

LEDGER szabad ügyviteli szoftver – közösségi verzió

0. Bemutató

- 0.1 Miért szabad?
- 0.2 Ledger bemutatása
 - 0.2.1. Ledger alapjául szolgáló SQL-Ledger
 - 0.2.2. A filozófia
 - 0.2.3. Ledger, a magyar sajátosságokra kifejlesztett verzió
 - 0.2.4. Modulok, funkciók

1. Kezdeti lépések

- 1.1. Telepítés
 - 1.1.1. Linux telepítés
 - 1.1.2. Hibaüzenetek és magyarázatok
- 1.2. Cégadatbázisok létrehozása
- 1.3. Új felhasználó létrehozása
- 1.4. Törzsadatok felvitele
 - 1.4.1. Vevők törzsadatainak felvitele
 - 1.4.2. Szolgáltatások törzsadatainak felvitele
 - 1.4.3. Szállítók törzsadatainak felvitele
 - 1.4.4. Cikkek törzsadatainak felvitele

2. A programkód

- 2.1. A Ledger kódjának felépítése
- 2.2. Példák
 - 2.2.1. Felület testreszabása
 - 2.2.2. Korlátozott hatáskörű módosítások
 - 2.2.3. Menük testreszabása
 - 2.2.4. Számla bővítése új adatokkal
 - 2.2.5. DirektSQL lekérdezések
 - 2.2.6. További mezők beépítése a lekérdezésekben
 - 2.2.7. Parancssoros hozzáférés

3. Távügyvitel Kft. által nyújtott szolgáltatások

0. Bemutató

0.1. Miért szabad?

A Ledger szabad ügyviteli szoftver használatával több szempontból is *szabadságot* élvezhetsz, amelyek a következők:

- **Hozzáférés szabadsága**
A szoftverhez ingyen hozzá lehet jutni, az internetről való letöltéssel. Nem kell előzetesen összeget kiadni azért, hogy telepíthesd, kipróbálhasd és teljes funkcionalitással használhasd ügyviteled megszervezéséhez. A letöltött verzió nem *demo verzió*, minden funkcionalitás teljesen elérhető benne. A letölthető verzióhoz **azonban nem jár felelősségvállalás**, amelyet a kinyomtatott számlán fel is tüntettünk. A szabályos számla kiállításához módosítani kell a programkódot és a módosítást elvégzőnek Felelősségvállalási Nyilatkozatot kell kiállítani a felhasználó számára.
- **A programkód megtekinthetőségének és módosíthatóságának szabadsága**
A letöltött szoftver tartalmazza a programkódot is, így a hozzáértő szakemberek megtekinthetik és szükség szerint módosíthatják őket. A felhasználó így nem lesz kiszolgáltatva a szoftver készítőjének, saját döntése szerint választhatja meg programozóját testreszabás esetén.
- **A felhasználás szabadsága**
A program korlátlan cég ügyvitelét korlátlan számú felhasználóval képes ellátni, a limitet csak a számítógépének kapacitása szabja meg. Ledger igény szerint használható csak cégen belül (intranet) teljesen elzárva az internettől, vagy interneten keresztül is (a biztonságos körülmények megteremtése ajánlott).
- **A továbbértékesítés szabadsága**
A Ledger programot nem csak saját használatra töltheted le, hanem partnereid számára is. Nincsenek továbbértékesítési korlátozások, amennyiben értesz a szoftverhez felhasználói és programozói szinten, akár Te is bevezetheted, testreszabhatod és támogathatod a szoftvert ügyfeleid számára.
- **A bevezetés szabadsága**
Dönthetsz úgy, hogy idődet nem kímélve autodidakta módon sajátítod el a program bevezetését és üzemeltetését, akár mindenféle költség nélkül. Cégünk különböző támogatási csomagokat kínál, tehát szabadon dönthetsz arról is, milyen mértékű támogatást veszel igénybe, akkor amikor és ahogyan akarsz.

0.2. A Ledger bemutatása

0.2.1. Ledger alapjául szolgáló SQL-Ledger

A Ledger alapjául szolgáló *SQL-Ledger* ügyviteli programot 1998-ban kezdte fejleszteni a DWS Systems nevű kanadai cég, és az azóta eltelt évek alatt az egyik legfejlettebb, és egyben a legígéretesebb szabad szoftverré vált a (szabad) ügyviteli programok piacán. Minek köszönheti ezt a kimagasló eredményt?

- A fejlesztés egy kézben van, egy stílust követ
- Rugalmas, testreszabható
- Platformfüggetlen (Linux, Windows, MacOS)
- A világon egyre többen használják, egyre több előfizetés a támogatásra
- A fejlesztőcég (DWS Systems) biztos anyagi alapokon működik
- Ingyenesen letölthető, a forráskód megvizsgálható
- A hibák gyors javítása a felhasználók visszajelzései alapján

A program megalkotásának célja az volt, hogy a vállalkozások egyszerűen meg tudják oldani azokat a nyilvántartási problémákat, amelyek alapvetően szükségesek a cég működése folyamán; valamint olyan jelentésekkel, illetve listákkal ellátni, amely jól mutatja a cég pillanatnyi helyzetét, ezzel segítve a vezetést a helyes stratégiai döntések meghozatalában.

A 2.4 sorozat a következő feladatokat tudja megoldani: (a teljesség igénye nélkül)

- - SQL adatbáziskezelő: gyors, megbízható, stabil
- - Kliens-szerver felépítés
- - Több raktár kezelése
- - Távoli hozzáférés az interneten keresztül, akár palmtop-ról is
- - Összekapcsolható internetes web-áruházzal
- - Ideális több telephely esetén
- - Naprakész információk
- - Minden adatot csak egyszer kell bevinni a rendszerbe (pl. rendelés -> szállítás -> számlázás -> könyvelés)
- - Több deviza kezelése
- - Korlátlan számú ügyfél, adatbázis
- - Partnerekhez rendelhető kedvezmények, fizetési határidők
- - Jogosultság beállítása
- - Lekérdezések exportálása: Excel, Open office,...
- - Utólagos módosítás lehetséges, egyben szükség esetén letiltható (audit kontroll)

0.2.2. A filozófia

A *Ledger* nem a bevett könyvelői megközelítés alapján lett megalkotva, hanem úgy, hogy könyvelési tudás nélkül is lehessen a programot használni. A könyvelőprogramok többsége elsőnek elvárja, hogy egy bizonylatot (pl. áru beszerzési számlát) könyvelésben rögzítsék --ez a jól ismert Tartozik / Követel tételek rögzítése --, majd ezek után kerül sor a konkrét bizonylat adatainak rögzítésére, tételeinek felvitelére.

