

Dr. Pál László, Sapientia EMTE, Csíkszereda

WEB PROGRAMOZÁS

4.ELŐADÁS

2015-2016

Adatbázis alapú alkalmazások készítése PHP-ben

Adatbázis alapú alkalmazás

2

- A leggyakrabban használt dinamikus alkalmazások adatbázis alapúak
- Ezért fontos ismerni ezeknek az adatbázisoknak a létrehozási lehetőségeit, módosítását (beszúrás, módosítás, törlés, stb.) használva a megfelelő PHP utasításokat

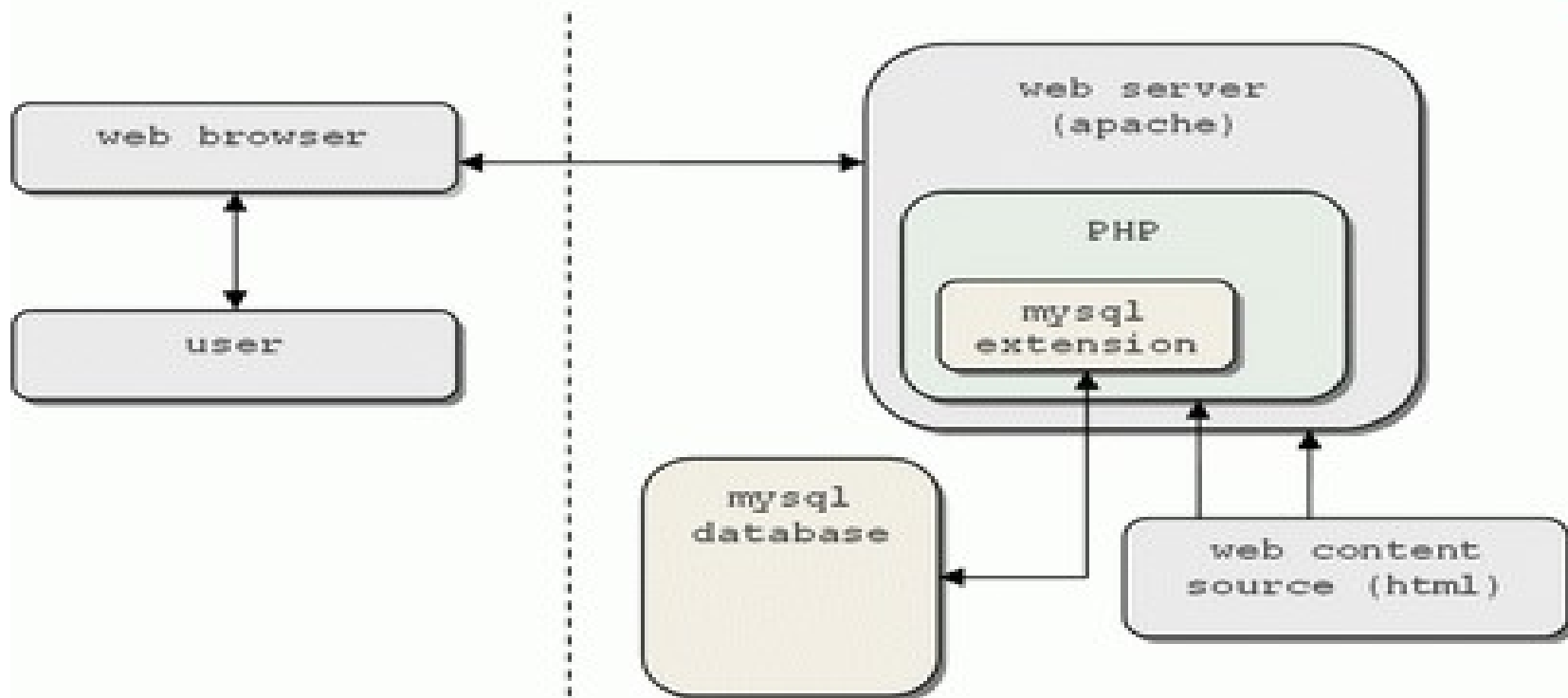
Adatbázis típusok

3

- Közvetlen hozzáférésű (fájlalapú)
 - ▣ Az adatbázis utasítások közvetlenül az adatokat tároló fájlokhoz fér hozzá
 - ▣ Példa: SQLite
- Közvetett hozzáférésű (szerveralapú)
 - ▣ Az adatokat tároló fájlokat egy adatbázis-szerver kezeli, minden más kód ezen a szerveren keresztül kezeli az adatokat
 - ▣ Példa: MySQL, PostgreSQL, Oracle, stb.

