nabývá nevyplněná položka hodnoty NULL.



Referenční integrita

- Definice cizího klíče
- FOREIGN KEY (oplat) TabPlatOnly
REFERENCES TabulkaPlatu(tpsazba)
-
- Zkrácená definice
- (oplat DECIMAL REFERENCES
TabulkaPlatu(tpsazba))



Získávání dat z databáze

- SELECT
 - Nejpoužívanější příkaz
- Syntaxe
 - Vyber vše z tabulky Zakaznik
 - SELECT * FROM Zakaznik;
 - SELECT *select_list*
[INTO *new_table*]
FROM *table_source*
[WHERE *search_condition*]
[GROUP BY *group_by_expression*]
[HAVING *search_condition*]
[ORDER BY *order_expression* [ASC | DESC]];



Získávání dat z databáze - sloupce

- Výběr sloupců z tabulky
 - SELECT Jmeno, Prijmeni FROM Zakaznik;
 - Výpis sloupců pod jiným jménem (aliasy)
 - SELECT Jmeno As Jméno FROM Zakaznik;
 - Přidání konstantního sloupce
 - SELECT 'Plat:', Plat, 'Prijmení:', Prijmeni FROM Zakaznik;
- Plat: 100 Prijmeni: Novak
Plat: 300 Prijmeni: Novotny



Získávání dat z databáze - sloupce

- Výpočty s hodnotami sloupců
 - `SELECT Plat, Plat+100 As Premie FROM ZAMESTNANEC;`
 - Spojování hodnot sloupců
 - `SELECT Jmeno + ' ' + Prijmeni AS CeleJmeno FROM ZAKAZNIK;`
- CeleJmeno
Ales Hala
Petr Novak



Získávání dat z databáze - sloupce

- Ne vždy se hodí údaje v tabulce pro prezentaci
- Zobrazit hodnoty v intervalech (Malý dluh, ...)
- Jindy přiřadit jinou hodnotu (AHI – Praha západ)

```
SELECT NazevAuta,  
CASE spz  WHEN 'AHI' THEN 'Praha zapad'  
          WHEN 'OPA' THEN 'OPAVA'  
          ...  
          ELSE 'Neznama SPZ'
```

```
END  
FROM EvidenceAut;
```



Získávání dat z databáze - sloupce

```
SELECT Jmeno, 'Kategorie dluznika' =  
CASE Dluh  WHEN Dluh <= 1000 THEN 'Maly dluh'  
          WHEN Dluh > 1000 AND Dluh < 5000 THEN  
          'Velky dluh'  
          ...  
          ELSE 'Externi dluh'
```

```
END  
FROM Zakaznik;
```



Získávání dat z databáze - řádky

- Vymezení výpisu duplicitních řádků
 - DISTINCT – musí následovat hned za SELECT
 - SELECT DISTINCT Navez, Cena FROM VYROBEK;
- Výpis omezeného počtu řádků
 - TOP – 100 největších dlužníků
 - SELECT TOP 100 * FROM Zakaznik ORDER BY Dluh DESC;



Výpis konkrétních řádků

- Pomocí klauzule WHERE
 - SELECT * FROM Zakaznik WHERE Dluh>100;
 - Můžeme vypsát i konkrétní sloupce
 - Do podmínky můžeme zahrnout i sloupec, který nebude zahrnut do výpisu
- Porovnávací operátory
 - =, <, >, <=, >=, BETWEEN, NOT BETWEEN
- Logické operátory
 - AND, OR, NOT
- Pro řetězce využíváme operátor LIKE
 - WHERE (Jmeno LIKE 'Pet%')



Příklad podmínek v klauzuli WHERE

- Porovnávání číselných hodnot
 - WHERE (Vek > 30)
 - WHERE (Vek < 30)
 - WHERE (Vek >= 30)
 - WHERE (Vek <= 30)
 - WHERE (Vek <> 30)
 - WHERE (Vek BETWEEN 30 AND 40)
 - WHERE (Vek NOT BETWEEN 30 AND 40)



Příklad podmínek v klauzuli WHERE

- Porovnávání řetězcových hodnot
 - WHERE (Jmeno = 'Jana')
 - WHERE (Jmeno <> 'Jana')
 - WHERE (Jmeno LIKE 'Ja%') – začíná
 - WHERE (NOT Jmeno LIKE 'Ja%') – nezačíná
 - WHERE (Jmeno LIKE '%na') – končí
 - WHERE (NOT Jmeno LIKE '%na') – nekončí
 - WHERE (Jmeno LIKE '%an%') – obsahuje
 - WHERE (NOT Jmeno LIKE '%an%') – neobsahuje



Zjištění hodnoty Null

- Operace s hodnotou Null vrací vždy Null
 - WHERE Jmeno = NULL
- Je nutné použít operátor IS NULL
 - WHERE Jmeno IS NULL
 - WHERE Jmeno IS NOT NULL



Řazení hodnot

- Databázový stroj nám vrátí neuspořádaná data
- Nemusí vrátit stejně uspořádané množiny
- Vzestupné (Ascending) přidáním ASC
- Sestupné (Descending) přidáním DESC
- Přidáním klauzule ORDER BY

```
SELECT Vek, Plat FROM Tab ORDER BY Plat;  
SELECT Vek, Plat FROM Tab WHERE Vek > '40'  
ORDER BY Plat DESC;
```