A *Ledger* azonban csak azt várja el a felhasználótól, hogy a bizonylat adatait rögzítse, mivel a rögzítéssel egyidejűleg a háttérben az --a program beüzemelésakor beállított törzsadatok alapján-- a megfelelő főkönyvi számlákra lekönyveli. A cég könyvelőjére csupán az ellenőrzés, illetve a számlákhoz, bizonylatokhoz nem kötődő könyvelés tételek (pl. bérek, értékcsökkenési leírás, zárás) rögzítésének feladata hárul.

0.2.3. Ledger, a magyar sajátosságokra kifejlesztett verzió

Hasonlóan a többi külföldön fejlesztett programhoz, a *Ledger* eredeti angol verziója (<http://www.sql-ledger.org>) is elsősorban globális megközelítést alkalmaz, tehát nem tud figyelemmel lenni a helyi sajátosságokra. Az alapvető könyvelési eljárások ez egész világon egyformák, mégis az ügyvitel során az alkalmazott módszerek, előírások jelentősen eltérhetnek.

Hasonlóképpen van ez a magyar sajátosságokkal is. Cégünk az Távügyvitel Kft. (<http://www.tavugyvitel.hu>) évek foglalkozik az Ledger magyar cégeknél történő bevezetésével, illetve a program továbbfejlesztésével. A folyamatos konzultáció és fejlesztőmunka eredményeképpen, az eredeti angol programot továbbfejlesztve, úgy alakítottuk át a programot, hogy az eredeti verzió előnyös tulajdonságait megtartva a magyar felhasználók számára is barátságos, és könnyen kezelhető legyen a felület, valamint megfeleljen a magyar előírásoknak.

A nagyobb változtatások listája a következő:

- Magyar fordítás beleértve a sablonokat is
- Magyar előírásoknak megfelelő számlázás (példányszámozás, számla kelte mező, sorszám folytonossága stb.)
- Bizonylatok rögzítésekor adóalap szerinti megkülönböztetés, kiválasztás
- Kézpénzes, átutalásos bizonylatok megkülönböztetése
- Bank, pénztár különválasztása
- Magyar módszer szerinti bankkivonat feldolgozás
- Továbbfejlesztett ÁFA analitika
- Kartonok megjelenítése ellenszámlával
- Vegyes, Pénztár/Bank, Szállító, Vevőnaplók
- A menü átszerkesztése
- +számos egyéb a használatot megkönnyítő funkciók

0.2.4. Modulok, funkciók

A Ledger moduljai, főbb funkciói a következők:

- **Számlázás**
 - Magyar rendeleteknek, törvényeknek megfelelel
 - Forint, devizaszámlázás
 - Kétnyelvű (export) számlasablon
 - Nyomatás közvetlenül képernyőre PDF, Postscript illetve html formátumban
 - A képernyőre megjelenített számla a világ bármely pontjáról elkészíthető, elmenthető
 - Stornó, helyesbítő számla kiállítása
 - Számlamásolat küldése e-mailen PDF formátumban
 - Saját igényeknek megfelelően a számla sablonja módosítható, saját logó beilleszthető
 - Előleg számla, Proforma számla készítés lehetősége
 - A számla tételeinek keresése a törzsadatállományból többféleképpen lehetséges
 - A törzsadatban beállított alapadatok a számla készítése alatt módosíthatóak
 - Többféle számlaformátum használata lehetséges egyszerre
 - A rögzített száma automatikus könyvelése a háttérben
 - Vevő, szállító folyószámla nyilvántartás
 - Tömbös vevő-, költség számla automatikus könyvelése
 - Kifizetett, kifizetetlen számlák listázása, többféle paraméter alapján való szűrése
 - A megjelenített számlák listájának további rendezése lehetséges, részösszegek számítása
 - A megjelenített listában keresés szövegrészlet alapján
 - Kifizetetlen számlák egy múltbéli dátumig
 - Lejárati lista, felszólítólevél nyomtatása
 - Vevő, szállító devizaszámlák listája (analitika), devizaárfolyam, összeg, devizanem oszlopokkal
 - Szállító, vevő számlák tételeinek listázása vevők, szállítók alapján, elemzés
 - A listák könnyen átemelhetők táblázatkezelőkbe (pl. Excel, Openoffice)
- **Ajánlatok, rendelések**
 - Rögzített ajánlatok, rendelések továbbküldése e-mailen PDF formátumban
 - Ajánlatból rendelés, rendelésből számla készíthető egy gombnyomással
 - Ajánlat és rendeléssablonok módosíthatóak
 - Rendelések ajánlatok lezárása automatikusan illetve manuálisan
 - Részteljesítés esetén automatikus rendeléskészítés (backorder)
- **Bank, Házipénztár**
 - Korlátlan bank, pénztár kezelése
 - Több számla egyidejű kiegyenlítése
 - Automatikus pénztárbizonylat-sorszámozás
 - Bevételi, kiadási pénztárbizonylat nyomtatása, utólag is tömegesen
 - Időszaki pénztárjelentés
 - Bankkivonat egyeztetése a bank főkönyvi számlával
 - Előlegek nyilvántartása
- **Raktár**
 - Korlátlan raktár (fizikai, logikai)
 - A raktárak hozzáférése felhasználókhöz köthető
 - A raktármozgás együtt történhet a számla kiállításával, rendelés rögzítésével is
 - Szállítólevél, bevételezési lista nyomtatása a folyamat több pontján is lehetséges

- Transzfer egyik raktárból a másikba, transzferbizonylat
- Paramétrezhető készletlisták
- Rendelt, kiszállított, vásárolt, eladott cikkek listája
- Raktári mozgások múltbéli története mozgásnem, időszak, cikk(csoport), bizonylat alapján
- Jövőbeli készletek lekérdezése (rögzített vevő és szállítórendelések alapján)
- Gyári szám követés a bizonylatokon
- Speciális többszintű árazás időszakhoz, mennyiséghez, partnerekhez kötve
- Százalékos kedvezmény partnercsoportokra, partnerekre, vagy csupán egyes tételekre
- A cikkekhez kép,rajz, mikrofilm linkelhető
- Saját termék összeállítás a raktáron lévő cikkekből (receptúra), automatikus készletlevonás raktárra vételkor
- A receptúra többszintű lehet (cikk, félkésztermék, termék)
- **Pénzügyi jelentések**
 - Fizetendő, Levonható ÁFA analitika, ÁFA kulcs szerint rendezhető
 - Számlatükör, Főkönyvi kivonat projekt, költséghely profitcenter alapján, időszakos szűkítéssel
 - Mérleg, eredménykimutatás, a sablon módosítható
 - Számlák lekérdezése projektek alapján
 - A jelentések könnyen átmenthetők táblázatkezelőbe
 - SQL lekérdezések a felületről, html és xls formátumban
- **Könyvelés**
 - Bejövő, kimenőszámlák automatikus könyvelése a megadott főkönyvi számlákra
 - Árfolyamnyereség, veszteség automatikus könyvelése
 - Vegyes könyvelés a bizonylathoz nem köthető tételek könyvelésére
 - Kartonok, naplók nyomtatása többféle paraméter alapján
 - Audit kontroll: A könyvelés lezárható egy adott napig
 - Főkönyvi számlák automatikus zárása a felhasználó által megadott főkönyvi számlára