Adatbázis alapú alkalmazás működése

4



5

Adatbázisok felépítése

Ismétlés

Adatbázis szerkezete

6

- Adatbázis séma:
 - ▣ Meghatározza az adatbázisban tárolt adatok struktúráját és a közöttük lévő kapcsolatokat
 - ▣ A relációs adatbázisokban az adatokat táblákban tároljuk, míg a kapcsolatokat kulcsokkal hozzuk létre
- Adatbázis adatok:
 - ▣ Mező: az adattábla egy oszlopának felel meg
 - ▣ Rekord: az adattábla egy sora
- Adatbázis: az adattáblák együttese

Adatbázis szerkezete

7

□ Példa:

Database

Table 1

Record 1	Field 1	Field 2	Field 3
Record 2	Field 1	Field 2	Field 3
Record 3	Field 1	Field 2	Field 3
Record 4	Field 1	Field 2	Field 3

Table 2

Record 1	Field 1	Field 2	Field 3
Record 2	Field 1	Field 2	Field 3
Record 3	Field 1	Field 2	Field 3
Record 4	Field 1	Field 2	Field 3

Adatbázis táblák

8

- A táblák oszlopai meghatározzák a tábla soraiban levő értékek típusát, méretét és egyéb beállításokat
- Példa:

The diagram illustrates database concepts using a sample table titled "Személyek". The table has four columns: "Név", "SzemélySzám", "SzületésiIdő", and "Foglalkozás". The rows represent individual records. Callouts identify the table as a "TÁBLA (egyed típus)", the columns as "MEZŐ (tulajdonságtípus)", a specific row as a "REKORD (egyed)", and a specific cell value as a "MEZŐÉRTÉK (tulajdonság)".

Személyek			
Név	SzemélySzám	SzületésiIdő	Foglalkozás
Virág Borbála	101558 AI	1970.02.10	Tanár
Tóth Áron	234511 ZK	1969.05.25	Vállalkozó
Tóth Áron	113297 UL	1960.09.01	Fodrász
Árva Viola	988326 AX	1967.09.18	Asszisztens
Bende László	656456 ZY	1970.02.10	Manager

A MySQL főbb adattípusai

9

□ Egész típusok

- TINYINT: 1 bájtos szám (-128-tól 127-ig, vagy 0-tól 255-ig, ha előjel nélküli)
- SMALLINT: 2 bájtos szám
- MEDIUMINT: 3 bájtos szám
- INT: 4 bájtos szám
- BIGINT: 8 bájtos szám

A MySQL főbb adattípusai

10

- Valós típusok
 - ▣ DECIMAL: fixpontos valós szám
 - ▣ DOUBLE: dupla pontosságú lebegőpontos valós szám
 - ▣ FLOAT: egyszeres pontosságú lebegőpontos szám

A MySQL főbb adattípusai

11

□ Valós típusok

- ▣ DECIMAL: fixpontos valós szám (Pontossága 53 tizedes jegy)
- ▣ DOUBLE: dupla pontosságú lebegőpontos valós szám (Pontossága 53 tizedes jegy)
- ▣ FLOAT: egyszeres pontosságú lebegőpontos szám (Pontossága 24 tizedes jegy)

A MySQL főbb adattípusai

12

- Szöveges típusok
 - ▣ VARCHAR(N): változó hosszúságú karakterlánc, amely 0-65535 karaktert tartalmazhat
 - ▣ TEXT: Max. 65,535 karakter hosszúságú szöveg

A MySQL főbb adattípusai

13

- Dátum és idő
 - ▣ DATE: csak dátum
 - ▣ TIME: csak idő
 - ▣ DATETIME: dátum és idő

Egyéb tulajdonságok a típusokkal kapcsolatban

14

- Bármelyik típusú oszlop értéke lehet **NULL**, kivéve, ha a sémában a **NOT NULL** paramétert megadtuk
- Az egész és a tört számoknál az **UNSIGNED** paraméterrel kizárhatjuk a negatív értékeket
- Az egész számoknál az **AUTO INCREMENT** paraméterrel automatikusan **sorszámozhatunk**
- A **TEXT** típust kivéve bármely oszlopnak lehet **alapértelmezett értéke**

