

Informační systémy

MySQL

Martin Trnečka

Katedra informatiky
Univerzita Palackého v Olomouci

Předchozí přednášky

- Jazyk PHP - dynamické webové stránky (plnohodnotné).
- Stále chybí klíčový prvek: potřeba uložení dat.
- Velice omezené možnosti uložení dat (PHP nemá integrovaný žádný prostředek pro zpracování a manipulaci s databázovými daty = PHP samo o sobě neimplementuje žádnou databázi).
- Alternativa noSQL databáze.
- Potřeba plnohodnotného databázového systému.
- MySQL nejpopulárnější DB systém pro jazyk PHP (existují napojení na další jazyky jako Ruby, Perl, C++).
- Nativní podpora MySQL (potřeba instalovat standardní rozšíření).

MySQL

- Autoři Michael Widenius a David Axmark (Švédsko).
- Představena v roce 1995 (založen na mSQL, low level, pomalý).
- Pojmenovaná po dceři M. Wideniuse (My).
- Databázový systém primárně určený pro web.
- Implementuje dialekt SQL.
- Obsahuje celou řadu funkcí z pokročilých SBŘD.
- Od začátku optimalizovaný na rychlost = zjednodušení jazyka (některé prvky jsou zařazovány do jazyka až v poslední době, například trigger, view).
- Aktuálně pod záštitou Oracle.
- Volně ke stažení a použití.
- Součástí LAMP balíčků.
- Aktuálně ve verzi 5.7., více informací na <http://www.mysql.com/>
- Rozšířený mýtus: MySQL je pouze pro malé DB.

MySQL

- Založeno na SQL, již znáte (INSERT, UPDATE, DELETE, SELECT a další).
- Datové typy: TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, DECIMAL, FLOAT, DOUBLE, BIT, CHAR, VARCHAR, BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, ENUM, SET, DATE, TIME, DATETIME, TIMESTAMP, YEAR, GEOMETRY, POINT, LINESTRING, POLYGON, GEOMETRYCOLLECTION, MULTILINESTRING, MULTIPOINT, MULTIPOLYGON.
- Celá řada úložných engine (způsob ukládání dat, např. MyISAM, InnoDB). Spousta odlišností (podpora atomických operací, rychlost, přenositelnost, velikost zpracovaných dat).
- PRIMARY_KEY, FOREIGN_KEY, AUTO_INCREMENT, UNIQUE, indexy a další.
- Základní tutoriál na:
<http://www.mysqltutorial.org/basic-mysql-tutorial.aspx>.

Práce s MySQL v PHP

Příklad (Spojení s DB)

```
1 $link = mysql_connect("server", "user", "password", "database");
2
3 // populární konstrukce
4 $link = mysql_connect("server", "user", "password", "database") or die(mysql_error());
5
6 // Check connection
7 if(!$link) {
8     die('Could not connect: ' . mysql_error());
9 }
10
11 // změna databáze
12 mysql_select_db("database2", $link);
13
14 mysql_close($link);
```

`mysql_close()` ukončí poslední spojení. Není nutné spojení se automaticky ukončuje po vykonání skriptu (potřeba pro persistenci).

Dotazy na DB

Příklad (Vytvoření DB)

```
1 $sql = 'CREATE DATABASE my_db';
2 if (mysql_query($sql, $link)) {
3     echo "Database my_db created successfully\n";
4 }
5
6 else {
7     echo 'Error creating database: ' . mysql_error() . "\n";
8 }
```

Příklad (Vytvoření tabulky)

```
1 $sql = "CREATE TABLE Persons (FirstName CHAR(30), LastName CHAR(30), Age INT)";
2
3 mysql_query($sql, $link);
```

Příkazy se vykonávají nad aktuální databází (vybrána pomocí `mysql_select_db`). Příkazy `mysql_create_db` a `mysql_drop_db` jsou již zastaralé.

Dotazy na DB

Základní prostředek `mysql_query`.