1. Kezdeti lépések

1.1. Telepítés

1.1.1. Linux telepítés

Előzetesen szükséges telepíteni a következő szoftvereket:

Perl, 5+
http szerver (Apache)
SQL szerver tranzakciókezeléssel (PostgreSQL 7.0+)
Adatbázismeghajtó: DBI, és a specifikus meghajtó: DBD-Pg
Számlyomtatáshoz: LaTeX, valamint Postscript képes nyomtatási rendszer és nyomtató

1. a, A program futásához előzetesen a *PostgreSQL* adatbázisban aktiválni kell a *plpgsql* procedurális nyelvet.

Meg kell keresni a *createlang* helyét és ki kell adni a parancsot root felhasználóként:

```
$ createlang -U postgres plpgsql template1
```

Ha nem kapunk visszajelzést, vagy

```
createlang: language "plpgsql" is already installed in database "template1"
```

üzenetet kapjuk vissza, akkor a parancs sikerült.

Amennyiben hibajelzést kapunk akkor manuálisan kell telepíteni a *plpgsql* nyelvet:

```
$ psql -U postgres template1
```

majd ki kell adni a parancsokat:

```
CREATE FUNCTION plpgsql_call_handler () RETURNS LANGUAGE_HANDLER AS  
    '$libdir/plpgsql' LANGUAGE C;
```

és:

```
CREATE TRUSTED PROCEDURAL LANGUAGE plpgsql  
    HANDLER plpgsql_call_handler;
```

1. b, A *PostgreSQL* adatbázishoz adatbázisfelhasználókat kell létrehozni, amelyet pl. a :

```
$ createuser -U postgres ledger
```

paranccsal tehetünk meg, ahol a *ledger* név az alapértelmezett *PostgreSQL* felhasználónév a programban (Más nevet is lehet használni, természetesen). A kérdések megválaszolásakor engedélyezni kell az új adatbázis létrehozásának jogát.

Az új adatbázis-felhasználó nevét kell használni majd a programban az új adatbázis és felhasználó létrehozásakor.

1. c, Meg kell keresni a *pg_hba.conf* állományt, és átírni a hasonló sorokat erre:

```
host all all 127.0.0.1 255.255.255.255 trust  
local all all trust
```

És újraindítás.

Tehát megbízunk minden felhasználóban aki a helyi gépen keresztül (ahol a *Ledger* fut) intéz parancsot a *PostgreSQL*-nek.

2.A letöltött tar.gz file kicsomagolása *usr/local* könyvtár alá (amennyiben vannak állományok az

`/usr/local/ledger` könyvtár alatt, mindent ki kell törölni a kicsomagolás előtt) :

```
cd /usr/local
tar -xzvf xxx_xxx.tar.gz
```

3. Kicsomagolás után az `sql-ledger.conf.default` file-t át kell nevezni `sql-ledger.conf` névre, amely állományban a nyomtató parancssor(oka)t lehet beállítani. A legtöbb Linux gépen működik az `lpr` parancs, ezért érdemes az előre beállított parancssort használni ('lpr' => 'Nyomtató'). Ha nem működik, azt a parancssort kell beírni, amivel a konzolról tudunk nyomtatni . Például, ha a

```
lpr -P Xerox valami.txt
```

működik, akkor a beírandó parancssor:

```
%printer = ('lpr -P Xerox' => 'Nyomtató');
```

4.A `http` szerver beállítása következik, ahol is meg kell keresni a `httpd.conf` konfigurációs file helyét és a következő bejegyzést kell tenni:

```
Alias /ledger/ /usr/local/ledger/
<Directory /usr/local/ledger>
    AllowOverride All
    AddHandler cgi-script .pl
    Options ExecCGI Includes FollowSymLinks
    Order Allow,Deny
    Allow from All
</Directory>
<Directory /usr/local/ledger/users>
    Order Deny,Allow
    Deny from All
</Directory>
```

4.B. Meg kell keresni a következő sort és érvényteleníteni kell (ki kell kommentezni):

```
#AddDefaultCharset ISO-8859-1
```

5.Végül lehetővé kell tenni a szerver számára, hogy írhasson bizonyos könyvtárakba:

```
cd /usr/local/ledger
# chown -R nobody:nobody users templates spool
```

ahol `nobody:nobody` a webszerver felhasználó. Pl: Fedora alatt `apache:apache`, Debian alatt `www-data:www-data` stb....

1.1.2. Hibaüzenetek és magyarázatuk

Not Found

The requested URL `/ledger/` was not found on this server.

Az `Alias` parancs nincs megjelenítve a szerverprogram (`apache`) konfigurációs file-ban.

```
Alias /ledger/ /usr/local/ledger/
```

hiányzik, vagy beírás után nem lett újraindítva az `apache`.

Error!

users/members : No such file or directory

Nincs még létrehozva a megadott nevű Ledger felhasználó.

Hiba!

users/members : Permission denied

A users könyvtár nem írható az apache számára:

```
cd /usr/local/ledger
# chown -R nobody:nobody users templates spool
parancs nem lett kiadva.
```

Forbidden

You don't have permission to access /ledger/login.pl on this server.

A ledger könyvtár, vagy a *login.pl* futtatása nincs engedélyezve a (linux) felhasználó számára.

Hiba!

FATAL: a(z) "ledger" felhasználó nem létezik

Nincs létrehozva a PostgreSQL *ledger* adatbázis-felhasználó

```
$ createuser -U postgres ledger
```

could not connect to server: dÿi¿P

Az *Adatbázis helye / Host* nem lett üresen hagyva, vagy rosszul lett kitöltve, vagy nem fut a PostgreSQL daemon.

