

Malé veľké databázy / 5.časť

V úvode si objasníme problémy, ktoré vznikli pri domácej úlohe z minulej časti. Pozrime sa bližšie na 6.příklad z úlohy: Čo sa stane, keď po zmazení niektorého záznamu znova vložíme jeden nový záznam?

Odpoveď znie: Nový záznam sa nevloží na koniec ostatných záznamov, ale server nájde prvé voľné miesto po zmazanom zázname a využije ho. Ak už neexistuje voľné miesto po zmazaných záznamoch, pridá nový vkladateľ záznam na koniec všetkých záznamov. Toto je veľmi dôležité uvedomiť si pri operáciách mazania a vkladania, inak by sme mohli dostávať nežiadúce výsledky.

8.příklad je ešte nebezpečnejší: Ak zadáme : **delete from meno_tabuľky** bez podmienky, zmaže sa **celá tabuľka!** A ako už vieme, príkaz *delete* je deštruktívny a prideme o všetky dáta v príslušnej tabuľke.

Takže pozor na presné zadávanie príkazov!

Toľko k domácej úlohe.

Teraz si spustíme klient-monitor *mysql* s parametrom *kniznica*, aby sme mohli operovať s našimi tabuľkami.

Príkaz SELECT

SQL, a teda aj MySQL ponúka veľa spôsobov, ktoré slúžia na manipuláciu s dátami v databáze. Zvyšujú tak komfort obsluhy a kvalitu práce. Jedným z najznámejších, najužitočnejších a najpoužívanejších príkazov jazyka SQL je istotne príkaz **SELECT** (výber). Pomocou tohto príkazu môžeme vytiahnuť z databáze každý dôležitý údaj a vyriešiť tak veľa situácií v získaní tej správnej informácie.

V závere minulej časti sme si ukázali najzákladnejší tvar príkazu SELECT na prezeranie dát v našich tabuľkách KNIHA a ZANER v databázi KNIZNICA:

```
mysql> select * from zaner;
```

Výsledkom tohto príkazu je výpis č.5-1:

```
mysql> select * from zaner;
+-----+-----+
| cis_odd | tematika |
+-----+-----+
| 1      | poezia   |
| 2      | roman    |
| 3      | krimi    |
| 4      | detska lit. |
| 5      | cestopis |
| 6      | lit. faktu |
| 7      | odborna lit. |
+-----+-----+
```

Príkaz SELECT toho však dokáže podstatne viac. Popis všetkých funkcií by obsahol desiatky strán, preto sa zameriame iba na tie najpoužívanejšie. (Všetky možnosti príkazu SELECT sa nachádzajú v príslušnej dokumentácii k danému SQL serveru.)

Základný popis príkazu SELECT v MySQL je nasledovný:

```
SELECT {zoznam výstupných položiek } FROM {zoznam tabuliek}
//WHERE {podmienky}//
//GROUP BY{zoznam položiek}//
//HAVING {skupinová podmienka}//
//ORDER BY {meno položky ako podmienka zotriedenia}//
```

Zoznam výstupných položiek môže byť množina názvov jednotlivých stĺpcov tabuľky alebo aj tzv. **agregačné funkcie**. Čo sú agregačné funkcie si ukážeme nižšie.

Zoznam tabuliek sú názvy tabuliek v danej databázi.

Parametre uvedené v lomítkach nie sú povinné.

Skúsme si aplikovať tieto podmienky prakticky na našej databáze **KNIZNICA**.

Vypísanie niektorých položiek

Ak chceme vypísať zoznam všetkých položiek (stĺpcov) danej tabuľky, použijeme znak * - hviezdičku. Hviezdička symbolizuje slovo „všetko“. Výsledok sme si ukázali vyššie.

