

10. Databázy

1

LIMIT, funkcie v MySQL

Obsah

2

- Obmedzenie výsledku dopytu – LIMIT
- Funkcie v MySQL
- Projekt č. 2
- Informácie o skúške

Obmedzenie rozsahu výsledku dopytu – LIMIT

3

- Pri práci s údajmi v databáze niekedy potrebujeme obmedziť výsledný počet riadkov, resp. obmedziť počet riadkov, ktorých sa má daný dopyt týkať (napr. `SELECT`, `UPDATE` alebo `DELETE`)
- Môžeme použiť klauzulu **LIMIT** – musí ísť až po `WHERE` a `ORDER BY`
- `SELECT ... LIMIT max_pocet_riadkov`
- `SELECT ... LIMIT zaciatok, max_pocet_riadkov`
- `UPDATE ... LIMIT max_pocet_riadkov`
- ...
- **zaciatok** sa čísluje **od 0**
- Využíva sa často pri vyhľadávaní, stránkovaní, atď.

LIMIT – príklady

4

- **SELECT * FROM tour_objednavky LIMIT 5**
vyberie prvých max. 5 záznamov z tabuľky
- **SELECT * FROM tour_objednavky LIMIT 6, 3**
vyberie max. 3 záznamy z tabuľky od riadku č. 6
(začína sa číslovať **od 0**)
- **DELETE tour_objednavky WHERE id_pouz='1'**
LIMIT 1
zmaže **prvú** objednávku používateľa s ID=1

Využitie LIMIT – stránkovanie

5

- Ak sa na stránke zobrazuje veľa údajov/záznamov, môže to byť veľký objem dát.
- Môžeme ich rozdeliť na niekoľko stránok (počet záznamov na stránku si definujeme).
- Využitie v našom systéme – výpis objednávok
- Ako zistiť, na koľko stránok rozdeliť výpis?
 - Počet záznamov / počet na stranu => zaokrúhlenie nahor = funkcia **ceil(číslo)** // `ceil(4.3) -> 5`

Zistenie počtu riadkov tabuľky

6

- **Riešenie s doterajšími znalosťami:**
 - `SELECT nejaky_stlpec FROM tabuľka`
 - `mysql_num_rows()`
- **Nevýhoda:** vyberú sa všetky riadky tabuľky aj keby ich bolo niekoľko tisíc => vyberá sa zbytočne veľa údajov (potrebujeme len 1 číslo)
- **Nové riešenie** =>

MySQL – funkcie na stĺpcoch

7

- **count** (počet), **avg** (priemer), **sum** (súčet), **min**, **max**
- **Výhoda: výsledkom je vždy len 1 riadok**
- **SELECT max(cena) FROM tour_destinacie**
- **SELECT count(id) FROM tour_objednavky**
- **SELECT count(id) as pocet FROM tour_objednavky**

count(id)

3

pocet

3

MySQL funkcie – využitie

8

- Zistiť, kto má najviac objednávok
- Ktorý zájazd je najlacnejší/najdrahší
- Zistiť, koľko objednávok má konkrétny používateľ
- Spočítať, koľkokrát si daný používateľ objednal izbu s výhľadom na more (izba_more)
- ...

Stránkovanie – riešenie

9

```
$pocet_obj = daj_pocet_objednavok();  
$na_stranu = 3;  
$pocet_stran = ceil($pocet_obj / $na_stranu);  
  
// strany budeme číslovať od 1  
for($i = 1; $i <= $pocet_stran; $i++) {  
    echo "<a href='\">$i</a> ";  
}
```

Stránkovanie – problémy

10

- Ako povedať php stránke, ktorú časť údajov (stránku) chceme zobrazit?
 - Môžeme využiť adresu prehliadača (ako pri úprave).
 - Môžeme si poslať **číslo strany** / začiatok a koniec / ...
- Ako vybrať z databázy len tie údaje, ktoré chceme zobrazit?
 - Musíme prepočítať vstupy do klauzuly LIMIT podľa čísla strany.
 - LIMIT 0, 3 ; LIMIT 3, 3 ; LIMIT 6, 3 ; ...
 - LIMIT (**\$strana - 1**) * **\$na_stranu**, **\$na_stranu**;