A böngészőben nem jelennek meg jól az ékezetes betűk

A *httpd.conf*-ban ki kell kommentezni az alapbeállítású Karakterkészletet, pl.:

```
# AddDefaultCharset ISO-8859-1
```

**Undefined subroutine &main::mentés called at bin/mozilla/admin.pl line 65.
Compilation failed in require at /usr/local/ledger/admin.pl line 91.**

A *httpd.conf*-ban ki kell kommentezni az alapbeállítású Karakterkészletet pl:

```
# AddDefaultCharset ISO-8859-1
```

Error!

connectDBStart()—socket failed: errno=2 No such file or directory

Windows alatt úgy akarunk létrehozni adatbázist, új felhasználót, hogy nem adtuk meg az *Adatbázis helyét* (localhost)

**ERROR: encoding LATIN2 does not match server's locale hu_HU.UTF-8
DETAIL: The server's LC_CTYPE setting requires encoding UTF8.**

A PostgreSQL adatbázis alapértelmezésben UTF-8 karakterkészletet használ, amely nem működik együtt a LATIN2 kódolással. Új adatbázis-struktúrát kell létrehozni LATIN2 támogatással pl. Fedora 10 esetén:

1. `$ /etc/init.d/postgresql stop`
2. `/var/lib/pgsql/data` könyvtár törlése (VIGYÁZAT: minden eddig létrehozott adatbázis elvész)

3. `$ /usr/bin/initdb -D /var/lib/pgsql/data --locale=hu_HU.ISO-8859-2`
4. *ledger* felhasználó létrehozása, *plpgsql* nyelv létrehozása, *pg_hba.conf* átírása (lásd 1.a, 1.b, 1.c pontok)
5. `$ /etc/init.d/postgresql start`

Ubuntu esetén:

1. előkészítés

```
cd /var/lib/locales/supported.d
```

'local' file -ba bele kell szerkeszteni :

```
hu_HU ISO-8859-2
```

mentés, majd

```
dpkg-reconfigure locales
```

így már ismerni fogja az iso8859-2 -t.

2. Postgresql inicializálás

állítsuk le az esetleg futó postgrest

```
/etc/init.d/postgresql-8.3 stop
```

mentsük el a */var/lib/postgresql/8.3/main* tartalmát valahova
töröljük a */var/lib/postgresql/8.3/main* könyvtárat;

adjuk ki :

```
/usr/lib/postgresql/8.3/bin/initdb --locale=hu_HU.iso-8859-2 -D  
/var/lib/postgresql/8.3/main >/tmp/init.txt
```

ha nincs hibaüzenet, másoljuk vissza a
postmaster.opts
root.crt
server.crt
server.key

fileokat, majd

```
/etc/init.d/postgresql-8.3 restart
```

Attempt to free unreferenced scalar: SV 0x8d67e5c, Perl interpreter: 0x85c7008 at SL/IS.pm line 240.

A telepített Ledger verzió csak az 5.8.x perl verzióval működik együtt.

I can't find the format fájl: 'latex.fmt'!

root felhasználóként le kell futtatni a következő parancsot:



<http://www.tavugyvitel.hu>
tel: (061) 266-1039

```
$ fmtutil-sys --byfmt latex
```

1.2. Cégszerver adatbázisok létrehozása

Egy cég ügyviteléhez létre kell hozni egy adatbázist, amelybe a program menti az adatokat. Amennyiben több cég ügyvitelét szeretné vezetni a programban mindegyik cégnek külön-külön létre kell hozni egy cégszerver adatbázist.

A cégszerver adatbázisokat az adminisztrációs felületen lehet létrehozni az `http://<ledger gyökérkönyvtár>/admin.pl` parancs kiadásával.

Nyissuk meg a böngészőnk, és üssük be a következő címekeket.

Példák:

`http://localhost/ledger/admin.pl` - ha ugyanazon a gépen van az adatbázis mint a felhasználói böngésző;

`http://192.168.1.1/ledger/admin.pl` - ha belső vállalati hálózaton használják (intranet);

`https://www.tavugyvitel.hu/mintakft/admin.pl` - ha távoli eléréssel használjuk a programot.

Az első használatkor nem kell jelszót megadni.

Belépés után a létező felhasználókat látjuk, és különböző menüpontokat:

Ledger		
Belépés	Név	Társaság
dbteszt	Tav Test	Tav Test
demo10	Horváth István	Marketing Demo Kft.
demo40	Molnár Lajos	Manager Demo Kft.
demo40_2	demo40_2	Investor.hu Kft.
investorhu	ADMIN Tav ügyvitel	Távügyvitel Kft. 2.
investorhu2	ss	ss
konyvelo	Countplus Kft.	Távügyvitel Kft.
sasdi	Sásdi András	Sásdi Kft.
tav_teszt	Távügyvitel Teszt	Távügyvitel Teszt Kft.
tavtest	Tav Test	Tav Test
tavtest3	tav test3	--
tavteszt2	Kabai J	--
tavugyvitel	Tav Ugyvitel	XYZ Szolgáltató Kft.
tomani	Tomani Tomani	Tomani Bt.

Új felhasználó	Rendszergazda jelszavának megváltoztatása
----------------	---

Módosításhoz klikkeljen a felhasználó nevére!

A Rendszergazda jelszavának megváltoztatása menüben változtassuk meg jelszavunkat, majd a **Postgresql adatbázis felügyelet** gombot nyomjuk meg:

Ledger Könyvelés / Adatbázis felügyelet

Adatbázis	
Adatbázis helye:	Port száma
Felhasználó ledger	Jelszó
Csatlakozva ehhez: template1	
Új cégszerver adatbázis	Cégszerver adatbázis frissítése
	Cégszerver adatbázis törlése

A meglévő adatbázisok előzetes ellenőrzése. A program itt még nem töröl és nem hoz létre semmit!
 Hagyja üresen a Hely és a Port mezőket, ha csak nem akar távoli kapcsolatot létesíteni

Mezők kitöltése:

Adatbázis helye:

Amennyiben az adatbázis ugyanazon a gépen van, mint a program és a szerver, akkor

- Linux szerver esetében üresen kell hagyni
- Windows szerver esetében : localhost

Ha az adatbázis más szerveren van, akkor annak a szervernek nevét (ip címét) kell megadni.

Felhasználó (és jelszó):

Az a felhasználónév, amelyet a PostgreSQL adatbázisban hoztunk létre, úgy, hogy képes új adatbázisokat létrehozni. Alapesetben már létezik egy ilyen (postgres), de érdemes újat létrehozni.

Példa:

```
$ createuser -U postgres ledger
```

A port száma, és a csatlakozva ehhez: mezőket nem kell kitölteni, megváltoztatni.

Az **Új cégadatbázis** gomb megnyomása után az **Új cégadatbázis** mezőbe be kell írni az adatbázis nevét, és a többi érték változtatlanul hagyása után a **Folytatás**-ra kattintva a program létrehozza a cégadatbázist, amelyet a **Cégadatbázis sikeresen létrehozva** üzenettel nyugtáz a folyamat végén.