Ak chceme vypísať len niektoré stĺpce z danej tabuľky, nahradíme hviezdičku vypísaním názvov požadovaných položiek v príkaze:

```
SELECT položka_1, položka_2, ...položka_n FROM meno_tabuľky
```

Ak zadáme príkaz:

```
mysql>select id, nazov, autor from kniha;
```

dostaneme výsledok, kde sú zobrazené iba stĺpce **id**, **nazov** a **autor**, ostatné sú ignorované. Zobrazí sa teda zredukovaná tabuľka ako je na výpise č.5-2:

```
mysql> select id,nazov,autor from kniha;
```

id	nazov	autor
1	Angelika a kral	Golon, Anne a Serge
2	KGB	Gordijevsky, Oleg
3	Bratia Ricovci	Simenon, Georges
4	Vtaky v trni	McCulloughova, Collen
5	Linux - prakticky pruvodce	Sobell, Mark G.
6	Naucte se programovat v Delphi	Binzinger, Thomas
7	Pouzivame linux	Welsh, M., Kaufman, L.

Agregačné funkcie

Agregačné funkcie sú typy dopytov, ktoré spracovávajú hodnoty z celých stĺpcov tabuľky. Zadávame ich do zoznamu výstupných položiek za príkaz SELECT.

V MySQL existujú tieto najpoužívanejšie agregačné funkcie:

SUM() - súčet numerických hodnôt v danom stĺpci

MIN() - nájdenie minimálnej hodnoty v danom stĺpci

MAX() - nájdenie maximálnej hodnoty v danom stĺpci

COUNT() - počet hodnôt (záznamov) v danom stĺpci

AVG() - aritmetický priemer numerických hodnôt v danom stĺpci

Vyskúšajme si tieto funkcie.

SUM()

Ak by sme chceli vedieť, aká je hodnota (hodnota = súčet cien jednotlivých kníh) našej knižnice. Pre zjednodušenie predpokladajme, že máme iba po jednom kuse z každej knihy.

Použijeme funkciu *SUM(cena)* takto:

```
mysql> select sum(cena) from kniha;
```

a dostaneme tento výpis č.5-3:

```
mysql> select sum(cena) from kniha;
```

sum(cena)
2385.00

Že sa nám nepáči názov stĺpca *sum(cena)*? Nevadí, ak by naša účtovníčka nepoznala tieto pojmy, môžeme pre ňu upraviť príkaz *select* takto:

```
mysql> select sum(cena) as 'Hodnota knih' from kniha;
```

Slovko **as** spôsobí, že premenuje stanovenú funkciu (v tomto prípade „*sum(cena)*“) na stanovený text, napr. *Hodnota knih*. Potom dostaneme výpis č.5-4:

```
mysql> select sum(cena) as 'Hodnota knih' from kniha;
+-----+
| Hodnota knih |
+-----+
|          2385.00 |
+-----+
```

No a to už je zrozumiteľnejší výpis.

MIN(), MAX()

Veľmi podobným spôsobom môžeme zistiť minimálnu alebo maximálnu cenu kníh v našej knižnici. Použijeme na to funkcie *MIN* a *MAX* s využitím formulky *AS 'konkrétny_text'*:

```
mysql> select min(cena) as 'Najmensia cena', max(cena) as 'Najvacsia cena knih' from kniha;
```

Výsledok je na výpise č.5-5:

```
mysql> select min(cena) as 'Najnizsia cena',
-> max(cena) as 'Najvacsia cena knih',
-> from kniha;
+-----+-----+
| Najnizsia cena | Najvacsia cena knih |
+-----+-----+
|          18.00 |          1073.00 |
+-----+-----+
```

Ak sa pozrieme do našej tabuľky **KNIHA**, vidíme, že tieto vyselektované hodnoty sú skutočne správne.

AVG()

Ak by nás zaujímala priemerná cena kníh v našej knižnici, použijeme funkciu *AVG()* takto:

```
mysql> select avg(cena) as 'Priemerna cena knih' from kniha;
```

a dostaneme výpis č.5-6:

```
mysql> select avg(cena) as 'Priemerna cena knih'
-> from kniha;
+-----+
| Priemerna cena knih |
+-----+
|          340.714286 |
+-----+
```

Count()

Ak sa chceme dozvedieť, koľko že to máme teraz v knižnici evidovaných kníh, použijeme funkciu *COUNT()*, ktorá zisťuje počet položiek v danom stĺpci.

Zadajme príkaz:

```
mysql> select count(*) as 'Pocet knih' from kniha;
```

a tu je výsledok (výpis č. 5-7):

```
mysql> select count(*) as 'Pocet knih'
-> from kniha;
+-----+
| Pocet knih |
+-----+
|          7 |
+-----+
```

Podme sa teraz pozrieť na „druhý koniec“ príkazu *SELECT*:

Formulácia podmienky WHERE

Formaláciu podmienky **WHERE** už trocha poznáme aj z iných príkazov SQL. Použili sme ju pri mazaní (*delete*) a upravovaní (*update*) jednotlivých záznamov v minulej časti seriálu. Aj pri príkaze **SELECT** spĺňa rovnakú úlohu - vyselektujú sa len tie záznamy, ktoré podmienke vyhovujú, ostatné záznamy sa ignorujú. Obsahom podmienky môžu byť:

- n operácie s textovým reťazcom (rovnosť, nerovnosť, podobnosť)
- n matematické operácie (rovnosť, nerovnosť, väčší než, menší než)
- n logické operácie (AND, OR, NOT)

Operácie z textovým reťazcom

Pri vyhľadávaní s formulkou **WHERE** s textovým reťazcom použijeme tento všeobecný zápis:

**SELECT položka_1, položka_2, ...položka_n FROM meno_tabuľky
WHERE názov_filtračného_stĺpca OPERAND hľadaný_reťazec**

Za **názov_filtračného_stĺpca** dosadíme názov stĺpca, podľa ktorého stanovujeme podmienku. Ak chceme vyhľadávať podľa názvu knihy, použijeme **NAZOV**, ak podľa ceny, použijeme **CENA**.

Za **OPERAND** dosadíme:

- „ = “ (rovná sa), ak sa má porovnať presný text podmienky,
- „ <> “ (nerovná sa), ak sa nemá vôber rovnať hľadanému reťazcu
- LIKE** (podobnosť), ak sa má hľadaný reťazec podobovať.

Za **hľadaný_reťazec** dosadíme stanovený text do apostrofov.

Tu je niekoľko príkladov:

= (rovná sa)

Ak by sme chceli vyhľadať názov, autora a cenu kníh, ktorých **VYDAVATEL** je vydavateľstvo Computer Press, použijeme príkaz:

```
mysql> select nazov, autor, cena from kniha where vydavatel = 'Computer Press';
```

Na výpise č.5-8 vidíme, že stanovenej podmienke vyhoveli dva záznamy:

```
mysql> select nazov, autor, cena from kniha
-> where vydavatel = 'Computer Press';
```

nazov	autor	cena
Linux - prakticky pruvodce	Sobell, Mark G.	1073.00
Pouzivame linux	Welsh, M., Kaufman, L.	494.00

<> (nerovná sa)

Naopak, ak by sme chceli vypísať všetky záznamy o knihách, len nie tie, ktoré sú z vydavateľstva Computer Press, zapíšeme podmienku takto:

```
mysql> select nazov, autor, cena from kniha where vydavatel <> 'Computer Press';
```

Výsledkom bude výpis zvyšných piatich záznamov, tak ako na výpise č.5-9:

```
mysql> select nazov, autor, cena from kniha
-> where vydavatel <> 'Computer Press';
```

nazov	autor	cena
Angelika a kral KGB	Golon, Anne a Serge	56.00
Bratia Ricovci	Gordijevsky, Oleg	239.00
Utaky v trni	Simenon, Georges	18.00
Naucete se programovat v Delphi	McCulloughova, Collen	66.00
	Binzinger, Thomas	439.00

LIKE

Ak nevieme presne názov knihy, nemôžeme použiť operand = . Ak vieme, aké slovo alebo časť slova daný názov obsahuje, použijeme **LIKE** a tzv. *divoké znaky*, ktorým hovoríme aj *žolíky* (jokes). Tieto v textových reťazcoch fungujú podobne ako žolíky v kartách - nahrádzajú iné znaky.