Příklad (Vytvoření dotazu)

```
1 $firstname = 'bart';
2 $lastname  = 'simpson';
3
4 $query = sprintf("SELECT firstname, lastname, address, age FROM friends
5   WHERE firstname='%s' AND lastname='%s'",
6   mysql_real_escape_string ($firstname), mysql_real_escape_string ($lastname));
7
8 $result = mysql_query($query);
```

`mysql_real_escape_string` zabezpečí správnost kódování, escape nebezpečných znaků. Nutnost v případě binárních dat.

Zpracování výsledku

- `mysql_fetch_row()` vrací výsledek dotazu (jeden řádek) jako číselné pole.
- `mysql_fetch_assoc()` vrací výsledek dotazu (jeden řádek) jako asociativní pole.
- `mysql_fetch_object()` vrací výsledek dotazu (jeden řádek) jako objekt

Příklad

```
1 $result = mysql_query("SELECT id, name FROM mytable");
2
3 // zjisti poctu radku ve vysledku dotazu
4 $num_rows = mysql_num_rows($result);
5
6 while ($row = mysql_fetch_array( $result )) {
7     printf("ID: %s Name: %s", $row["id"], $row["name"]);
8 }
```


Problémy s kódováním češtiny

Ve starších verzích problematické kódování diakritiky. Obvykle dva typy problémů:

- 1 Data se zobrazují ve špatném kódování, na serveru jsou uloženy korektně.
- 2 Data se zobrazují správně, ale databáze není přímo čitelná (špatné kódování).

Příklad (Řešení problémů s kódováním)

```
1 mysql_query("SET CHARACTER SET utf8");  
2 mysql_query("SET character_set_results=utf8");  
3 mysql_query("SET character_set_connection=utf8");
```

Bezpečnost

- Viz. předchozí přednášky.
- Je čistě na programátorovy, aby zajistil a ošetřil data ukládaná do DB.
- Šifrování hodnot v databázi (například hesla uživatelů). Například pomocí vestavěných PHP funkcí `md5`, `sha1`.
- Přímá podpora v MySQL, více na: <https://dev.mysql.com/doc/refman/5.7/en/encryption-functions.html>
- Základní systém oprávnění: lze garantovat přístup k DB a povolit či odepřít možnost provádět základní operace (`UPDATE`, `DELETE`, `SELECT`, `INSERT`) a oprávnění pro pokročilejší funkcionalitu (`TRIGGRY`).
- Běžná praxe: jeden web = jedna databáze = nemusíte řešit oprávnění.

Nástroje pro práci s MySQL

Velice populární webové nástroje (obvykle naprogramované v PHP).

- phpMyAdmin, <http://www.phpmyadmin.net/>. Robustní a velice rozsáhlý nástroj pro práci s DB. V poslední době snaha přehnanou vizualizaci, dynamičnost (AJAX) a další „komfortní“ funkcionalitu = poněkud těžkopádné.
- Adminer, <http://www.adminer.org/cs/>. Český projekt. Správa DB v jednom jediném souboru. Ne tak rozsáhlý jako phpMyAdmin, ale umí vše co je potřeba pro běžnou práci s DB.
- Další software klasický s GUI nebo konzolový.

Odbočka: ORM

- Současný trend Object Relation Mapping.
- Databázové tabulky koincidují s objekty. Standard ve většině webových frameworků ať už na bázi PHP či jiného jazyka.
- Zjednodušuje dotazování, vede na jednodušší a přehlednější kód.
- V duchu MVC architektury.
- Spíše pro pokročilé použití (přirozený vývoj).

Příklad (Doctrine 2)

```
1 /** @entity */
2 class Kniha
3 {
4     /** @id @column(type="integer") @GeneratedValue */
5     public $id;
6
7     /** @column(type="text") */
8     public $navez;
9 }
```


Důležité pojmy:


- MySQL a možnosti PHP pro komunikaci s MySQL serverem.

Poznámka:

Návrh databázového schématu. Běžná praxe: primary key vždy integer reprezentující identifikátor položky (pro i proti).

Čtení na doma:

 Ullman L., PHP and MySQL for Dynamic Web Sites, 4th Edition, 2012, ISBN 978-0-321-78407-0. (kapitoly 4-7, 9).

 <http://www.mysqltutorial.org/>