Elsődleges kulcs

15

- Az **elsődleges kulcs** a rekordok **egyértelmű beazonosítására** szolgáló oszlop a táblában
- Minden értékének **különbözőnek** kell lennie
- Az **elsődleges kulcs** akár **több oszlopból** is állhat
 - ▣ Ekkor az **értékek kombinációinak** kell eltérniük
- Példa:

Customer ID	Forename	Surname
1	Simon	Jones
2	Emma	Price
3	Laura	Jones
4	Jonathan	Hale
5	Emma	Smith

Simple primary key

Primary Key

ProductID	VendorID	AverageLeadTime	StandardPrice	LastReceiptCost
1	1	17	47.8700	50.2635
2	104	19	39.9200	41.9160
7	4	17	54.3100	57.0255
609	7	17	25.7700	27.0585
609	100	19	28.1700	29.5785

ProductVendor table

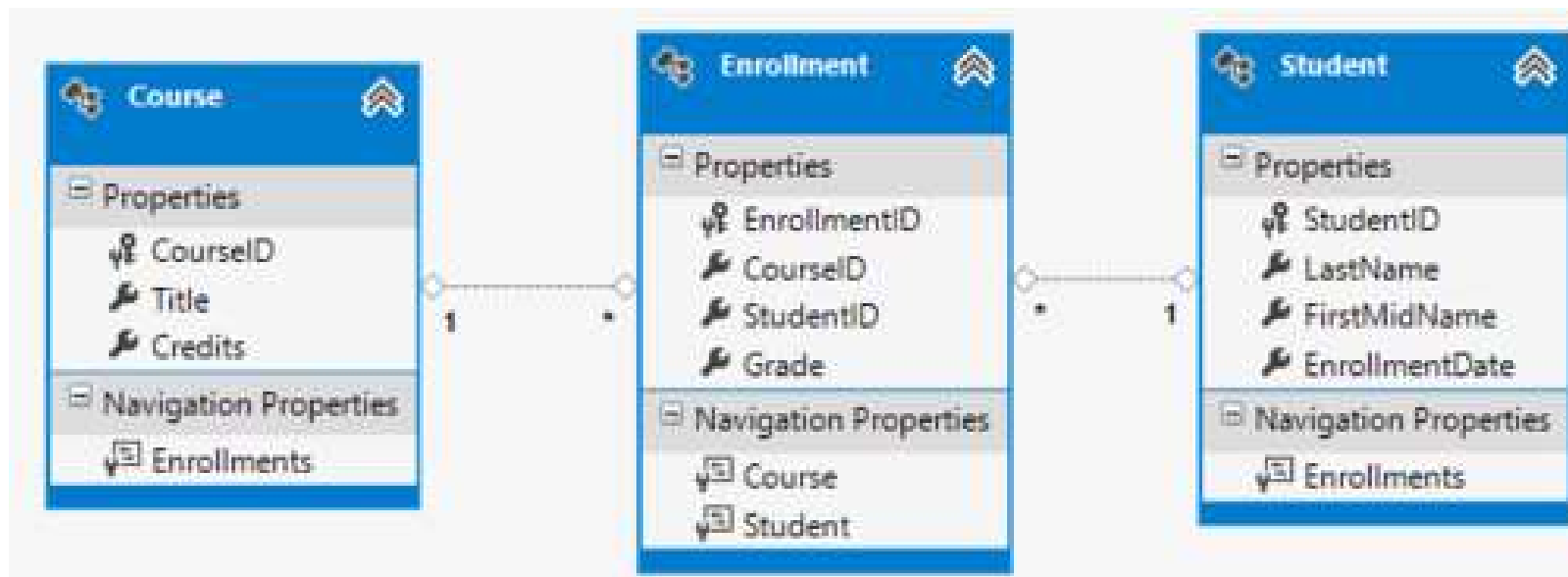
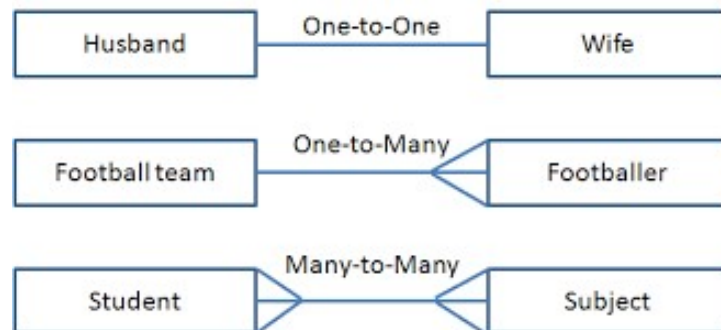
Táblák közötti kapcsolatok