A fenti képen egy példabeállítás látható. A bal oldalon a belépéshez és a számlázáshoz szükséges adatok találhatóak.

A jobb oldalon a *Lenyíló limit* értékénél nagyobb számú vevő / szállító esetén a program nem lenyílóablakban jeleníti meg őket, hanem egy input mezőbe kell beírni a vevő/szállító nevét illetve néhány betűjét és az összes ilyen találatból ki kell választani a végsőt.

A *Menü szélessége* szabályozza, hogy a menü kerete milyen széles legyen pixelben.

A *betű kódlap* a magyar ISO-8859-2 legyen a kalapos ú,ő stb. elkerülése végett.

A *munkamenet időtartama (mp)* mezőben kell megadni másodpercben a munkamenet hosszát.

A *Nyomtató* lenyílóablak azokat a nyomtatóbeállításokat tartalmazza, amelyet az `sql - ledger . conf` állományban lehet megjelölni a helyi nyomtató nevére.

Végül kiválaszthatjuk mely felhasználó sablonjait használjuk, illetve ha újakat akarunk akkor az *Új sablon* mezőben kell ezt egy névvel jelezni. Ekkor ezen a néven a program létrehoz egy könyvtárat a `<gyökér>/templates` könyvtárban és ide menti le az *Új sablon nyelve* pontban megadott nyelvű sablontípusokat.

A *Cégadatbázis* mezőben azt kell megadnunk, mely cégadatbázishoz kapcsolódik a felhasználó. A *Felhasználó* mezőben pedig azt az adatbázisfelhasználót kell megadni, amely létrehozta, illetve jogosultságot kapott a cégadatbázishoz. Windows telepítés esetén az *adatbázis helye* mezőbe 'localhost' értéket kell írni.

1.3. Új felhasználó létrehozása

Példa: A könyvelője képviselői szolgáltatást nyújt Önnek, ezért hozzáférést kell biztosítani számára a bevételek idején. A külön felhasználónév létrehozásával a könyvelő csak azokat a menüket éri el, amelyre szüksége van.

Nyomja meg az **Új felhasználó** gombot!

Ledger Könyvelés Rendszerfelügyelet / Új felhasználó	
Belépés	Dátumformátum
Jelszó	Számformátum
Név	Lenyíló ablak limit
E-mail	Menü szélessége
Aláírás	Nyelv
Telefon	Betű kódlap
Fax	Munkamenet időtartama (mp)
Társaság	Dokumentumok könyvtára
adószám	Stíluslap
	Nyomtató
	Sablon használata
	Új sablon
	Új sablon nyelve
Adatbázis	
Meghajtó	Adatbázis helye:
Adatbázis	Port száma
használó	Jelszó
Hozzáférési jogok	
használó	
Bevételek	<input checked="" type="checkbox"/> Új számla kiállítása <input checked="" type="checkbox"/> Új átutalásos rögzítése

Belépés, Jelszó

A megadott felhasználónévvel és jelszóval tud bejelentkezni a Ledger bejelentkezési felületén.

Név

A felhasználó teljes neve, pl. Molnár József.

Társaság, cím, adószám:

Számlázás során használt kötelezően kitöltendő adatok ,a számla kiállító adatai.

Lenyíló ablak limit

Amennyiben a program űrlapjain megjelenített lenyíló ablakok tételeinek száma meghaladja az itt megadottat, a program keresőmezőt kínál fel a lenyíló ablak helyett.

Munkamenet időtartama

A bejelentkezés után a program automatikusan jelszót kér, ha az itt megadott időtartamig (másodpercben kell megadni: 3600 / 60 = 60 perc) nem történik esemény.

A mezők kitöltése után már csak a felhasználó által elérhető menüpontokat kell kijelölni, és a **Mentés** gombot megnyomni.

Tipp: A felhasználó létrehozása során az esetek túlnyomó többségében bőven elég a *Belépés, Jelszó, E-mail, Társaság, Cím, adószám* kitöltése valamint az elérhető menüpontok beállítása!

1.4. Törzsadatok felvitele

Ahhoz, hogy bizonylatainkat rögzíthessük, előzetesen fel kell tölteni a rendszert törzsadatokkal. A legfontosabb törzsadatok a következők:

- *Főkönyvi számlatükör*
Főkönyvi számlák, az automatikus könyveléshez. A telepítés során egy példa számlatükör már betöltődik, nem kell vele foglalkoznia, csupán akkor ha éles könyvelésre szeretné használni a programot.
- *Vevők*
Név, cím, ÁFA kulcs stb.
- *Szolgáltatások*
Szolgáltatások törzsadatai, pl. kódszám, szöveges leírás, eladási ár, speciális ár, ÁFA kulcs stb.
- *Szállítók*
Név, cím, ÁFA kulcs stb.

Tipp: Cégünk a Tárvügyvitel Kft. fontosnak tartja, hogy minél simábban és problémamentesebben tudja elindítani ügyvitelének vezetését a rendszerben. Ezért vállaljuk a legfontosabb törzsadatok, nyitó főkönyvi számlák, nyitott vevő- szállítószámlák tömeges bevitelét, amennyiben ezt Ön elektronikus formában valamelyik táblázatkezelő formátumában (Excel, Openoffice) eljuttatja hozzánk! Kérjük, konzultáljon velünk!

1.4.1. Vevők törzsadatainak felvitele

Karbantartás / Vevők / Új vevők felvitele menüpontban a *kékkel* színezett mezőket mindenképpen ki kell tölteni:

Új vevő	
Számlázási cím	Szállítási cím
Vevő száma <input type="text" value="4"/> U: 3	<input type="text" value=""/>
Cég neve <input type="text" value="Példa Kft."/>	<input type="text" value=""/>
Irányítószám <input type="text" value="1112"/>	<input type="text" value=""/>
Város <input type="text" value="Budapest"/>	<input type="text" value=""/>
Cím <input type="text" value="Fő u. 12"/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>
Ország <input type="text" value=""/>	<input type="text" value=""/>
Kapcsolattartó <input type="text" value=""/>	<input type="text" value=""/>
Telefon <input type="text" value=""/>	<input type="text" value=""/>
Fax <input type="text" value=""/>	<input type="text" value=""/>
E-mail <input type="text" value=""/>	<input type="text" value=""/>
E-mail másolat <input type="text" value=""/>	<input type="text" value=""/>
<input checked="" type="checkbox"/> Fizetendő ÁFA 20% <input type="checkbox"/> Fizetendő ÁFA 15% <input type="checkbox"/> Bruttó adótartalom <input type="checkbox"/> Fizetendő ÁFA 5% <input type="checkbox"/> Fizetendő ÁFA adómentes	
Hitelkeret <input type="text" value=""/>	Fizetési határidő <input type="text" value="7"/> nap
Adószám <input type="text" value=""/>	Fizetési mód <input type="text" value="átutalásos"/>
	Teljesítés helye <input type="text" value=""/>
	Deviza <input type="text" value="HUF"/>
Megjegyzés <input type="text" value=""/>	Engedmény <input type="text" value=""/> %
	Bankszámlaszám <input type="text" value=""/>
	Számlatípus <input type="text" value=""/>

Vevő száma

Használjuk mindig a következő sorszámot. A legutoljára rögzített számot a rendszer a mező mellett mutatja, jelen példában ez a 3-as volt.

