PHP-MYSQL

ALAPOK

- belépés saját adatbázisba (phpMyadmin) és könyvtárba (winscp)
- php.zip letöltése
- php alapok**: ciklus.php**
- **ÖNÁLLÓ FELADAT**: **rendez.php**
- biblio script futtatása, külső kulcsok helyreállítása phpmyadmin segítségével
- titles tábla tartalmának listázása php-ból: **listaz.php**
- **ÖNÁLLÓ FELADAT:** ez alapján kilistázni a member táblát
- ürlapok: **param.php, cel.php,**
- **ÖNÁLLÓ FELADAT**: kombinálni a kettőt: **param_cel.php**.
- A POST és a GET közti különbség: **form.php**
- **ÖNÁLLÓ FELADAT**: írjunk olyan php lapot, amely a beadott címre illeszkedő könyveket listázza! (kombinálni a listaz.php-t és a param.php-t), megoldás: **kereso.php**
- **futtat.php**: a foreach bemutatása
- *törzsadat-karbantartás legördülő listával:*
    - **publishers.sql** futtatása (megcsinálja a publishers táblát)
    - feltesszük, hogy a könyv azonosítása már egy korábbi böngésző űrlapon megtörtént, megvan az ISBN, és szeretnénk a címet, kiadót editálni
    - minta: **konyvadat.php**, combo.php
- **ÖNÁLLÓ FELADAT**: ehhez hasonló: új tábla készítése kézzel: membertype (type_id tinyint, type_name varchar), szintén InnoDB motorral, legalább két felhasználótípus (pl. normál, törzstag stb.) felvitele kézzel. A type_id mezőhöz hozzunk létre primary indexet, member.membertype mezőhöz is készítsünk indexet (tábla -> szerkezet -> reláció nézet), majd kössük külső kulcs kényszerrel össze a két mezőt.
  A feladat a tag nevének és típusának editálása egy php lapon. Megoldás: **tagadat.php** a biblio_megold könyvtárban.

Opcionális, ha van rá idő

KÖNYVTÁRI RENDSZER

- komolyabb rendszer: a könyvtári kölcsönzés demo, forrás elemzése
    - adatbázis-séma a weblap alapján, session-kezelés
    - index.php: hidden input
    - **tag.php**: 2 könyv kölcsönzése
    - tag.php listázás megértéséhez a combo.php alapján a checkbox működése (name és value különbsége): a name csak akkor szerepel a _POST tömbben, ha be lett x-elve trükk: a name=kolcs[$i] névvel a _POST-ban többdimenziós tömböt hozunk létre
    - tag.php: join szintaxisa JOIN kulcsszó nélkül (megy a JOIN is az új mysql-en)
    - tag.php: figyelmeztetés, a böngészőben látszik a hidden input (tagID)
    - tag.php: dátumhoz intervallumot hozzá lehet adni + jellel

    - **kivesz.php**: minden kölcsönzés 10-zel csökkenti az egyenleget
    - gondolunk a konkurenciára is (hátha közben más kivette a könyvet!)
- ÖNÁLLÓ FELADAT: tag.php lapon az egyenleg kiírása
- ÖNÁLLÓ FELADAT**: visszahoz.php** befejezése
    - Büntetés: ha több, mint 15 napja vitte el, akkor b = 5*(idő-15)
    - segítség a napok számához: az eltelt napok száma: datediff(now(),lending.DateStart)

MYSQL: TÁROLT ELJÁRÁS + TRANZAKCIÓVEZÉRLÉS

- **tárolt eljárás** készítése és meghívása: kézzel beírjuk a select * from member sort a listaz nevű új eljárás szövegének, és call listaz(); paranccsal végrehajtjuk a phpMyadmin-on és a **futtat.php** révén is.
- **kivesz.sql**: tárolt eljárás a kivesz.php tranzakciójára, meghívás php-ból
- ezt kiegészítjük tranzakció-kezeléssel: **kivesz_tr.sql** (Próba!)
- az InnoDB számára a default izolációs szint a repeatable read, állítható az adatbázis sql ablakában SET-tel, és megnézhető localhost Variables listában, a tx_isolation változó. (SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED; SELECT @@tx_isolation;)
- ÖNÁLLÓ FELADAT: a visszahoz tranzakció átírása a szerverre (elég büntetés nélkül), tesztelés

SQL-INJEKTÁLÁS

- SQL-injektálás elve: user tábla támadása az **inject/from.php**-val
    - *paraméterbe rejtett kifejezés*: összes rekord listázása: ‘ or 1 or ‘
    - *schema mapping*: mezőnevek kitalálása: ' and id is null and ' (ha nem kapunk szerver hibát, kitaláltuk). Ha nem találjuk el, ezt kapjuk: lekérdezési hiba: Unknown column 'userid' in 'where clause'
    - táblanevek kitalálása: ' or 1 in (select 1 from user) or ' (itt Table ’users’ doesn’t exist hibaüzenetet kapunk, ha nem találtuk el)
    - a következő húzás adatmódosítás lenne:

      select * from user where user = '' and password = '' ; update user set password = ''; --

      HA több parancs végrehajtását is engedné a Php környezet.
    - és így tovább…
- védekezés: tárolt eljárás (lásd fent) vagy prepared statement. Az elkészítés után az utasítás nem lesz már többször parszolva (nem stringként kapja meg a driver), ezért egy paraméterből nem lehet egy kifejezés vagy több utasítás. A típus ellenőrzése, tisztítása a szerver által történik. Dinamikus SQL-t nem használunk. Lásd: **form_ps.php**
- Egy tipikus támadás lépései az elfelejtett jelszó funkcióra alapozva (jelszó elküldetése a felhasználó email címére):

    - egy űrlap-mezőbe egy ’. Ha erre valami „server error” jön (pl. HTTP Error 500 Internal server error), akkor ez a bemenet támadható
    - schema mapping a hibaüzenetek alapján, lásd fent. A jelszavas tábla neve és mezői megvannak.
    - ha a login megenged adatmódosítást: új felhasználó beszúrása a user táblába (problémák: not null mezők, nincs elég hely az SQL beírására, az új felhasználónak nincsenek jogai, mert azok egy másik táblában vannak)
    - ennél jobb: egy vezető (nagy jogosultságú) felhasználó email címét megszerezni pl. a cég weblapjáról, és egy injektált UPDATE utasítással átírni ezt az email címet a users táblában a mi címünkre. Utána „elfelejtettem a jelszavamat”.
- az elkövetettt hibák
    - dinamikus SQL alkalmazása
    - elmaradt a felhasználói bemenetek tisztítása (megengedett karakterek) –nem minden mezőre megoldható, pl. dátum, email megoldható, de név, megjegyzés nehezen. Ugyanez áll az idézőjelek leszűrésére. Ettől függetlenül a mysqli_real_escape_string használandó.
    - hibaüzenet visszaírása a kliensnek (fejlesztés idején hasznos de utána veszélyes!)
    - a bejelentkező lap olyan jelszóval rendelkezzen, amellyel nem lehet az adtabázist módosítani, csak a user táblát olvasni. Sikeres bejelentkezés után új kapcsolatot nyitunk. Semmilyen alkalmazás sem használja az sa logint.
    - találati rekordok számát ellenőrizni kell
    - jelszavak helyett SHA-2 hash tárolandó

ÖNÁLLÓ FELADAT: a korábbi interaktív **kereso.php** megvalósítása prepared statement és/vagy tárolt eljárás használatával