

7. Databázy

1

Vkladanie a vyberanie údajov

Obsah

2

- Klúče
- Vkladanie údajov – INSERT
- Vyberanie údajov – SELECT

Identifikácia údajov

3

- Prečo nepostačuje doterajšia tabuľka `tour_objednavky`?
- Nič nám nebráni, aby sme zadali 2x ten istý záznam (všetky rovnaké údaje).
- Objednávky tak nemôžeme jednoznačne identifikovať. Problém napr. pri úprave/mazaní.
- => **vytváranie kľúčov** v tabuľkách (najčastejšie identifikátor záznamu – **ID**)

Kľúč

4

- Jednoznačná identifikácia záznamu (riadka) v tabuľke
- **Primárny** (v 1 tabuľke môže byť najviac 1)
- **Unikátny** (v 1 tabuľke ich môže byť viac)
- Môže sa skladať z viacerých stĺpcov
- Kľúč – iné pomenovanie je index

Vlastnosť – auto_increment

5

- Ako priradovať/zvyšovať čísla záznamov?
- Môžeme nastaviť atribút **auto_increment**, ktorý nastaví hodnotu na **maximum + 1**
- Tento atribút môže mať nastavený LEN kľúč
- V tabuľke `tour_objednavky` pridáme položku **id** (primárny kľúč), ktorou budeme identifikovať objednávky a nastavíme jej vlastnosť **auto_increment**.

#	Stĺpce	Typ	Zotriedenie	Atribúty	Nulový	Predvolené	Extra	Akcia
<input type="checkbox"/>	1 id	smallint(6)			Nie	None	AUTO_INCREMENT	Zmeniť Odstrániť More ▼
<input type="checkbox"/>	2 meno	varchar(30) utf8_slovak_ci			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	3 dospeli	tinyint(4)			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	4 deti	tinyint(4)			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	5 zajazd	smallint(6)			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	6 poistenie	tinyint(1)			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	7 izba_more	tinyint(1)			Nie	None		Zmeniť Odstrániť More ▼
<input type="checkbox"/>	8 kurz_potapania	tinyint(1)			Nie	None		Zmeniť Odstrániť More ▼

Vkladanie údajov – INSERT (1)

6

- **INSERT INTO nazov_tab SET** poloz1='hodn1', poloz2='hodn2', ... polozX='hodnX'
- **INSERT INTO nazov_tab** (poloz1, poloz2, ... polozX) **VALUES** ('hodn1', 'hodn2', ... 'hodnX')
- Nemusia byť uvedené všetky položky
- Nemusia byť v poradí ako sú definované v tabuľke
- **INSERT INTO tour_objednavky SET**
meno='Janko Hraško', dospelci='2', deti='4',
zajazd='8'

Zápis objednávky do DB

7

- Ako presne zapísať dopyt na vloženie objednávky do tabuľky?
- Musíme využiť pole **\$_POST** (odoslané údaje).
- **Pozor na úvodzovky a apostrofy pri dopyte!**
- Vytvoríme funkciu **zapis_objednavku()**

- ```
$sql = "INSERT INTO tour_objednavky SET meno=" .
$_POST['meno'] . "
dospeli=" . $_POST['pocet_dospelych'] . "
deti=" .
$_POST['pocet_deti'] . "
zajazd=" . $_POST['zajazd']
";
```

# Vkladanie údajov – INSERT (2)

8

- **INSERT INTO nazov\_tab VALUES**  
( 'hodn1', 'hodn2', ... 'hodnN' )
- Musia byť uvedené všetky hodnoty položiek v tabuľke a v poradí, ako sú definované v tabuľke.
- Pri položke id môžeme uviesť špeciálnu hodnotu **NULL** (môžeme ju používať len pri auto\_increment kľúčoch a pri položkách, ktorým nastavíme, že môžu byť NULL).
- **INSERT INTO tour\_objednavky VALUES (NULL, 'Janko Hraško', '2', '4', '8', '0', '0', '0')**



# Odporúčania pri INSERT

- Výsledkom dopytu INSERT je hodnota TRUE alebo FALSE.
- **Hodnoty položiek tabuľky treba uvádzať v apostrofoch – kvôli bezpečnosti.**
- INSERT INTO tour\_objednavky SET meno='Janko Hraško', dospelci='2', deti='4', zajazd='8'
- Či sa údaje vložili do tabuľky si pozrieme v phpMyAdmin.
- **mysql\_insert\_id(spojenie)** – vráti hodnotu auto\_increment vygenerovanú posledným dopytom INSERT. Túto funkciu treba zavolať hneď po dopyte INSERT.