16

- Különböző táblák között úgy tudunk kapcsolatot teremteni, hogy használjuk az egyik tábla rekordjainak elsődleges kulcsát a másik táblában
- Háromféle kapcsolatot használhatunk:
 - ▣ 1:1 – az egyik tábla egy rekordjához a másik táblában **nulla vagy egy** rekord tartozik
 - ▣ 1:N – az egyik tábla egy rekordjához a másik táblában **nulla vagy egy** rekord tartozik
 - ▣ N:M – **két 1:N kapcsolat** kombinációja

Példák kapcsolatokra

17



Adatbázis indexek

18

- A tábla mezőinek felvétele és tulajdonságainak meghatározása után célszerű beállítanunk az egyes mezők indexelését
- Index létrehozásával meggyorsíthatjuk az adatbázis rendezését, illetve az adatbázisban való keresést
- Azokra az oszlopokra, amelyek értékeiben gyakran keresünk, érdemes **indexet** tenni
- Az index hátránya: lassabb adatmódosítás, az adatbázis több helyet foglal az indexek miatt

Az SQL nyelv

19

- Az SQL (Structured Query Language) relációs adatbázis-kezelők lekérdezési nyelve
- Segítségével a különböző típusú adatbázisokat azonos módon kezelhetjük
- Adatdefiníciós és adatkezelési utasításokkal rendelkezik

Az SQL nyelv

20

- Fontosabb utasítások:

- Új tábla létrehozása:

`CREATE TABLE személyek (keresztnev VARCHAR(30),
vezeteknev VARCHAR(30), kor INT);`

- Adatok szelektálása (SELECT):

`SELECT * FROM személyek;` - az összes mezőt visszaadja

`SELECT kor, keresztnev FROM személyek;` - csak két mezőt ad vissza

`SELECT * FROM személyek WHERE keresztnev = 'Gábor';`

feltételes lekérdezés

Az SQL nyelv

21

□ Fontosabb utasítások:

□ Új sor beszúrása táblázatba:

```
INSERT INTO személyek ( keresztnév, vezeteknev, kor )  
VALUES ( 'János', 'Kovács', 36 );
```

□ Létező bejegyzés módosítása:

```
UPDATE személyek SET keresztnév = "Gábor" WHERE vezeteknev =  
"Szakács";
```

Csak azokban a sorokban változtatja "Gábor"-ra a *keresztnév* mező értékét, ahol a *vezeteknev* mező a "Szakács" karakterláncot tartalmazza.

A MySQL adatbázis szerver



22

- Több-felhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver
- Ingyen használható
- Adminisztráció:
 - ▣ Parancssorból
 - ▣ Grafikus felületek (MySQL Administrator és MySQL Query Browser)
 - ▣ phpMyAdmin (az XAMPP-n belül)
(<http://localhost/phpmyadmin/>)

Adatbázis műveletek PHP-ben

23

- Általában a következő műveleteket hajtjuk végre egy adatbázissal kapcsolatosan:
 - Kapcsolódás a MySQL serverhez
 - Az adatbázis kiválasztása
 - Műveletek az adatbázisban
 - Tábla kiválasztása
 - Adatok felvétele, törlése, frissítése, stb.
 - Adatbázis-kapcsolat bezárása

Csatlakozás a MySQL szerverhez

24

- Mielőtt elkezdhetnénk dolgozni az adatbázissal, csatlakoznunk kell a kiszolgálóhoz. A PHP-ben erre a *mysql_connect()* függvény szolgál (procedurális).

- Szintaxis:

```
mysql_connect(servername, username, password);
```

- servername: a kiszolgáló neve
- username: felhasználó
- password: felhasználó jelszava

- Példa:

```
$kapcsolat = mysql_connect( "localhost", "root", "jelszo" );  
if ( ! $kapcsolat )  
    die( "Nem lehet csatlakozni a MySQL kiszolgálóhoz!" );
```


Csatlakozás a MySQL szerverhez

25

□ Más kapcsolódási módok:

▣ OOP alapú

```
$conn = new mysqli($servername, $username, $password);  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
echo "Connected successfully";
```

▣ PDO (*PHP Data Objects*)

```
try {  
    $conn = new PDO("mysql:host=$servername;dbname", $username, $password);  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    echo "Connected successfully";  
} catch (PDOException $e) {  
    echo "Connection failed: " . $e->getMessage();  
}
```

Adatbázis kiválasztása

26

- Miután létrehoztuk a kapcsolatot (az egyszerűség kedvéért **\$kapcsolat**-nak fogjuk hívni), ki kell választanunk az adatbázist, amiben dolgozni szeretnénk:

```
mysql_select_db("adatbazis", $kapcsolat);
```

- A fenti sorban megadtuk, hogy az „adatbazis” nevű adatbázist szeretnénk használni, mégpedig a \$kapcsolat nevű kapcsolaton keresztül.

Adatbázis kiválasztása

27

- Más módszerek:

- ▣ OOP változat:

```
$conn = new mysqli('localhost', 'my_user', 'my_password', 'my_db');
```

```
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

Adatbázis-kapcsolat bezárása

28

- Ha már nincs szükségünk az adatbázis-kapcsolatra, le kell zárunk azt (általában a PHP kód végén, az utolsó sorban):

```
mysql_close($kapcsolat);
```

vagy

```
$conn->close();
```

Adatbázis nyitás-zárás példa

29

- A „db_mysql” nevű adatbázis megnyitása

```
$felhasznalo = "root";
$jelszo = "";
$adatbazis = "db_mysql";
if ($conn = @mysql_connect("localhost", $felhasznalo, $jelszo)) {
    print '<p>Sikeres kapcsolódás a MySQL kiszolgálóhoz!</p>';
    if (@mysql_select_db($adatbazis,$conn)) {
        print '<p>Sikeresen kiválasztott adatbázis.</p>';
    } else {
        print '<p>Nem lehet kapcsolódni az adatbazishoz</p>';
    }
    mysql_close($conn);
} else {
    print '<p>Nem lehet kapcsolódni a MySQL kiszolgálóhoz!</p>';
}
```

Adatbázis nyitás-zárás példa

30

- A „db_mysql” nevű adatbázis megnyitása

```
$conn = new mysqli("localhost", "root", "", "db_mysql");  
  
if ($conn->connect_error)  
    die("Connection failed: " . $conn->connect_error);  
else  
    print "Sikeres kapcsolódás."  
  
$conn->close();
```

- Nem létező adatbázis esetén:

Warning: mysqli::mysqli(): (HY000/1049): Unknown database 'db_mysql' in
Connection failed: Unknown database 'db_mysql'

SQL parancsok futtatása PHP-ből

31

- Miután elkészült az adatbázis-kapcsolat, és kiválasztottuk a megfelelő adatbázist, a létrehozott kapcsolaton SQL parancsokat futtathatunk.
- Először egy szöveges változóban eltároljuk a parancsot:
Pld. `$sql = "SELECT * FROM tablanev";`
- Majd a parancsot lefuttatjuk a megadott kapcsolaton:
`mysql_query($sql, $conn);`

Vagy

`$conn->query($sql)`

- A visszatérített érték pld. SELECT esetén egy eredmény halmaz sikeres futás esetén, különben FALSE érték
- INSERT, UPDATE, DELETE, DROP esetén TRUE siker esetén, különben FALSE

Adatbázis létrehozása – 1.Példa

32

□ Procedurális változat:

```
// Create connection
$conn = mysqli_connect($servername, $username, $password);|
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
```


Adatbázis létrehozása – 2. Példa

33

OOP változat:

```
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
```

Tábla létrehozása – 1.Példa

34

□ OOP változat

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}
```

Tábla létrehozása – 2.Példa

35

□ Procedurális változat

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

mysqli_close($conn);
```

Adat beszúrás – 1.Példa

36

□ OOP változat:

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Adat beszúrás – 2.Példa

37

□ Procedurális változat:

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
```

Adatok lekérdezése - SELECT

38

- Ahhoz, hogy a lekérdezés eredményét - az úgynevezett eredménytáblát - fel tudjuk dolgozni, először el kell tárolnunk egy változóban:

```
$eredmeny = mysql_query($parancs);
```

- Az eredménytábla sorainak száma:

```
mysql_num_rows($eredmeny)
```

Ha a fenti függvény visszatérési értéke 0 (nulla), akkor az eredménytábla üres, tehát nem találtunk a keresési feltételeknek megfelelő rekordot, különben feldolgozhatjuk az eredménytáblát

