

SZÁMÍTÓGÉPES ADATBÁZIS-KEZELÉS

A MySQL adatbáziskezelő

PHP folytatás

JDBC, ODBC

ADATBÁZISOK

- ▶ Adatbázis = adott formátum és rendszer szerint tárolt adatok összessége.
- ▶ DBMS feladatai:
 - ▶ adatstruktúra (adatbázisséma) definiálása,
 - ▶ adatok aktualizálása (új felvétel, törlés, módosítás),
 - ▶ lekérdezési lehetőségek,
 - ▶ fejlesztő környezet biztosítása célalkalmazások létrehozásához.
- ▶ Relációs modell (1970): az adatok kétdimenziós táblázatokban tárolódnak

MYSQL

- ▶ MySQL: népszerű, ingyenes adatbáziskezelő rendszer (“DBMS”).
- ▶ A későbbiekben (PHP, JDBC, ODBC) arra látunk majd példákat, hogy lehet kapcsolódni hozzá
- ▶ Többfelhasználós, relációs alapelvekre épül
- ▶ A kurzus keretein belül a hozzá tartozó SQL nyelvjárást fogjuk használni
 - ▶ sajnos néhány így megtanult dolog nem fog működni pl. Oracle használatakor
- ▶ Adminisztrációjához parancssori eszközök (mysql, mysqladmin) vagy a **phpMyAdmin** rendszer használatos
- ▶ A kabinetben is elérhető

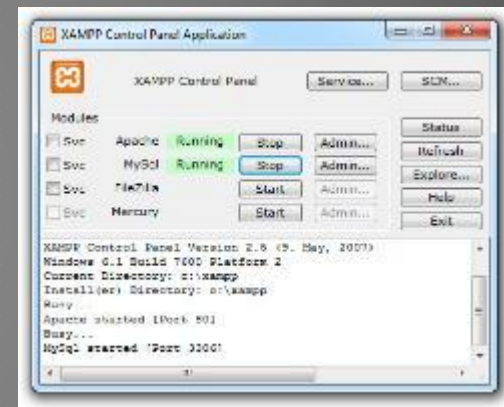
MYSQL / PHPMYADMIN

The screenshot displays the phpMyAdmin interface for a MySQL database on localhost. The interface is organized into several sections:

- Navigation Panel (Left):** Lists databases: geeklog, information_schema, mysql, phpmyadmin, sakila, sakila2, and test.
- Top Navigation:** Includes tabs for Adatbázisok, SQL, Állapot, Folyamatok, Jogok, Exportálás, and Több.
- General Settings:** Shows 'MySQL kapcsolat illesztése' set to 'utf8_general_ci'.
- Appearance Settings:** Shows 'Nyelv - Language' set to 'Magyar - Hungarian', 'Téma / Séma' set to 'pmahomme', and 'Betűméret' set to '82%'. A 'More settings' link is also present.
- MySQL Information:**
 - Szerver: Localhost via UNIX socket
 - Szerver verzió: 5.1.54-1ubuntu4
 - Protokoll verzió: 10
 - Felhasználó: root@localhost
 - MySQL karakterkészlet: UTF-8 Unicode (utf8)
- Webszerver Information:**
 - Apache/2.2.17 (Ubuntu)
 - MySQL kliens verzió: 5.1.54
 - PHP-kiterjesztés: mysql
- phpMyAdmin Information:**
 - Verziószám: 3.4.2, utolsó stabil verzió: 3.4.2
 - [Dokumentáció](#)
 - [Wiki](#)
 - [Hivatalos honlap](#)
 - [Contribute](#)
 - [Get support](#)
 - [List of changes](#)

XAMPP

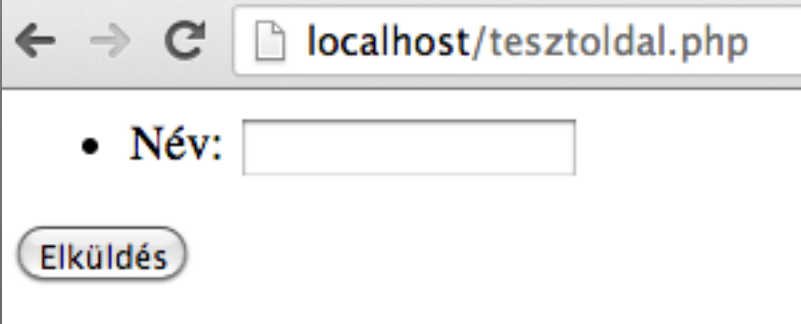
- ▶ PHP+MySQL alapú oldalak fejlesztéséhez kiváló rendszer
- ▶ platformfüggetlen, Windows, Linux, Mac OS X alatt egyaránt használható
- ▶ tartalma:
 - ▶ Apache HTTP szerver
 - ▶ MySQL, phpMyAdmin
 - ▶ PHP interpreter
 - ▶ (Perl támogatás)
 - ▶ Könnyen használható vezérlőpult
- ▶ Telepítés és elindítás után a böngészőben a <http://localhost> címen elérhető lesz az {xampp}/htdocs könyvtár
- ▶ A nyitólapot ebbe a könyvtárba helyezzük el index.php néven



Otthon telepíteni! [4]

WEBES ŪRLAP LÉTREHOZÁSA

A feladat létrehozni egy olyan űrlapot, amelyen található egy szövegbeviteli mező a felhasználó nevének bekérésére, és egy “Küldés” feliratú gomb. A bevitt adatot juttassuk el a szervernek, és ha a név “Cser Lajos”, jelenítsünk meg egy oldalt “Demonstrátor” szöveggel, egyébként pedig “Szia XY” legyen a válasz!



The screenshot shows a web browser window with the address bar displaying "localhost/tesztoldal.php". The page content includes a form with a label "• Név:" followed by a text input field. Below the input field is a button labeled "Elküldés".

TESZTOLDAL.PHP

```
1 <html>
2 <head>
3     <title>Tesztoldal</title>
4 </head>
5 <body>
6 <form action="udvozol.php" method="post">
7 <ul>
8 <li> Név: <input type="text" name="nev" /> </li>
9 </ul>
10 <input type="submit" />
11 </form>
12 </body>
13 </html>
```

POST HTTP eljárással az udvozol.php-nak elküldjük az űrlapon bevitt adatokat.

Számos további vezérlő létrehozható (pl. rádiógombok, jelölőnégyzetek stb.)

UDVOZOL.PHP

```
1 <html>
2 <head>
3     <title>Masik oldal</title>
4 </head>
5 <body>
6 <h1>Ez az erkezesi oldal</h1>
7 <?php
8 if ($_POST["nev"] == "Cser Lajos")
9 {
10     echo "Demonstrator";
11 }
12 else
13 {
14     echo "Szia " . $_POST["nev"];
15 }
16
17 ?>
18 </body>
19 </html>
```

Ezen az oldalon már PHP kód is van. Az űrlap adatait a beépített `$_POST` tömb tartalmazza.

FELADAT

Készítsünk PHP nyelven mini angol-magyar szótárat: a felhasználó megadhat egy szót, és ha az benne van a szótárunkban, kiírjuk a magyar megfelelőjét, ha nincs benne, a “nincs találat” szöveg jelenjen meg!

```
1 <html>
2 <head>
3   <title>Szotar</title>
4 </head>
5 <body>
6 <form action="fordit.php" method="get">
7 <ul>
8 <li> Enter a word: <input type="text" name="word" /> </li>
9 </ul>
10 <input type="submit" value="Find"/>
11 </form>
12 </body>
13 </html>
```

szotar.php

FELADAT / FORDIT.PHP

```
1 <html>
2 <head>
3   <title>Szotar</title>
4 </head>
5 <body>
6 <h1>Találatok</h1>
7 <?php
8 $szotar = array("one" => "egy", "two" => "kettő",
9               "dog" => "kutya", "apple" => "alma");
10 $keresett_szo = $_GET["word"];
11 if (isset($szotar[$keresett_szo]))
12     $talalt_szo = $szotar[$keresett_szo];
```

```
13     if (isset($talalt_szo))
14     {
15         $valasz = $keresett_szo . ": " . $talalt_szo;
16         echo $valasz;
17     }
18     else
19     {
20         $valasz = "Nincs találat";
21         echo $valasz;
22     }
23
24 ?>
25 </body>
26 </html>
```

KAPCSOLÓDÁS MYSQL-HEZ PHP-BAN (EGY EGYSZERŰ MEGKÖZELÍTÉS)

1. Kapcsolat nyitása

```
$kapcsolat = mysql_connect("localhost", $user, $pwd);
```

2. Ellenőrizzük, van-e kapcsolat

```
if (!$kapcsolat)
{
    die ("Nem lehet csatlakozni a MySQL kiszolgálóhoz!");
}
```

3. Ellenőrizzük, ki lehet-e választani a szükséges adatbázist, ha nem, hozzuk létre azt

```
if (!(mysql_select_db($adatbazis, $kapcsolat)) )
{
    mysql_query("CREATE DATABASE ...");
    mysql_select_db($adatbazis, $kapcsolat);
}
```

4. Adjunk meg SQL nyelven megfogalmazott utasításokat

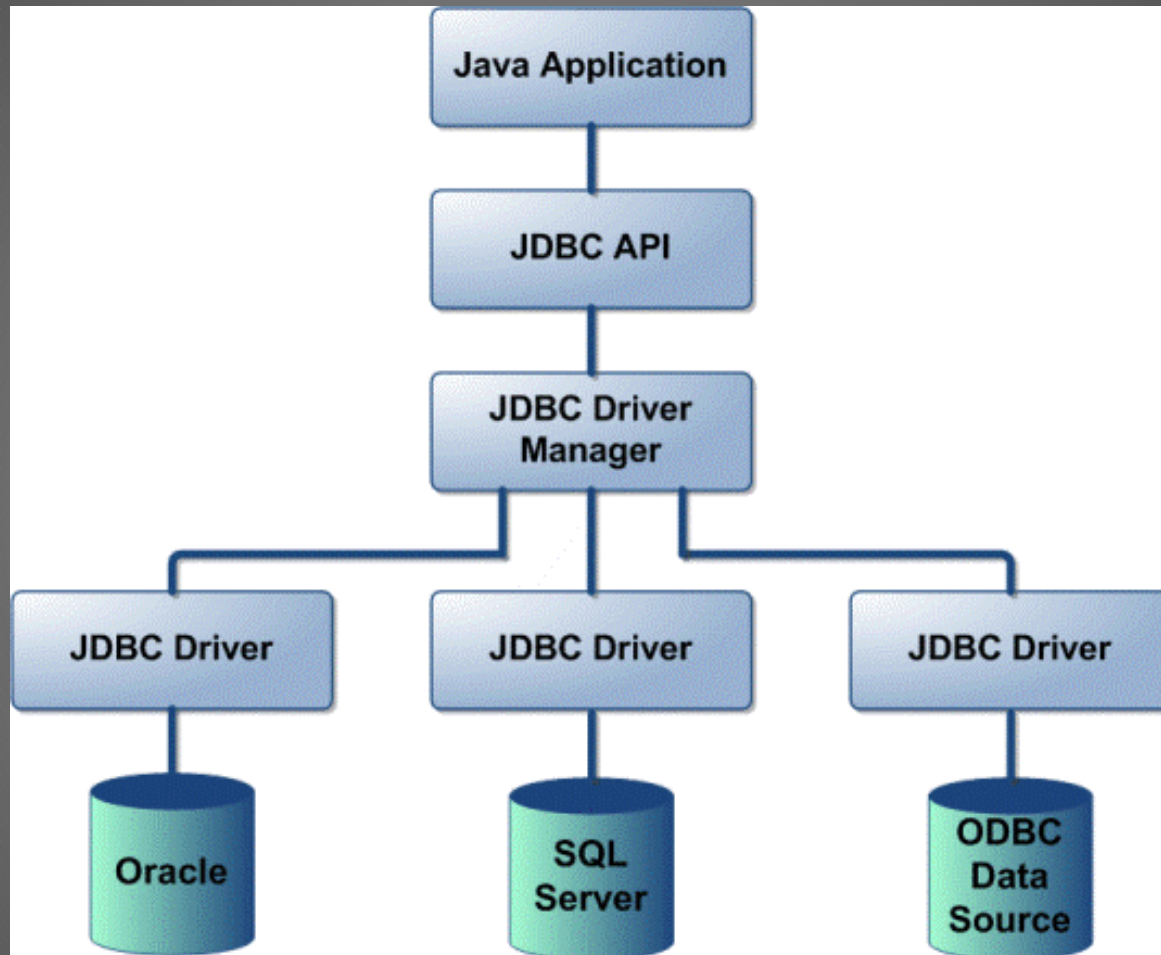
```
mysql_query($utasitas, $kapcsolat);
```

! A gyakorlati életben nem szoktuk ezeket a függvényeket alkalmazásaink kifejlesztéséhez használni, helyette számos API, keretrendszer áll rendelkezésre, amivel hatékonyabb és biztonságosabb eredményt kaphatunk. Jelen kurzus keretein belül viszont megelégszünk ezzel a megoldással is.