# Výpis objednávok

10

- Výpis všetkých objednávok zatiaľ vložíme na stránku administrácie (**login.php**).
- Vytvoríme funkciu **vypis\_objednavky()**

Počet objednávok: 4

| meno a priezvisko | počet dospelých | počet detí | destinácia | poistenie | izba ku moru | kurz potápania |
|-------------------|-----------------|------------|------------|-----------|--------------|----------------|
| Janko Hraško      | 2               | 4          | 8          | 0         | 1            | 0              |
| Roman Hrušecký    | 2               | 1          | 2          | 1         | 0            | 1              |
| Janka Pekárová    | 2               | 0          | 5          | 1         | 1            | 0              |
| Jožko Mrkvička    | 5               | 3          | 9          | 0         | 1            | 0              |

Už ste prihlásený. Ak chcete, môžete sa odhlásiť.

# Vyberanie údajov – SELECT

11

- **SELECT** poloz1, poloz2, ... polozN **FROM** nazov\_tab
- **SELECT \* FROM** nazov\_tab  
(\* = **všetky stĺpce tabuľky**)
- Uvedený dopyt **SELECT** vyberie zadané (alebo všetky) stĺpce **zo všetkých riadkov tabuľky**.

# Ako spracovať výsledok

12

- Dopyt SELECT vráti výsledok (**result**) – zoznam riadkov, ktoré vyhovujú dopytu alebo hodnotu FALSE.
- S výsledkom nemôžeme len tak pracovať, musíme použiť ďalšie funkcie.
- **mysql\_num\_rows(result)** – vráti počet riadkov výsledku
- **mysql\_fetch\_assoc(result)** – vráti/vyberie jeden riadok (nasledujúci) z výsledku ako asociatívne pole (**indexy pol'a = názvy stĺpcov**)

# Štruktúra stránky pri SELECT

13

```
...
$result = mysql_query($sql, $link);
if ($result) {
 while ($row = mysql_fetch_assoc($result)) {
 // spracuj jednotlivé riadky výsledku dopytu
 // pristupujeme cez $row['nazov_stlpca']
 }
 mysql_free_result($result);
} else { // dopyt sa NEpodaril }
...
```

# Funkcia vypis\_objednavky()

14

```
...
$sql = "SELECT * FROM tour_objednavky"; // definuj dopyt
$result = mysql_query($sql, $link); // vykonaj dopyt
if ($result) {
 echo '<p>Počet objednávok: ' . mysql_num_rows($result) . '</p>';
 echo '<table border="1">';
 echo '<tr><th>meno </th><th>počet dospelých</th> ... </tr>';
 while ($row = mysql_fetch_assoc($result)) {
 echo '<tr><td>' . $row['meno'] . '</td><td>' . $row['dospeli'] .
 '</td></tr>';
 }
 echo '</table>';
 mysql_free_result($result);
} ...
```

# Užitočné príkazy

15

- **mysql\_free\_result(result)** – uvoľní pamäť, ktorú zaberá výsledok **result** (používa sa pri dopytoch SELECT)
- **mysql\_error(spojenie)** – vráti chybovú správu pri poslednom dopyte
- **mysql\_errno(spojenie)** – vráti číslo chyby pri poslednom dopyte

# Rozšírenia dopytu SELECT (1)

16

- Môžeme vybrať len **niektoré riadky podľa podmienky, prípadne ich usporiadať.**
- SELECT ... **WHERE podmienka** – spresnenie výberu (výsledkom nemusia byť všetky riadky)
- SELECT \* FROM tour\_objednavky **WHERE poistenie='1'**
- SELECT ... **WHERE podmienka1 AND podmienka2**
- SELECT ... **WHERE podmienka1 OR podmienka2**
- SELECT \* FROM tour\_objednavky **WHERE poistenie='1' AND dospeli>='2'**
- Podmienok môže byť viacero. Môžeme ich uzatvoriť do zátvoriek.



# Rozšírenia dopytu SELECT (2)

17

- SELECT ... **ORDER BY položka ASC/DESC** – usporiadanie výberu (ASC – vzostupne / DESC – zostupne)
- SELECT \* FROM tour\_objednavky **ORDER BY meno ASC**
- Môžeme usporiadať podľa viacerých stĺpcov – oddeľujeme čiarkou (pri každej položke uvedieme ASC alebo DESC).
- SELECT \* FROM tour\_objednavky **ORDER BY poistenie DESC, meno ASC**
- **Najprv musí ísť WHERE a až potom ORDER BY**
- SELECT \* FROM tour\_objednavky **WHERE poistenie='1'**  
**ORDER BY meno ASC**
- Neoplatí sa nahrádzať rozšírenia dopytu funkciami PHP.

# Najbližšie cvičenia, prednáška

18

- Cvičenia 28.3.2013 – nie sú (rektorské voľno)
- Prednáška 1.4.2013 – Veľkonočný pondelok



# Ďakujem za pozornosť 😊

19

Priestor na vaše otázky