

PHP – Adatbázis

Tartalomjegyzék

PHP – ADATBÁZIS.....	1
TARTALOMJEGYZÉK.....	1
EDDIG VOLT	1
MAI ANYAG.....	1
TEHÁT RÉSZLETESEBBEN	1
<i>Kapcsolódás web-alkalmazásokból adatbázisokhoz.....</i>	<i>1</i>
<i>Biztonsági kérdések</i>	<i>2</i>
PHP	2
MySQL egyszerű általános interface.....	3
Oracle egyszerű általános interface	3
Egy példa alkalmazás	4
MA MEGTANULTUK.....	5
EZUTÁN JÖN	5

Eddig volt

- Már képesek vagyunk kliens oldali megjelenítésre (HTML)
- Ottani adatok ellenőrzésére (Java script futtatás)
- Adatok feltöltésére és szerver oldali programozásra (Webszerver, PHP)
- Ismerjük az architekturális lehetőségeket.

Mai anyag

- Megtanuljuk, hogyan lehet egy web-alkalmazással adatbázishoz csatlakozni
- Pontosan megnézzük a PHP lehetőségeit
 - MySQL-hez
 - Oracle-höz

Tehát részletesebben

Kapcsolódás web-alkalmazásokból adatbázisokhoz

- Az adott nyelvben saját függvények és/vagy osztályok vannak az egyes adatbázis-kezelő rendszerekhez. Ezekkel a függvényekkel egy interface-n keresztül tudjuk az adatbázisok által nyújtott lehetőségeket programozni.
- **NAGYON FONTOS!!!** Minden adatbázisművelet előtt hozzá kell kapcsolódnunk a megfelelő adatbázishoz, és mindig le kell zárunk a létrejött kapcsolatot a munka végeztével.
- Ha él a kapcsolat, akkor SQL utasításokat tudunk kiadni az adatbázisban az adott nyelv függvényeinek segítségével, vagyis az interface-n keresztül közvetlenül az adatbázisban.
- Lefuttathatunk utasításokat, lekérdezhetünk adatokat, eredmény-lista sorait visszakaphatjuk.
- Ezért nem csak a nyelv speciális utasításait kell ismernünk, hanem az SQL-vel is tisztában kell lennünk.
- *Van mód a modulárisabb programozásra is, de most erre részletesen nem térünk ki, csak a biztonságról szóló részben lesz egy kevés róla. (Design, programozás, adatbázis-alkalmazás fejlesztés, tartalomfejlesztés mind elkülönülhet egy projektben több ember végezheti.)*

Biztonsági kérdések

- Hogy lehet SQL utasításokat összeállítani?
 - Sztring-műveletekkel
 - Más módon (BIND)
- Közvetlen SQL utasításokat adjunk ki?
 - Miért ne? (INSERT INTO, DELETE, SELECT, több táblát érintenek(?))
 - Miért igen? (Tárolt eljárások (gyorsabb, átláthatóbb (modulárisabb), jogosultságok kezelése))
- Példák
 - Van két változónk: user="alma" és pwd="korte"
 - SQL a belépéshez: " SELECT count(1) from users where userid=""+user+" and passwd=""+pwd+""; "
 - Vagyis: " SELECT count(1) from users where userid='alma' and passwd='korte'; "
 - **DE HA**: user="" és pwd=" or '1'=1"
 - SELECT count(1) from users where userid=" and passwd=" or '1'=1"; **is BEENGED.**
 - Előfordulhat, hogy a string műveletekkel olyan a változó értéke, ami módosítja az SQL utasítást (pl az előbb.), ezért jobb a BIND használata, ahol egy adatbázisbeli változóhoz rendelhetünk hozzá egy külső (programnyelvi) változót, az értékek a memóriában cím szerint rendelődnek össze és a tartalom egy-az-egyben másolásra kerül.
 - " SELECT count(1) from users where userid=:userid and passwd=:passwd; ", majd a `:userid` adatbázisbeli változóhoz hozzárendeljük a `user` változót, illetve a `:passwd`-hez a `pwd`-t.
 - Mégjobb megoldás, a tárolt eljárás használata (persze BIND-dal), így ott még több műveletet lehet gyorsan végrehajtani: Menteni a rossz próbálkozások számát, letiltani egy idő után, jelszó cseréjét kezdeményezni, ha régóta ugyanaz, menteni az IP-címet, időt, ...