Az eredménytábla feldolgozása

39

- Ha az eredménytábla nem üres, sorról sorra feldolgozhatjuk azt. Ehhez egy ciklust használunk, aminek minden iterációjában egy `$sor` nevű tömbbe olvassuk ki az aktuális rekord mezőinek nevét és értékét:

`mysql_fetch_array($eredmeny)` – visszaadja a tábla aktuális sorát tömb formájában

Példa:

```
while ($sor = mysql_fetch_array($eredmeny)) {  
    ... az aktuális rekord feldolgozása ...  
}
```

Az eredménytábla feldolgozása

40

- Az eredménytábla kiíratása soronként:

```
while ($sor = mysql_fetch_array($eredmeny)) {  
    print $sor["mező1"] . "<br>";  
    print $sor["mező2"] . "<br>";  
    print $sor["mező3"] . "<br>";  
    print $sor["mező4"] . "<br><br>";  
}
```


Az eredménytábla feldolgozása

41

□ OOP változat:

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while ($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"] . " - Name: " . $row["firstname"] .
            " " . $row["lastname"] . "<br>";
    }
} else {
    echo "0 results";
}

$conn->close();
```

Az eredménytábla feldolgozása

42

□ Procedurális változat:

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
```

Rekordok törlése – 1.Példa

43

□ OOP változat

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}
```

Rekordok törlése – 1.Példa

44

□ Procedurális változat

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
```

Rekordok frissítése – 1.Példa

45

□ OOP változat

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();
```

Rekordok frissítése – 2.Példa

46

□ Procedurális változat

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
```

Adatok feltöltése - Példa

47

□ Diákok adatainak rögzítése adatbázisban:

```
$conn = new mysqli("localhost", "root", "", "db_mysql");  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
$sql = "INSERT INTO diakok (nev,szak,atlag) VALUES ('$nev','$szak','$atlag')";  
$result = $conn->query($sql);  
if ($result===TRUE) {  
    $conn->close();  
    echo "Köszönjük! Az adatokat elmentettük.<br>";  
    echo "<a href='feltolt_diakok.php'>Uj adat</a><br>";  
    echo "<a href='listaz_diakok.php'>Adatok listazasa</a><br>";  
} else  
    echo "Muveleti hiba.\n";
```

Adatok listázása - Példa

48

□ Diákok adatainak listázása:

```
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM diakok";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while ($sor = $result->fetch_assoc()) {
        print $sor["id"] . "      " . $sor["nev"] . "      " . $sor["szak"] .
            "      " . $sor["atlag"] . "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
```


Példa

49

Kérem a felhasználó nevét:

A felhasználó felvétele az adatbázisba

Az adatbázisban szereplő felhasználók:

Kiss Lajos ▲
Nagy Aladar ▼

A kiválasztott felhasználó törlése az adatbázisból

Példa

50

```
include("db_config.php");
if (isset($_POST['submit'])) {
    $user = $_POST['username'];
    if (isset($user) && ($user !== "")) {
        $sql = "INSERT INTO Users (id,UName) VALUES ('','$user')";
        $result = $conn->query($sql);
    }
}

if (isset($_POST['userdel']) && isset($_POST['userid'])) {
    $uid = $_POST['userid'];
    $sql = "DELETE FROM Users WHERE id=$uid";
    $result = $conn->query($sql);
}
```

```
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbconn = mysql_connect($dbhost, $dbuser, $dbpass);
$dbname = "db_mysql";
// Create connection
$conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Példa

51

```
if (isset($_POST['userdel']) && isset($_POST['userid'])) {
    $uid = $_POST['userid'];
    $sql = "DELETE FROM Users WHERE id=$uid";
    $result = $conn->query($sql);
}
print "<h3>Az adatbázisban szereplő felhasználók:</h3>";

print "<form name=\"\" action=\"\" method=\"POST\">\n";
print "<select name=\"userid\" size=\"11\">\n";

$redm = $conn->query("SELECT * FROM Users");
while ($sor = $redm->fetch_assoc()) {
    $id = $sor["id"];
    $name = $sor["UName"];
    print "<option value=\"$id\" />$name\n";
}
print "</select><br /><br />";
print "<input type=\"submit\" name=\"userdel\" value=\"A kiválasztott\"";
print "</form>";

$conn->close();
```

Könyvészet

52

1. <http://www.w3schools.com/>
2. <http://bcecid.net/author/solymosimate/>