Divoké znaky sú v SQL databázach dva :

% - **percento**, ktoré nahrádza ľubovoľnú skupinu znakov

_ - **podtržítka** (podčiarkovník), ktorý nahrádza iba jeden znak.

Predstavme si, že existuje akási kniha o **programovaní**, ale nevieme jej presný názov. Vtedy použijeme príkaz:

```
mysql> select nazov, autor, vydavatel, cena from kniha
      where nazov like '%program%';
```

Výsledok je na výpise č.5-10:

```
mysql> select nazov, autor, vydavatel, cena
      -> from kniha
      -> where nazov like '%program%';
```

nazov	autor	vydavatel	cena
Naučte se programovat v Delphi	Binzinger, Thomas	Grada	439.00

Treba si uvedomiť, že v tomto ilustračnom príklade, kde máme iba sedem záznamov, je výsledok tohto hľadania pomerne jednoznačný. V praxi, kde sú v tabuľkách milióny záznamov, je nutné čo najpresnejšie popísať danú podmienku pre vyhľadávanie, aby sme dostali čo najkonkrétnejšie výsledky, inak by vyššie uvedenej podmienke mohli vyhovieť aj stovky knižných titulov.

Preto sa často používa aj znak _ podtržítka, ktoré nahrádza iba jeden znak v podmienke.

Podmienka s matematickými operáciami

Práca s príkazom SELECT, kde je v podmienke matematická operácia sa oveľa ľahšie líši od práce s textovým reťazcom. Je tu iba jedna zmena - číselné hodnoty sa nezapisujú do apostrofov.

Pri vyhľadávaní s formulkou **WHERE** s číselnou hodnotou použijeme tento všeobecný zápis:

```
SELECT položka_1, položka_2, ...položka_n FROM meno_tabuľky
WHERE názov_filtračného_stĺpca OPERAND číselná_hodnota
```

Za operand môžeme dosadiť:

„=“ (rovná sa)

„<>“ nerovná sa

„<“ menší než

„>“ väčší než

Tu je niekoľko príkladov:

Ak chceme vyhľadať knihy, ktorých cena je rovná konkrétnemu číslu, použijeme zápis:

```
mysql> select nazov, autor, cena from kniha
      where cena = 66;
```

Výsledok je na výpise č.5-11:

```
mysql> select nazov, autor, cena
-> from kniha
-> where cena = 66;
```

nazov	autor	cena
Utaky v trni	McCulloughova, Collen	66.00

Znakom <> (nerovná sa) by sme analogicky vypísali všetky ostatné knihy, len nie tie, čo stáli 66 korún, tak ako je to na výpise č. 5-12:

```
mysql> select nazov, autor, cena
-> from kniha
-> where cena <>66;
```

nazov	autor	cena
Angelika a kral	Golon, Anne a Serge	56.00
RGB	Gordijevsky, Oleg	239.00
Bratia Ricovci	Simenon, Georges	18.00
Linux - prakticky pruvodce	Sobell, Mark G.	1073.00
Naučte se programovat v Delphi	Binzinger, Thomas	439.00
Pouzivame linux	Welsh, M., Kaufman, L.	494.00

Taktiež môžeme stanoviť podmienku > (väčší než) alebo < (menší než).
Knihy drahšie ako 400 Sk sú na výpise č.5-13:

```
mysql> select nazov, autor, cena
-> from kniha
-> where cena > 400;
```

nazov	autor	cena
Linux - prakticky pruvodce	Sobell, Mark G.	1073.00
Naučte se programovat v Delphi	Binzinger, Thomas	439.00
Pouzivame linux	Welsh, M., Kaufman, L.	494.00