ÁFA kulcs

Minimum egy ÁFA kulcsot ki kell választani.

Fizetési határidő

Mindenképpen értéket kell adni neki, ha nincs határidő akkor 0.

Fizetési mód

Bármilyen lehet, de jellemzően: készpénzes, átutalásos, csekk, utánvét
Vevőszámlakészítés során ezt lehet módosítani.

Számlatípus

Ha a Deviza EUR vagy USD, akkor a számlatípusnak export értéket kell adni, így a felvitt vevőnek kétnyelvű számlát lehet kiállítani.

Megjegyzés

Az itt bevitt szöveg automatikusan bemásolódik a belső megjegyzés rovatba számlakészítés során.

Szállítási cím

Amennyiben más mint a számlázási cím. Ezek az adatok nem jelennek meg a számlán.

A **Mentés** gomb megnyomása után az adatok rögzítésre kerülnek.

A **Vevőszámla** gomb megnyomása után az adatok rögzítésre kerülnek, és egyből készíthetjük a számlát az rögzített vevő nevére.

Új vevőt közvetlenül is felvihetünk a számlakészítés felületről is az **Új vevő** gomb megnyomásával.

1.4.2. Szolgáltatások törzsadatainak felvitele

Karbantartás / Szolgáltatások / Új szolgáltatás

Új szolgáltatás							
Szám	Szöveges leírás						
U1	Ügyviteli szolgáltatás						
Számkapcsolatok							
Árbevétel	9110--Belföldi árbevétel					Frissítve	2007-05-30
Költség	5990--Költségek					Listaár	
Eladási ár							
Egység							
VTSZ/SZJ							
Adó <input checked="" type="checkbox"/> 4670--Fizetendő ÁFA 20% <input type="checkbox"/> 4671--Fizetendő ÁFA 15% <input type="checkbox"/> 4672--Fizetendő ÁFA 5% <input type="checkbox"/> 4673--Fizetendő ÁFA adómentes							
Megjegyzés							
Vevő	Határ	Eladási ár	Dev	Mikortól:	Meddig:		
			HUF				
<input type="button" value="Adatok frissítése"/> <input type="button" value="Mentés"/>							

Árbevétel

Melyik főkönyvi számlára gyűjtse a szolgáltatásból származó bevételt.

Listaár

Csak információs célt szolgál, nem jelenik meg a számlán.

Eladási ár

Megjelenik a számlán, de módosítható. Akkor érdemes kitölteni, ha túlnyomórészt fix áron történik a szolgáltatás eladása

ÁFA kulcs

Legalább egy ÁFA kulcsot ki kell jelölni. Ha EU-n belül értékesítik és belföldön is, akkor a *Fizetendő ÁFA 20%* és a *Fizetendő ÁFA adómentes* kulcsokat is ki kell jelölni. A számlázás során a *vevő ÁFA kulcs* beállítása fogja eldönteni melyikkel számol az adott szolgáltatásnál.

Speciális ár

Az eladási áron kívül *Vevőkre* illetve *Árcsoportokra* vonatkozó speciális árat is megadhatunk. Válasszuk ki a vevőt, illetve az árcsoportot, majd adjuk meg a rá vonatkozó speciális árat. Opcionálisan megadhatjuk, hogy hány darab vásárlása felett érvényes ár (Határ), és akár azt is ettől meddig érvényes az ár. Amennyiben ezek a feltételek nem teljesülnek, a program az eladási árat ajánlja fel.

Tipp: Árcsoportot előzetesen a *Karbantartás / Árcsoportok* menüpontban hozhatunk létre, majd egy vevőt a törzsadat űrlapján tudjuk egy adott árcsoportba besorolni.

1.4.3. Szállítók törzsadatainak felvitele

Karbantartás / Szállítók / Új szállító felvitele menüpontban a *kékkel* színezett mezőket mindenképpen ki kell tölteni:

Új szállító	
Számlázási cím	Szál
Szállító száma <input type="text" value="3"/> U: 2	<input type="text"/>
Cég neve <input type="text" value="Teszt Szállító Kft."/>	<input type="text"/>
Irányítószám <input type="text"/>	<input type="text"/>
Város <input type="text"/>	<input type="text"/>
Cím <input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
Ország <input type="text"/>	<input type="text"/>
Kapcsolattartó <input type="text"/>	<input type="text"/>
Telefon <input type="text"/>	<input type="text"/>
Fax <input type="text"/>	<input type="text"/>
E-mail <input type="text"/>	<input type="text"/>
E-mail másolat <input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/> Visszaigényelhető ÁFA 20%	
Adó: <input type="checkbox"/> Visszaigényelhető ÁFA 15%	<input type="checkbox"/> Bruttó adótartalom
<input type="checkbox"/> Visszaigényelhető ÁFA 5%	
<input type="checkbox"/> Visszaigényelhető ÁFA adómentes	
Hitelkeret <input type="text"/>	Engedmény <input type="text"/> %
Adószám <input type="text"/>	
Bankszámlaszám <input type="text"/>	
Deviza <input type="text" value="HUF"/>	
Megjegyzés <input type="text"/>	

A vevőkkel ellentétben itt jóval kevesebb kötelező adatot kell rögzíteni, a képen látható a minimális kitöltés igény a Teszt Szállító Kft. esetében. Amennyiben bővebb nyilvántartást vezet szállítóiról, akkor érdemes a cím, kapcsolattartó, bankszámlaszám stb. mezőket kitölteni.

Ha a **Mentés** helyett a **Szállító tranzakció**, **Beszerzési számla**, **Beszerzési rendelés**, **Ajánlatkérés** gombok valamelyikét nyomja meg, a szállító törzsadatainak mentését követően a gomb címének megfelelő beviteli úrlapra ugrik a program, az Új szállító rögzítése felajánlása helyett.

1.4.4. Cikkek törzsadatainak felvitele

Raktár / Cikkek / Új cikk

Árbevétel

Melyik főkönyvi számlára gyűjtse a cikkből származó bevételt.