Stránkovanie – číslo strany ako vstup

11

- Stačí, keď do odkazu vložíme názov parametra s hodnotou, napr. **str=2**.
 - Vstup stránky v adrese prehliadača musíme dať za znak ?, napr. **?str=2**
 - Nemusíme uvádzať názov súboru, lebo sa použije aktuálna stránka
- echo "\$i ";

Stránkovanie – výber správnych záznamov

12

- Treba počítať s tým, že číslo strany ako vstup nebude existovať, resp. môže byť hocičo => **treba ošetriť!**

```
if (isset($_GET['str'])) $strana = $_GET['str'];  
else $strana = 1;
```

- Záznamy sa číslujú od **0**, teda
- `$sql = "SELECT ... LIMIT " . ($strana - 1) * $na_stranu . ", $na_stranu";`

Projekt č. 2 – základné pravidlá

14

- Musíte pokračovať v projekte č. 1.
- Odolnosť voči vstupom z formulára a jednoduchým zmenám adresy prehliadača.
- Musí byť použitá databáza, aspoň 2 tabuľky, ktoré nesmú byť nezávislé (**povinná súčasť projektu**).
- Údaje na prácu s databázou **MUSIA** byť na 1 mieste.
- Projekt umožní pracovať s údajmi (vyberať, vkladať, upravovať a mazať) **s využitím databázy (povinná súčasť projektu)**.
- Projekt musí bežať na serveri DAVINCI.
- Projekt sa musí odlišovať od prednášok a cvičení.
- Projekt musí byť syntakticky odladený.
- Termín odovzdania: **11.5.2013 do 23:55**.
- Odovzdávanie len cez systém Moodle (**len vo formáte .zip**). Iným spôsobom odovzdané riešenie nebudeme akceptovať.

Projekt č. 2 – hodnotenie

15

- výpis údajov [**1b**]
 - vyhľadávanie, resp. filtrovanie údajov (SELECT + WHERE) [**2b**]
 - usporiadavanie údajov aspoň podľa 2 rôznych položiek (SELECT + ORDER BY) [**2b**]
- pridávanie údajov [**2b**]
- aktualizácia údajov [**2b**]
- mazanie údajov [**1b**]
- stránkovanie údajov (LIMIT) [**1b**]
- odolnosť voči zlým vstupom nielen vo formulári, ale aj zmenou parametrov v adrese prehliadača [**2b**]
- ošetrovanie práce s databázou (+ príslušné výpisy) [**1b**]
- návrh databázy + export vo formáte SQL [**2b**]
- rozsah, náročnosť, formátovanie zdrojového kódu, komentáre kódu, validita zdrojového kódu [**2b**]
- Ak **nebude** formulár user-friendly + neuvedenie typov vstupov [**mínus body**]

Skúška

16

- Pri počítačoch v H3, H6, dĺžka 3,5 hodiny
- K dispozícii budú všetky materiály zo systému Moodle
- Budete môcť využívať prinesené tlačené materiály, knihy, príručky atď.
- Nebudete môcť využívať žiadne komunikačné programy, notebooky, USB zariadenia, CD
- Opravovanie bude po skúške (presný čas a miesto oznámime na skúške v daný deň)
- **Varovanie:** Akékoľvek opisovanie a kopírovanie riešení a iné podvádzanie budeme riešiť disciplinárnym konaním u dekana fakulty.
- **Treba si priniest' ISIC a index**

Skúška – termíny

17

- Termíny budú známe najskôr v štvrtok 2.5.2013.
- Prihlasovanie na skúšku: **cez systém AIS**
- Odhlásiť z termínu skúšky sa môžete **12 hod.** pred jeho začiatkom.

Ďakujem za pozornosť 😊

18

Priestor na vaše otázky