

Adatbázis rendszerek I

2011.11.17

Mai témák:

DB-mintapélda

mySQL

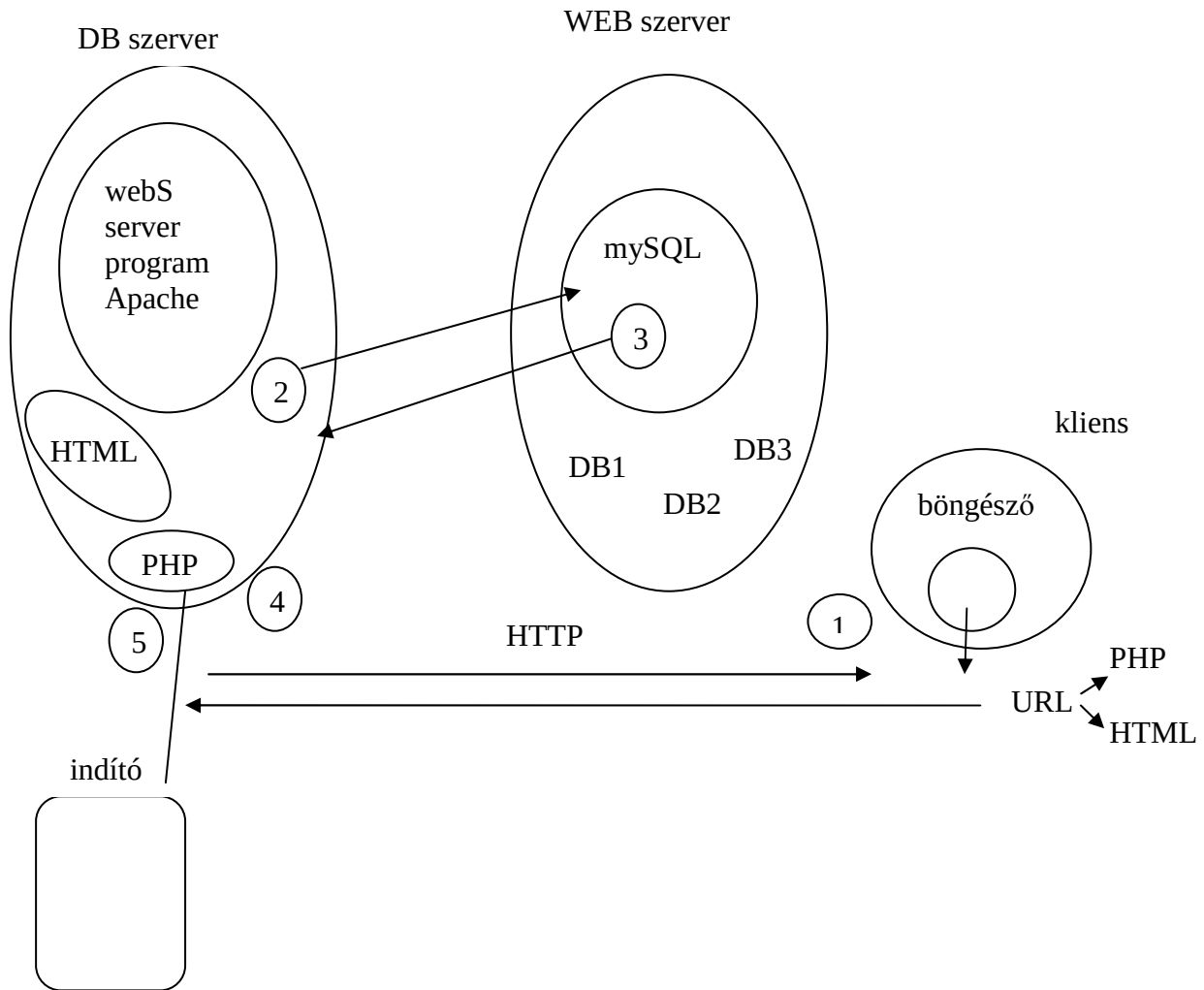
Versenyzők nyilvántartása

az n-edik versenyző nevének ki listázása

kezelés + DB kapcsolat

DBMS telepítés

architektúra tervek



Hogyan telepíthetünk mySQL-t?

1. Telepítő letöltése, futtatás

2. Paraméterezés

Adatkezelés jellege:

- transactional database only
- multifunctional database

A tranzakció kezelés plusz erőforrást igényel, akkor kell tranzakció kezelés ha módosítás orientált.

mySQL-ben a tranzakció kezelési mód IDODB-ben valósítható meg.

Az IDODB garantálja, hogy biztosítja a teljes integritás őrzést.

MSISAM - régebbi tárolási mód, gyorsabb

3. DBMS processzek kezelése

4. telepítés helye

a műveletek jellege

OLTP tranzakció orientált rendszer - sok kicsi parancs módosítás jelleggel, pl
helyjegyfoglalás

OLAP adatelemzés

5. karakterkészlet kezelése

utólag nem módosítható

Két dolgot kell megadni:

1. karakterkészletet
2. rendezési sorrendet

6. szerver elérési címe

7. DBA jogosultsági beállítása

Megjegyzés: Ha a root jelszó elfelejtődne, akkor rendszerszintű felhasználóként az op-rendszer rendszergazdája is átveheti általában.

A telepítés után egy példány jön létre.

A működő példány 2 komponensből áll:

- DBMS
- DB

2 leggyakoribb mód van:

1. grafikus menüből
2. parancssorból:
 - **net start**
 - **net stop**

Állományok

- programállományok
- paraméterállományok
- adatbázis állományok

Egy adatbázis több fájlban osztozik szét.

CREATE DATABASE név;

Minden alkalmazáshoz célszerű új adatbázist létrehozni

SHOW DATABASES - megjelenítés

USE név - kiválasztás

SELECT DATABASE() - lekérdezés

Felhasználó account létrehozása:

CREATE USER név **IDENTIFIED BY** 'név'

jogot kell adni neki:

GRANT ALL ON db.obj **TO** feltétel

Paraméterezés és adatvesztés elleni védelem

A legfontosabb: mentés **BACKUP**

Van egyszerű mód: **MYSQL DUMP**

DATADUMP funkció

1. megadjuk melyik adatbázist
2. megadjuk hová mentse

A *DUMP* szöveges formátumú SQL parancsokat tartalmazó mentés.

A *DUMP* helyetti *BACKUP* rutinok binárisan tárolódnak.

Tárolt eljárás (függvény)

VERSENYZOK [kod int PRIMARY KEY, név char], pont int

Kötelező-e függvénnyel megvalósítani ?

nem kötelező

Megjegyzés: ez lehetne out típusú eljárással is megvalósítani

Megjegyzés: A CURSOR lényege - az eredményt egy ideiglenes táblába leteszi és onnan rekordonként olvassa

delimiter //

CREATE FUNCTION NESIK (POZ INT) RETURN INT

BEGIN

**DECLARE C1 CURSOR FOR SELECT NEV, PONT FROM VERSENYZOK ORDER BY
PONT DESC;**

DECLARE CONTINUE HANDLER FOR NOT FOUND SET VEGE=1;

OPEN C1;

SET VÉGE=0;

SET I=1;

WHILE VÉGE=0 DO

FETCH C1 INTO SNÉV, SPONT

IF POZ=1 TNÉV

SET EREDMÉNY=SNÉV

SET VÉGE=1

END IF

SET I=I+1

END WHILE

CLOSE C1

RETURN EREDMÉNY

END

//delimiter

Megjegyzés a függvénnyel kapcsolatban

"

Logikája - Adatbázisból rendezve kérdezzük le a rekordokat majd az n-ediket kiemeljük a listából

" **DECLARE CONTINUE HANDLER FOR NOT FOUND SET VEGE=1;**" - jelentés:

ha NOT FOUND hiba lép fel, akkor folytasd a munkát és még tedd 1-re a változót

1. Az eredményt ciklusban, rekordonként vesszük át
2. Leállítási feltétel
3. a hibajelzésen keresztül megy -> NOT FOUND
4. a kliensnél jelezni kell, hogy ne álljon le hibákra
5. a hiba kezelésére szolgál a HANDLER mechanizmus

"

Indító lap (HTML)

```
<HTML>
  <BODY>
    <H2>ADATFELVITEL</H2>
    <FORM ACTION="valasz.php" METHOD="GET">
      POZÍCIÓ: <INPUT TYPE="TEXT" NAME="b1"> <br>
        <INPUT TYPE="SUBMIT" VALUE="TOVÁBB">

    </FORM>
  </BODY>
</HTML>
```

PHP program

```
<HTML>
  <BODY>
    <H2>Válasz</H2>
    <?PHP
      $v = $_GET["sorszám"]
    ?>
  </BODY>
</HTML>
```

Saját megjegyzés:

Szerintem nem írtam le a teljes php programot, nem értek a php-hoz, nem tudom hogy kell lennie

SQL parancsok kiadása a programból

Kell hozzá:

SQL API (PHP)

Lépések:

1. kapcsolat felvétel
\$l = mysql_connect ('localhost', 'nev', 'jelszo')
2. parancsküldés
\$l mysql_query ("sql_parancs", \$l)
3. feldolgozás (CURSOR)
\$a=mysql_fetch_row(\$r)
\$a[1] -> mező értéke
4. lezárás
mysql_close (\$l)