JDBC (JAVA DATABASE CONNECTIVITY)

- ▶ alkalmazásprogramozási interfész (API) Java nyelven készülő programokhoz
- ▶ java.sql csomag tartalmazza a vonatkozó osztályokat – ezt importálni kell a programunkban
- ▶ Működési elv:
 - ▶ a programunkban a JDBC-nek adunk át utasításokat
 - ▶ A DriverManager nevű összetevő kezeli az egyes típusú adatbázisokhoz való kapcsolódást biztosító illesztőprogramokat
 - ▶ A kapcsolat létrehozása előtt a megfelelő drivert regisztráljuk
 - ▶ A kérésünk eljut a megfelelő adatbázishoz, és végrehajtódik

JDBC



JDBC / KAPCSOLÓDÁS MYSQL-HEZ

```
import java.sql.*;
public class Jdbc {
    public static void main(String args[]){
        try {
            Statement stmt;
            ResultSet rs;
            Class.forName("com.mysql.jdbc.Driver"); // driver regisztralasa MySQL-hez.
            String url = "jdbc:mysql://localhost:3306/test";
            Connection con = DriverManager.getConnection(url, "usr", "pass");
        }
    }
}
```

utasítás

eredmények tárolására

a Java-nak alapjában véve **nem** része a MySQL driver, azt először magunknak kell beszerezni [1]

JDBC / LEKÉRDEZÉS

```
stmt = con.createStatement();  
stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,  
    ResultSet.CONCUR_READ_ONLY);  
rs = stmt.executeQuery("SELECT * from proba");
```

```
// irassuk ki az eredményeket  
System.out.println("Display all results:");  
while(rs.next()){  
    String id = rs.getString("id");  
    String vnev = rs.getString("Vezeteknev");  
    String knev = rs.getString("Keresztnev");  
    System.out.println(id + " | " + vnev + " | "+ knev);  
}
```

egy `ExecuteQuery()` függvényhívással adjuk meg, mi legyen az utasítás, a végén nincs pontosvessző. (Eredményt visszaadó utasításokhoz)

addig igaz, amíg van fel nem dolgozott eredmény a válaszban, és ilyenkor egyből a következőre is lép

JDBC / KAPCSOLAT LEZÁRÁSA

```
// zárjuk a kapcsolatot
```

```
    con.close();
```

```
    // hiba eseten irassuk ki a stack-et (az egész program egy try blokkban  
    // volt)
```

```
    }catch( Exception e ) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    }
```

```
}
```


ODBC (OPEN DATABASE CONNECTIVITY)

- ▶ A JDBC-hez hasonlóan számos adatbáziskezelővel képes kapcsolatot létesíteni, de nem Java környezetből, hanem C-ből.
- ▶ A megfelelő header fájlok include-olása után az alábbi adatstruktúrákkal dolgozhatunk:
 - ▶ Környezet (Environment): a kliens hozza létre a DBMS-sel való kapcsolat előkészítéséhez.
 - ▶ Kapcsolat (Connection): DBMS-sel való kapcsolat leírására szolgál. Egy környezethez több kapcsolat tartozhat.
 - ▶ ODBC-utasítás (Statement). Egy SQL utasítás leírására szolgál. Minden ODBC-utasítás valamely kapcsolathoz tartozik. Ugyanaz az ODBC-utasítás különböző időpontokban különböző SQL-utasításokat tartalmazhat.
- ▶ A fentiek kezelése handle-k (az adatstruktúrára mutató pointererek) segítségével történik.
- ▶ Ezek típusai sorrendben SQLHENV, SQLHDBC, SQLHSTMT.

MEGJEGYZÉSEK

- ▶ [1] A MySQL-hez szükséges JDBC driver beszerezhető itt:
<http://dev.mysql.com/downloads/connector/j/>
- ▶ [2] Németh Gábor példái nyomán
- ▶ [3] Forrás: dr. Katona Endre: Adatbázisok
<http://www.inf.u-szeged.hu/~katona/db-ea I .pdf>
- ▶ [4] Az XAMPP letölthető a következő címről:
<http://www.apachefriends.org/en/xampp.html>