Knihy lacnejšie ako 400 Sk sú na výpise č.5-14:

```
mysql> select nazov, autor, cena
-> from kniha
-> where cena < 400;
```

nazov	autor	cena
Angelika a kral	Golon, Anne a Serge	56.00
RGB	Gordijevsky, Oleg	239.00
Bratia Ricovci	Simenon, Georges	18.00
Utaky v trni	McCulloughova, Collen	66.00

Podmienka s logickými operáciami

Logické operácie v podmienkách príkazu SELECT sú tesne zviazané s vyššie uvedenými matematickými alebo textovými operáciami. Pomocou logických operandov môžeme spájať niekoľko podmienok do jedného príkazu. Obečný popis príkazu SELECT s využitím logických operácií vyzerá takto:

```
SELECT položka_1, položka_2, ...položka_n FROM meno_tabuľky  
WHERE podmienka_1 LOGICKÝ_OPERAND podmienka_2 LOGICKÝ_OPERAND podmienka_n
```

Za logický operand môžeme použiť:

AND - a zároveň

OR - alebo

NOT (!=) - negácia = zápor

Záverom niekoľko príkladov:

AND

Ak hľadáme knihy, ktoré patria do kategórie odbornej literatúry a **ZÁROVEŇ** ich cena je väčšia ako 450 Sk, zadáme príkaz:

```
mysql> select nazov, autor, vydavatel from kniha
      where cis_odd = 7 AND cena > 450;
```

```
mysql> select nazov, autor, vydavatel
      -> from kniha
      -> where cis_odd = 7 AND cena > 450;
```

nazov	autor	vydavatel
Linux - prakticky pruvodce	Sobell, Mark G.	Computer Press
Pouzivame linux	Welsh, M., Kaufman, L.	Computer Press

Vidíme, že tejto zloženej podmienke vyhovujú dva záznamy.

OR

Ak hľadáme knihy, ktoré vydalo nakladateľstvo Computer Press **ALEBO** Grada, zadáme príkaz:

```
mysql> select nazov, autor, vydavatel from kniha
      where vydavatel = 'Computer Press' OR vydavatel = 'Grada';
```

```
mysql> select nazov, autor, vydavatel
      -> from kniha
      -> where vydavatel = 'Computer Press' OR vydavatel = 'Grada';
```

nazov	autor	vydavatel
Linux - prakticky pruvodce	Sobell, Mark G.	Computer Press
Naucte se programovat v Delphi	Binzinger, Thomas	Grada
Pouzivame linux	Welsh, M., Kaufman, L.	Computer Press

Tentokrát vyhoveli podmienke tri záznamy.

!= (NOT)

Ak chceme nájsť knihy, ktoré NEVYDALO nakladateľstvo EAAP, zadáme príkaz:

```
mysql> select nazov, vydavatel, cena from kniha
      where vydavatel != 'EAAP';
```

Výsledok je na výpise č.5-17:

```
mysql> select nazov, vydavatel, cena
      -> from kniha
      -> where vydavatel != 'EAAP';
```

nazov	vydavatel	cena
Angelika a kral	Slovensky spisovatel	56.00
Bratia Ricovci	Smena	18.00
Utaky v trni	Slovensky spisovatel	66.00
Linux - prakticky pruvodce	Computer Press	1073.00
Naucte se programovat v Delphi	Grada	439.00
Pouzivame linux	Computer Press	494.00

Isto nás napadne, že je možné tieto typy podmienok vzájomne kombinovať, aby sme mohli vytvoriť aj veľmi zložitú podmienku.

Nabudúce si vysvetlíme ďalšie parametre príkazu SELECT.

Miroslav Oravec

Poznámka:

Internetová stránka, venovaná tomuto seriálu sa presťahovala z určitých dôvodov na adresu mior.host.sk. Zároveň prosím o pochopenie, že nie je ešte úplná, lebo pracujem na jej vývoji.