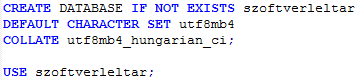
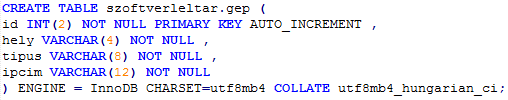
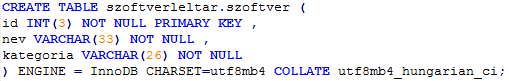
Szoftverleltár (gyakorlás) feladat megoldása

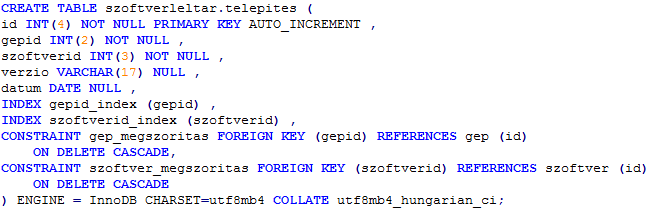
Ma már a legtöbb intézményben természetes tartozék a számítógép, majdnem minden irodában megtalálható. A telepített hardver- és szoftvereszközökről általában automatikusan generált nyilvántartás készül. Az adatbázis egy oktatással foglalkozó cég nyilvántartásának szoftverekre vonatkozó részéből tartalmaz kivonatot.

Az adatbázis és szerkezetének kialakítása



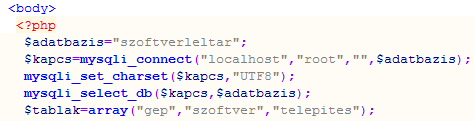






Programozás

Először az adatbázishoz való kapcsolódás és a karakterkódolás beállítása az első lépés. Az adatbázis nevét megadjuk egy változóban és annak tábláit pedig egy tömbben. Erre azért van szükség, mivel az adatbázis műveletekhez egy univerzális megoldást szeretnénk használni: a tömb elemeire hivatkozva, majd azon belül azok mezőit bejárva végezzük el az adatmanipulációs eljárásokat valamint lekérdezve a rekordokat, ezek segítségével helyezzük el azok mezőit a weboldalon létrejövő táblázatokban:

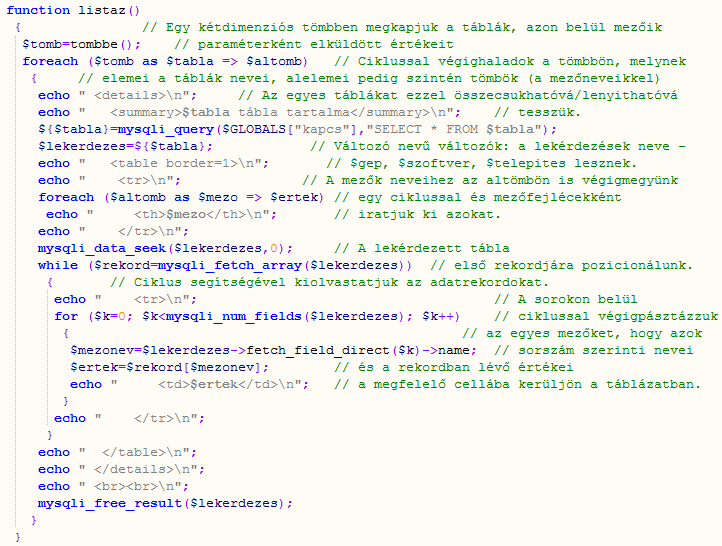


Ezek után, ha nem került elküldésre az egyik ’submit’ típusú gomb sem, azaz nem lett elküldve semmilyen paraméter, nincs beállítva a $\_GET szuperglobális tömb egy eleme sem, az URL a php fájl nevére végződik, akkor alapból az űrlap kerüljön megjelenítésre, a megfelelő méretekben, mezőhosszokkal, alapértelmezett értékekkel, szükséges/opcionális, csak olvasható vagy írható megkötésekkel:

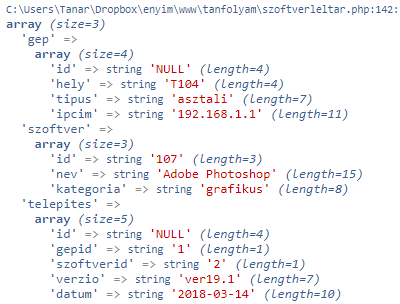


Ha a ’megnéz’ gomb lett elküldve, akkor listázásra kerülnek az adatrekordok HTML táblázat formájában. De mivel ezt az összes gombra kattintás esetében meg kell tenni, ezért ezt egy külön eljárásban érdemes megírni és azt meghívni mindhárom esetben:

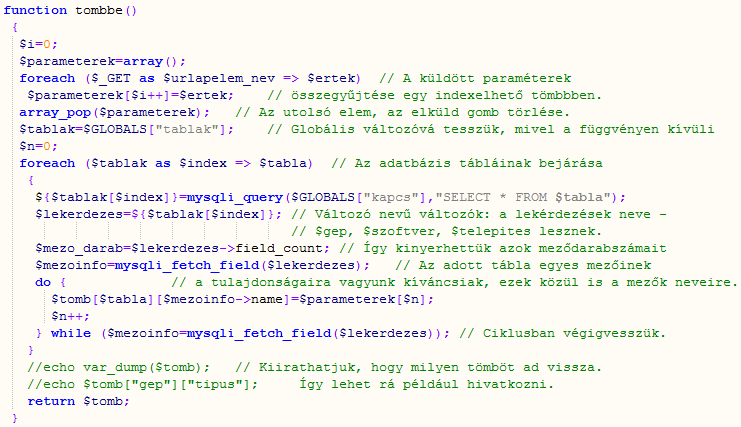




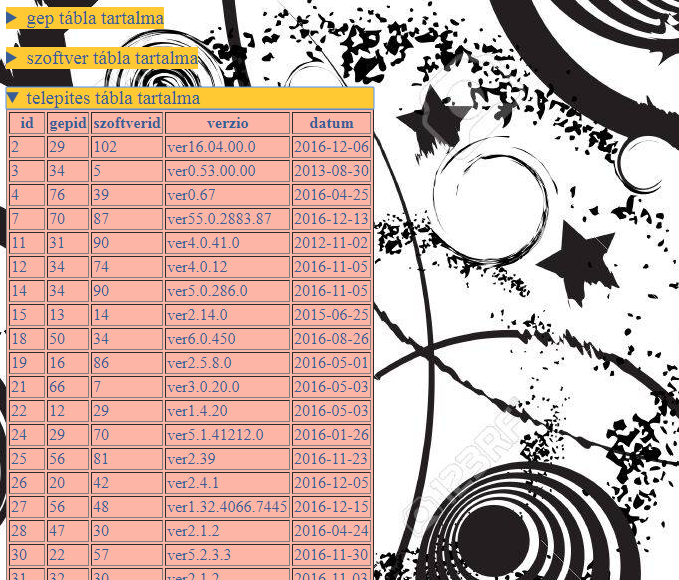
Figyeljük meg, hogy a ’listaz’ eljárás első sorában egy ’tombbe’ nevű függvény kerül meghívásra, ami visszatérési értékként a $\_GET szuperglobális tömb paramétereit és azok értékeit szortírozza szét az adatbázis táblaelnevezései valamint azok mezőnevei szerinti többdimenziós, asszociatív tömbbe:



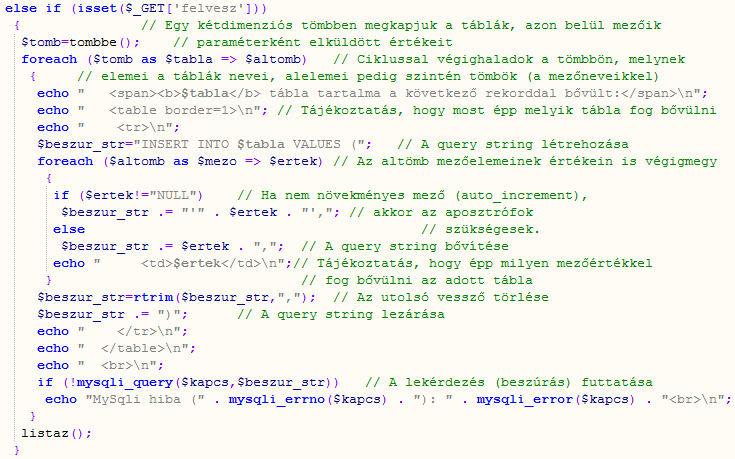
De hogyan jön létre ez a tömb? A már említett ’tombbe’ nevű függvénnyel:



A $\_GET asszociatív tömb – amely a küldött űrlap elem neveit és azok megadott értékeit tartalmazza – egy indexelhető, ’parameterek’ nevű tömbbe helyezzük át, mintegy konvertálva azt. Az utolsó elemét töröljük az ’array\_pop’ függvénnyel, ami az elküldött gomb, mivel erre nem lesz szükségünk. A php forráskódunk elején megadott táblák tömböt bejárjuk és sorban lekérdezzük a tartalmukat, amiket változó nevű lekérdezés változókba helyezünk el. Mindezt azért, hogy hozzájussunk az egyes táblák mezőinek darabszámaihoz, illetve egy hátultesztelős ciklust használva egyéb mezőinfókhoz. A ciklusmagban felépítjük a tömböt, melynek „tábla-adik” és „mezőnevedik” pozíciójába az indexelhető tömb n-edik paraméterértéke kerül. Az így feltöltött tömb kerül átadásra, mint visszatérési érték.