PHP

- A PHP a MySQL adatbázist nagyban támogatja, függvényei beépített függvények (teljes beépített interface).
- Az Oracle adatbázist nem támogatja, ehhez az interface-t elérhetővé kell tenni, sőt a PHP-t futató *webszerveren* egy *Oracle-klienst* is kell telepíteni az **OCI osztályokkal**, hogy használni tudja a szerveret.
- Két megközelítést nézünk meg:
 1. Egyszerű általános interface-t írunk, melyet SQL utasításokat futtathatunk adatbázisokban, illetve onnan adatokat kérdezhünk le. (*ma*)
 2. Oracle specifikus műveleteket próbálunk ki, nem általános interface-ben (BIND; tárolt eljárás; file fel-, letöltés/BLOB kezelés) (*következő órán*)
- 1. A következő műveletekre lesz szükségünk:
 - a. Connect: `DB_connect()`: Csatlakozás az adatbázishoz
 - b. Disconnect: `DB_disconnect()`: Az adatbáziskapcsolat megszüntetése
 - c. Execute: `DB_exec($sql)`: Futtatunk egy SQL utasítást
 - d. Select: `DB_query($sql)`: Futtat egy lekérdező SQL utasítást
 - e. Next row: `DB_getnextrow()`: Az eredmény halmaz következő sorát adja vissza

MySQL egyszerű általános interface

```
<?
$dbid      = 0;
$result    = 0;
$num_of_rows = 0;
$row       = 0;

function DB_connect()
{
    global $dbid;

    $dbid = mysql_connect("SZERVER", "USER", "PWD");
    mysql_select_db("ADATBAZIS");
}

function DB_exec($query)
{
    global $result;
    global $num_of_rows;

    $result = mysql_query($query);
    return $result;
}

function DB_query($query)
{
    global $result;
    global $num_of_rows;

    $result = mysql_query($query);
    if($result)
    {
        $num_of_rows = mysql_num_rows($result);
    } else {
        $num_of_rows = 0;
    }
    return $result;
}

function DB_getnextrow()
{
    global $row;
    global $result;

    if($result)
    {
        $row = mysql_fetch_assoc($result);
    } else {
        $row = 0;
    }
    return $row;
}

function DB_disconnect()
{
    global $dbid;
    global $result;

    mysql_free_result($result);
    mysql_close($dbid);
}

?>
```

Oracle egyszerű általános interface

```
<?
$conn      = 0;
$stmtmt    = 0;

function DB_connect()
{
    global $conn;

    $conn=OCILogon("USER_SCHEMA", "PWD", "ADATBAZIS/TNSALIAS");
}
```

```

        if (!$conn) {
            exit;
        }
    }

function DB_exec($query)
{
    global $conn;

    $stmt = OCIParse($conn,$query);
    if ($stmt) {
        if (OCIExecute($stmt, OCI_DEFAULT)) {
            OCICommit($conn);
            return 1;
        } else {
            OCIRollback($conn);
            return 0;
        }
    }
}

function DB_query($query)
{
    global $conn, $stmt;

    $stmt = ociparse($conn,$query);
    if ($stmt) {
        ociexecute($stmt,OCI_DEFAULT);
    }
    return $result;
}

function DB_getnextrow()
{
    global $conn, $stmt;

    if (OCIFetchInto($stmt,$row,OCI_ASSOC)) {
        return $row;
    }
}

function DB_disconnect()
{
    global $conn, $stmt;

    OCIFreeStatement($stmt);
    if ($conn) {
        OCILogoff($conn);
    }
}

?>

```

Egy példa alkalmazás

- Zenei adatokat tároljuk egy táblában. Listázzuk ki őket, törölhessünk, módosíthassunk illetve vehessünk fel új bejegyzést.
- Adatbázis létrehozása
 - <https://xml.inf.elte.hu/mysqlXWING/>
 - <https://xwing.inf.elte.hu/xoa/login.jsp>
 - create_table_with_data.sql
 - COMMIT
- DB.INC létrehozása (ez lesz a központi adatbázisválasztó file) + MYSQL.INC és ORACLE.INC módosítása
 - \$dbid = mysql_connect("xwing.inf.elte.hu", "wabpX", "wabp2005");
 - mysql_select_db("wabpX");
 - illetve
 - \$conn=OCILogon("wabpX_etrazon", "wabp2005", "XWINGDB");

- Keret létrehozása: index.html
- Menü: menu.html
- Listázás szűréssel: main.php
- Törlés: torol.php
- Szerkesztés: edit.php + beszur.php + javit.php

Ma megtanultuk

- Hogyan kell egy web-alkalmazással adatbázishoz csatlakozni,
- Hogy kell ezt csinálni PHP-ből MySQL-hez és Oracle-höz.

Ezután jön

- BIND
- BLOB kezelés
- Multimédiás tartalmak kezelése dinamikusan
- Letöltés / feltöltés

- dinamikus Flash