

PHP

Adatbázis kezelés

Kétféle adatbázis elérési lehetőség PHP-ból:

- Natív: specifikusan egy adatbázis kezelő rendszerhez készített
 - Példák: „mysql_” illetve „mysqli_” kezdetű függvények
 - mysql_ függvények használata NEM ajánlott, használjuk a mysqli_ függvényeket
- Absztrakció: több adatbázis kezelő rendszerrel is használható. Előnye, hogy ha más típusú adatbázis kezelő rendszerre váltunk, akkor nem kell módosítani a kódot, hátránya, hogy nagyobb teljesítmény igényű, mint a natív megoldások
 - Példa: PDO kapcsolatkezelő

A mi adatbázis kezelő rendszerünk a MySQL, amit eddig csak weben, böngészőből értünk el a phpMyAdmin felület segítségével. Most a célunk, hogy a weboldalunkról is elérhessük.

Használat lépései

- Kapcsolódás az adatbázishoz
- SQL lekérdezés megírása
- SQL lekérdezés előkészítése, paraméterek helyének kihagyásával (ha van)
- Paraméterek behelyettesítése (ha van)
 - Az utólagos behelyettesítés védelmet nyújt SQL kódbeillesztés (SQL injection) ellen, illetve ugyanazt a lekérdezést többször futtathatóvá teszi különböző paraméterekkel
- Lekérdezés futtatása
- Eredmény kezelése
 - SELECT esetén az eredményhalmaz sorait végigolvassuk
 - INSERT, UPDATE, DELETE esetén azt vizsgáljuk, hogy sikerült-e a lekérdezés

Adatbázis tábla elkészítése:

Mezők:

- id (szám, INT)
 - azonosító, auto increment (egyedi minden sorra, folytonosan növekszik)
- nev (korlátos hosszú szöveg, VARCHAR)
 - véleményező neve
- email (korlátos hosszú szöveg, VARCHAR)
 - véleményező email címe
- szoveg (korlátlan hosszú szöveg, TEXT)
 - vélemény
- datum (dátum, TIMESTAMP)
 - vélemény bejegyzésének dátuma (bejegyzéskor, módosításkor automatikusan frissül)

```
CREATE TABLE IF NOT EXISTS `velemeny` (  
  `id` int(10) unsigned NOT NULL auto_increment,  
  `nev` varchar(100) character set utf8 NOT NULL,  
  `email` varchar(100) character set utf8 NOT NULL,  
  `szoveg` text collate utf8_unicode_ci NOT NULL,  
  `datum` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
```

Magyarázat a továbbiakban, a kódban kommentként szerepel (// kezdetű sorok)

Kapcsolódást végző függvény:

```
function connect() {  
  // kapcsolódás az adatbázishoz  
  // paraméterek:  
  // - adatbázis helye (localhost, ha ugyanazon a szerveren, ahol a kód fut,  
  // egyébként a szerver URL-je)  
  // - adatbázis felhasználónév  
  // - adatbázis jelszó  
  // - adatbázis neve (atw.hu-n ugyanaz, mint a felhasználónév)  
  $mysqli = new mysqli("localhost", "my_user", "my_password", "my_db");  
  
  // ellenőrizzük, hogy sikeresen kapcsolódunk-e, ha nem, akkor kiírjuk a hibát  
  if (mysqli_connect_errno()) {  
    print "Kapcsolat hiba: " . mysqli_connect_error();  
    exit();  
  }  
  
  return $mysqli;  
}
```

Select:

```
function getComments() {
    $mysqli = connect();

    // előkészítjük a lekérdezést, azt megadjuk az adatbázis kapcsolatnak
    // kapunk egy Statement-et, amit eltárolunk egy változóba
    $stmt = $mysqli->prepare(
        "SELECT nev,email,szoveg FROM velemeny ORDER BY datum DESC");
    // lekérdezés futtatása
    $stmt->execute();
    // eredményeket a paraméterként megadott változókba teszi bele
    $stmt->bind_result($name, $email, $comment);
    // végigmegyünk az eredményhalmaz sorain
    // (fetch() mindig a következő sorra ugrik, amíg van)
    while ($stmt->fetch()) {
        // kiírjuk az adott sorát az eredményhalmaznak, felhasználva az előbb
        // megadott változókat, amik a megfelelő mező értékét tartalmazzák
        print '<a href="mailto:'. $email. '">'. $name. '</a><br/>'
            . $comment. '<br/><br/>';
    }

    // statement és kapcsolat bezárása
    $stmt->close();
    $mysqli->close();
}
```

Insert:

```
function addComment() {
    // Leellenőrizzük, hogy a Submit gomb megnyomásra került-e, vagyis vizsgáljuk,
    // hogy létezik-e commentSubmit névvel kérdésben érkezett paraméter
    if (isset($_REQUEST['commentSubmit'])) {
        // kapcsolódást végző függvény meghívása, a kapcsolat
        // eltárolása a $mysqli változóba
        $mysqli = connect();

        // előkészítjük a lekérdezést, kihagyva a paraméterek helyét
        // (ahova mi helyettesítünk értéket)
        $stmt = $mysqli->prepare("INSERT INTO
            velemeny (nev,email,szoveg) VALUES (?, ?, ?)");
        // paraméterek behelyettesítésének megadása
        // első paraméter: típusok felsorolása
        // (p1: s = szöveg, i = egész szám, d = tetszőleges szám)
        $stmt->bind_param('sss',
            $_POST['name'], $_POST['email'], $_POST['comment']);
        // lekérdezés futtatása, eredményről tájékoztató szöveg kiírása
        if ($stmt->execute()) {
            echo 'Új vélemény bejegyezve!';
        }
        else echo 'Hiba a vélemény bejegyzésekor!';

        // statement és kapcsolat bezárása
        $stmt->close();
        $mysqli->close();
    }
}
```

Tehát a fontos függvények sorrendben

A nagybetűs részek értelemszerűen kitöltendőek.

- Kapcsolódás az adatbázishoz
 - `$mysqli = new mysqli("localhost", "MY_USER ", "MY_PASSWORD ", "MY_DB");`
- Lekérdezés előkészítése (paraméterekkel – ha van)
 - `$stmt = $mysqli->prepare("IDE JÖN A TETSZŐLEGES SQL LEKÉRDEZÉS, PARAMÉTEREKET KIHAGYVA");`
- Paraméterek behelyettesítése (ha van paraméter)
 - `$stmt->bind_param('PARAMÉTERTÍPUSOK JELEI', PARAMÉTEREK FELSOROLÁSA VESSZŐVEL ELVÁLASZTVA);`
- Lekérdezés futtatása
 - `$stmt->execute();`
 - INSERT, UPDATE, DELETE esetén
 - Ellenőrizzük a lekérdezés futtatásának sikerességét
 - `if ($stmt->execute()) {`
`echo 'Új vélemény bejegyzve!';`
`}`
`else echo 'Hiba a vélemény bejegyzésekor!';`
 - A `$stmt->execute()` logikai értékkel tér vissza (igaz vagy hamissal), mely azt jelzi, hogy sikeres volt-e a lekérdezés futtatása
- Eredményfeldolgozása (SELECT esetén)
 - Eredményhalmaz oszlopaihoz változók megadása
 - `$stmt->bind_result($name, $email, $comment);`
 - Eredményhalmazon következő sorára ugrás
 - `$stmt->fetch()`
 - Ezt egy while ciklusban szokták használni:
 - `while ($stmt->fetch()) { IDE JÖN A bind_result-nál MEGADOTT VÁLTOZÓK FELHASZNÁLÁSA, KIÍRATÁS }`
- Kapcsolat lezárás, erőforrások felszabadítása
 - `$stmt->close();`
 - `$mysqli->close();`