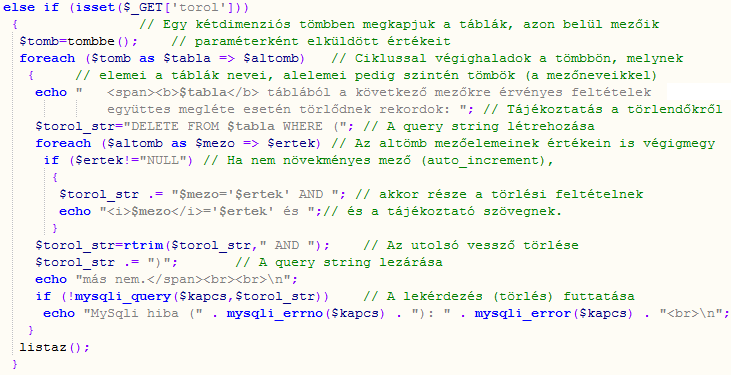
Visszatérve a ’listaz’ eljárásra az így megkapott kétdimenziós tömböt ’foreach’ ciklussal végigpásztázzuk, amelynek első dimenziójában az adatbázis táblái helyezkednek el. Ezeknek kérdezzük le egyenként az összes rekordját változó nevű lekérdezés változókba. Majd elkezdjük a HTML tábla felépítését a kapott sorokból. Közben egy belső ’foreach’ ciklussal az adott tábla mezőin lépegetünk végig, mivel a mezőnevekre szükségünk van a táblázatunk mezőneveinek, mezőfejléceinek megjelenítéséhez. A rekordok HTML táblázatban történő rögzítéséhez egy előltesztelős ciklussal végighaladunk a lekérdezés eredményeként megkapott sorokon (előtte nem baj, ha az elsőre pozícionálunk). Egy ezen belüli ciklussal a kapott rekordok mezőit vesszük sorra, hogy a HTML táblázatunk celláiba el tudjuk őket helyezni. Majd zárjuk a táblázatot, a ’details’ elemet és felszabadítjuk a lekérdezés által lefoglalt helyet. A megnéz metódus (’listaz’ eljárás) futtatása révén kapott képernyőkép:

Ha a ’felvesz’ gomb lett elküldve, akkor a következő szelekciós ág hajtódik végre:



Itt is szeretnék megkapni az asszociatív tömböt táblákba, mezőikbe besorolva az elküldött paraméterértékeket. Majd az eddigiekben is használt egymásba ágyazott ’foreach’ ciklusokkal végig megyünk a táblákon és azok mezőin miközben két dolgot teszünk: létrehozzuk a feladatspecifikációban kért egysoros táblázatokat (amelyek az új rekordokat jelölik) valamint generáljuk a query sztringet, amely jelen esetben az adattáblákba történő beszúrást végző INSERT parancsok lesznek. Mivel a VALUE-ban megadott, vesszővel elválasztott, új mezőértékek utolsó eleme után már nem kell a ciklusban még utoljára kiadott utolsó vessző, így azt egy jobb „nyírással” (rtrim) töröljük. Végül a táblázat sorának és magának a táblázatnak a lezárása után futtatjuk a kapott karakterlánccal a ’mysqli\_query’-t, elvégezve az adott rekord, adott táblába történő beszúrását. Mindhárom tábla adatmanipulációjának elvégzése után listázzuk a tartalmukat, az eredmény nyomonkövethetősége végett. A felvesz metódus kimenete:

A ’torol’ gomb lenyomása esetén is hasonló a metódus:



A kapott tömb tábláit és mezőit bejárjuk azért, hogy szintén létrehozzunk egy query sztringet, de most egy törlésre alkalmasat. Közben még egy tájékoztató karakterláncot is generálunk, amely a megadott mezőket és a hozzá tartozó feltételeket közli velünk, amelyek, ha „összecsengenek” bizonyos rekordokkal, akkor azok törlődnek az adatbázisból. Ez a szelekció is a ’listaz’ eljárás meghívásával zárul. A törlés metódus eredménye:



Legvégül lezárjuk a kapcsolatot:



A feladatspecifikációba lefektetett formázási beállítások háttérképpel, háttérszínekkel, betűszínekkel, szélességek megadásával, keretek- és betűkre vonatkozó beállításokkal, külső- és belső margókkal, lekerekített elemekkel, árnyékolássokkal, kurzor és szövegdekorációval, stb. a következőkben követhetjük nyomon:

