**DATABÁZOVÝ SYSTÉM MySQL**

K nastartování a správě DB je nejprve potřeba nainstalovat server klienta. Pro jednoduché lokální databáze je vhodným nástrojem **WampServer**, který obsahuje **Apache** – webový server, **PHP** a **MySQL**, to jsou základní součásti pro správný chod a správu databáze na lokálním PC. Ve WampServeru si můžete fungování nastavit v pohodlném grafickém nastavení, bez něho byste museli všechno nastavovat ručně a zdlouhavě.

K založení a správě databáze se potom používá jednoduché prostředí na ***localhostu*** zvané **phpmyadmin**. V tomto prostředí si programátor dokáže nastavit potřebné údaje k tomu, aby si vytvořil databázi přesně k jeho účelu.

K **založení databáze** stačí v phpmyadminu kliknout na tlačítko *Nová* v levém sloupečku stránky. Dále se napíše jméno databáze a vhodné kódování (utf8­\_czech\_ci) a stiskem tlačítka *Vytvořit*, je databáze vytvořena a lze ji vidět opět v levém sloupečku.

Pokud už nějakou databázi vlastníme ve formátu \*.sql, můžeme ji do phpmyadminu **importovat** pomocí tlačítka *Import* v horním vodorovném menu. Stejně tak si můžeme databázi **exportovat** na jiný počítač, opět stiskem *Export* v horním menu.

K **úpravě** databáze stačí jednu vybrat a pomocí grafického nastavení ji upravit. Samozřejmostí je, že ji můžeme upravit i pomocí SQL kódu buď v phpmyadmin anebo přes PHP script.

* *DROP DATABASE název – odstranění databáze (přijdeme o všechny tabulky a data)*
* *CREATE TABLE názevTabulky*

*(*

*Id Int NOT NULL AUTO\_INCREMENT,*

*Adresa Varchar(100) NOT NULL,*

*Telefon Varchar(12) NOT NULL,*

*);*

* *Vytvoření tabulky se sloupcemi*
* *INSERT INTO názevSloupce VALUES() – vložení dat do tabulky*

**Entita** je objekt reálného světa, který se jednoznačně odlišuje od jiného. Je to tedy něco, o čem potřebujeme v systému uchovat nějaké informace - **atributy**. (ČLOVĚK-jméno, příjmení; MĚSTO-název, poloha…).

**SQL a PHP**

Spojení SQL databáze pomocí PHP:

* *$con = mysqli\_connect(“serverName“, “jméno“, “heslo“, “názevDatabázeProPřipojení“);*

V proměnné $con dále vidíme úspěšný/neúspěšný status, zda jsem se připojili k databázi

Vložení dat do tabulky v databázi:

* $dotaz = “INSERT INTO tabulka (jmeno, prijmeni) VALUES (‘Petr’, ‘Novotný’)“; //vložení dat do tabulky
* $dotaz = “SELECT \* FROM tabulka“; // \* = vše, jde nahradit názvy sloupců(oddělené čárkou)

// můžeme vložit i podmínku WHERE, která nám vybere pouze data, které podmínku splňují: WHERE jmeno=„Petr“;

* $dotaz = “UPDATE tabulka SET jmeno=’Pavel’ WHERE id=2”; // Změna dat v tabulce

Jedná se o pouhý dotaz, co se má provést. Provádění se dělá následovně.

* $con->query($dotaz);

Tímto příkazem se provede dotaz do databáze, pokud vše proběhne správě.

**Hashování**

= Vstupní data jsou na výstupu razantně změněna.

**MD5** [md5(text)] – používal se u hesel, dnes už by se kvůli špatné bezpečnosti neměl používat. Byly vytvořené tzv. **rainbow** tables, které obsahují miliony kombinací hesel s předpočítaným hashem. MD5 hashovala 2 stejné text stejně, tzn., že když bylo v databázi několik stejných hesel, jejich hash byl taky stejný. Proto se pro větší bezpečnost zavedlo přidání „soli“ – tzn, že k heslu byl ještě připsán třeba email a výsledný hash byl jiný. Výsledná délka hashe je 32 znaků.

**BCRYPT** – Použití v PHP pomocí funkce **password\_hash(text)** – defaultně nastaven BCRYPT, ale můžeme změnit za jiný. Od MD5 se liší tím, že používá **náhodnou sůl** a kvalitnější algoritmus. Při stejném vstupu **nedostaneme vždy stejný výstup** (pokud jsou tedy dvě stejné hesla, z hashe to nepoznáme). Výstupem je řetězec s 60 znaky, který začíná znakem $ a ten má 3 části – **1.** určuje typ použitého algoritmu, **2.** je sůl, **3.** samotný hash ($2y$10$.vGA1O9wmRjrwAVXD98HNOgsNpDczlqm3Jq7KnEd1rVAGv3Fykk1a).K rozšifrování se pak musí použít funkce **password\_verify(heslo, hash)**, která vrátí true nebo false. Další výhodou je i jeho **pomalost** (>100ms).