Listaár

Csak információs célt szolgál, nem jelenik meg a számlán.

Eladási ár

Megjelenik a számlán, de módosítható. Akkor érdemes kitölteni, ha túlnyomórészt fix áron történik a cikk eladása

ÁFA kulcs

Legalább egy ÁFA kulcsot ki kell jelölni. Ha EU-n belül értékesítik és belföldön is, akkor a *Fizetendő ÁFA 20%* és a *Fizetendő ÁFA adómentes* kulcsokat is ki kell jelölni. A számlázás során a *vevő ÁFA kulcs* beállítása fogja eldönteni melyikkel számol az adott cikknél.

Speciális ár

Az eladási áron kívül *Vevőkre* illetve *Árcsoportokra* vonatkozó speciális árat is megadhatunk. Válasszuk ki a vevőt, illetve az árcsoportot, majd adjuk meg a rá vonatkozó speciális árat. Opcionálisan megadhatjuk, hogy hány darab vásárlása felett érvényes ár (Határ), és akár azt is, hogy mettől meddig érvényes az ár. Amennyiben ezek a feltételek nem teljesülnek, a program az eladási árat ajánlja fel.

Tipp: Árcsoportot előzetesen a *Karbantartás/Árcsoportok* menüpontban hozhatunk létre, majd egy vevőt a törzsadat űrlapján tudjuk egy adott árcsoportba besorolni.

2. A programkód

2.1. A Ledger kódjának felépítése

A gyökérkönyvtárban lévő futtatható perl állományok töltik be a bin/mozilla könyvtárban lévő azonos nevű perl állományokat, amelyek a funkciókat tartalmazzák. A funkciók az SL könyvtárban lévő modulokat hívják meg, amelyek főleg az SQL adatbázissal kapcsolatos műveletekért felelősek.

Például a Bevételek/Új számla kiállítása menüre kattintva a program meghívja a bin/mozilla/is.pl állomány add funkcióját, amely a számlakiállítás űrlapjának megjelenítéséért felelős. Az add funkció megrajzolja az űrlapot valamint betölti az SL/IS.pm modult (use SL::IS.pm) amely az adatbáziskapcsolatért felelős rutinokat tartalmazza.

A könyvtárak tartalma a következő:

css
stíluslapok
directsql
SQL lekérdezések kimeneteinek stíluslapjai
icons
ikonok
js
javascript állományok (prototype könyvtár)
locale
fordítási állományok
scanned
bizonylatokhoz feltöltött dokumentumképek
spool
csoportos nyomtatáshoz használt ideiglenes állományok
sql
adatbázis létrehozásával és frissítésével kapcsolatos SQL parancsok
templates
Latex és Html sablonok,
users
felhasználók konfigurációs állományai.

A gyökérkönyvtárban lévő functions.conf-ban lehet különböző kapcsolókkal szabályozni a szoftver alapvető működését (pl. \$nonnegativestock_true = 1 nem engedi mínuszba a raktárat), valamint az \$maccess változóval lehet modulokat ki és be kapcsolni.

Az sql-ledger.conf-ban lehet különböző könyvtárútvonalakat illetve nyomtatóbeállításokat változtatni. A menu.ini-ben lehet a menüket bővíteni.

2.2. Példák

2.2.1. Felület testreszabása

Ha elkezdünk dolgozni egy szoftverrel, a felület testre szabhatósága nagyon fontos a következő okok miatt:

- Olyan betűméretek illetve színösszeállítást érdemes használni, amely *legkevésbé fárasztja a szemünket*
- Több adatbázis használata esetén különböző színekkel tudjuk megkülönböztetni őket
- Végül több felhasználó esetén felhasználónként lehet különbséget tenni a belépések között.

A Ledger különálló CSS állományban biztosítja a testreszabhatóság lehetőségét, amelyet a felhasználói felületen is szerkeszthetünk.

Ha a *Stíluslap* menüpontra rákattintunk, akkor a felület kinézetéért felelős *ledger.css* állomány tartalmát láthatjuk a képernyőn. A *Módosítás* gomb megnyomásával szerkeszthető formában látjuk a tartalmat, amelyet módosíthatunk és elmenthetünk (Linux operációs rendszeren a *css* könyvtárat és a tartalmát írhatóvá kell tenni az *Apache* felhasználó számára).

Lássuk, hogyan tudjuk egyszerűen befolyásolni a kinézetet. A háttérszín jelenleg fehér:

```
body.menu {  
  background-color: rgb(247, 247, 246);  
  font-family: Verdana, Arial, Helvetica;  
  font-size: 7.5pt;  
  color: black;  
}
```

változtassuk szürkére:

```
body {  
background-color: grey;  
.....</pre>
```

A *lenyíló ablakok* és a *szöveges mezők* betűméretét is befolyásolhatjuk, ha módosítjuk a betű méretét például:

```
input, textarea, select {  
  font: Arial, Helvetica, sans-serif;  
  border: 1px solid #999999;  
  background-color: #CCCCCC;  
  margin-top: 1px;  
  color: black;  
  padding: 0 2px;  
  font-size: 9pt;  
}
```

A példákban igencsak az egyszerű módosításokra volt szó, a *CSS stíluslapokban* rejtőző lehetőségeknek csak a felhasználó tudása szab határt.

Az interneten számos CSS stíluslapokkal foglalkozó leírás, cikk, példa található, mint például:

- www.csszengarden.com
- www.w3schools.com/css/default.asp
- www.westciv.com/style_master/academy/css_tutorial/

2.2.2. Korlátozott hatáskörű módosítások

Minden cég előbb utóbb beleütközik abba a problémába, hogy bár a funkciók jól illeszkednek a céges folyamatokra, mégis kisebb módosítás kényelmesebbé, hasznosabbá tenné a szoftvert. Az olyan módosításokat, amelyeket a szoftver felhasználói felületéről nem tudunk beállítani, kénytelenek vagyunk a *forráskód módosításával* elvégezni. Mielőtt azonban saját magunk 'belenyúlnánk' a forráskódba, a következő szempontokat érdemes figyelembe venni:

- A forráskód módosítása a Felelősségvállalási nyilatkozat érvényességének elvesztésével jár. (ha van)
- Tényleg értünk a *perl* programozási nyelvhez? Nem lenne érdemesebb szakemberre bízni?
- A módosítás minden felhasználóra érvényes legyen, vagy csak bizonyos felhasználókra?
- Ha újabb verzió jön, hogyan viszem tovább a módosítást?

Tegyük fel, hogy úgy egyeztünk meg a könyvelőnkkel, hogy az általunk rögzített költségszámlákat bizonyos időszakonként ellenőrzi. A szoftver kétféle beállítási lehetőséget biztosít arra, hogy bizonyos felhasználók csak megtekinthessék, de ne módosíthassák (módosítás, törlés) a rögzített bizonylatokat.

Nekünk ez nem elég, mert mi azt szeretnénk, hogy a könyvelőnk módosíthassa, de *ne törölhesse* a bizonylatot még véletlenül sem. Íme egy példa, amikor az üzleti folyamatainkra nem illeszkedik a szoftver logikája, tehát a forráskódot módosítani kell. A megoldási terv egyszerű: ha a könyvelő nem látja a *Törlés* gombot, akkor nem tudja megnyomni még véletlenül sem. Viszont a többi felhasználónak szüksége van a *Törlés* gombra, tehát a módosítás csak akkor legyen érvényben, ha a könyvelőnk lép be a rendszerbe.

A forráskód figyelmes megvizsgálása után rájöhethetünk, hogy a kérdéses változtatást a *bin/mozilla* könyvtár *ap.pl* állományában kell elvégezni. Az *ap.pl* állomány tartalmazza a *form_footer* funkciót, amely felelős az űrlapok alján található gombok megjelenítéséért.

A Törlés gomb megjelenítését a következő kódrészlet végzi:

```
if (!$form->{locked}) {  
    print qq|  
    <input class=submit type=submit name=action onclick="return checkform()" value="|.  
$locale->text('Post').qq|">  
    <input class=submit type=submit name=action value="|. $locale->text('Delete').qq|">  
|;  
}
```

Tehát ha a felhasználó jogosult rá, a program megjeleníti a *Rögzítés* és a *Törlés* gombot. A kódrészletből tehát azt a sort kell kitörölni, amely a *Törlés* gombot jeleníti meg.

Ha kitöröljük *Törlés* gombot megjelenítésért felelős kódrészletet a *bin/mozilla/ap.pl* állományból, a hatás globális lesz, tehát egyetlen felhasználó sem fogja látni a *Törlés* gombot. Hogyan szűkítjük le a módosítás hatását a könyvelőnkre, aki a *konyvelo* felhasználó névvel szokott belépni a rendszerünkbe?

A szoftver erre elegáns megoldást kínál a fejlesztőknek. A módosítást ugyanabba a könyvtárba *konyvelo_ap.pl* néven kell elmenteni, amiből a szoftver tudni fogja, hogy a módosítást csak a *konyvelo* felhasználó belépése esetén töltsse be. Ennek a lehetőségnek köszönhetően több különböző módosított verziók is futhatnak egy telepítésen belül, felhasználók neveihez kötöten.

Maradt még egy problémánk: hogyan őrizzük meg saját változtatásunkat, ha egy friss kiadást másolunk rá az általunk használt verzióra? Mivel a változtatásunk a *konyvelo_ap.pl*-ben van, a friss kiadásban lévő *ap.pl* ezt nem írja felül. Csupán azzal kell törödnünk, hogy mi a *különbség* a régi és az új *ap.pl* között. A különbséget lehet (de nem kötelező) a *konyvelo_ap.pl* állományba átvinni, attól függően hogy mennyire fontos nekünk ez a különbség (pl. ha egy hibát javít, akkor érdemes a könyvelőnk számára is elérhetővé tenni ezt a hibajavítást).

A különbségek kiderítésére és egy másik állományba való beillesztésére nagyon jó programok állnak rendelkezésre a fejlesztők számára.

Természetesen ha globális változtatásunk van és nem akarjuk, hogy a frissítés felülírja a változtatásunkat, nem kell minden egyes felhasználóhoz (*user1_ap.pl*, *user2_ap.pl* stb.) állományt létrehozni. Erre szolgál a *custom_* előtag, tehát a *custom_ap.pl* minden felhasználó számára érvényes módosítást eredményez a Ledger működésében. A *custom_* előtag nagyon jól használható egy olyan módosítás tárolására, amely az adott cégre specifikus, és nincs esélye hogy bekerüljön a a szabadon letölthető verzióba.

A *custom_* előtag jó szolgálatot tehet az adatbázis szerkezetének módosításakor is. Ha például a vevő törzsadatait egy plusz mezővel egészítjük ki (a cég honlap címe), akkor a *customer* táblát is ki kell egészíteni egy további oszloppal. A biztonsági mentést végző funkciónak azonban tudnia kell arról, hogy a *customer* tábla változott, egyébként nem menti el helyesen az adatokat. Ennek érdekében el kell helyeznünk egy *Pg-custom_tables.sql* állományt az *sql/* könyvtárban amelynek tartalma a következő lehet a jelen példa alapján:

```
ALTER TABLE customer ADD COLUMN url TEXT;
```

2.2.3. Menük testreszabása

Különböző alkalmazások használata során megszokhattuk, hogy a menü saját igényeink szerint bővíthető, módosítható.

Nincs ez másképp a Ledger-rel sem, ahol a *menu.ini* állomány módosításával bővíthetjük menüinket. Például, a devizás tranzakciók rögzítése során gyakran látogatunk el az MNB oldalára, ahol deviza középárfolyamokat teszik rendszeresen közzé. Jelentősen könnyítene a helyzetünkön, ha egy menüpont mutatna a megfelelő oldalra, és egy klikkelésre megjelenne az összes deviza. A fenti cél eléréshez csupán annyit kell tennünk, hogy bővítjük a *menu.ini* állományt a következő sorokkal:

```
[MNB deviza]  
module=null  
href=http://www.mnb.hu/arfolyamok
```

Nézzünk egy másik példát.

Nem csak teljesen új menüpontokat hozhatunk létre, hanem meglévő menüpontjainkat újrendezhetjük, csoportosíthatjuk. Például hozzunk létre egy *Kedvencek* főmenüt, ahol a leggyakrabban használt lekérdezéseinket sorolhatjuk fel:

```
[Kedvencek]
module=menu.pl
action=acc_menu
target=acc_menu
submenu=1
```

```
[Kedvencek--Vevőszámlák listája]
module=ar.pl
action=search
nextsub=ar_transactions
```

```
[Kedvencek--Vevő lejáratilista]
module=rp.pl
action=report
report=ar_aging
```

Tehát meg kell keresni annak a menünek a leírását a *menu.ini* állományon belül, amelyet a *Kedvencek* alá szeretnénk sorolni.

2.2.4. Számla bővítése új adatokkal

Előfordulhat, hogy olyan adatot kell megjelenítenünk, amely alapesetben nem szerepel a számlán. Sajnos az űrlapon semmilyen formában nem szerepel a *contact* változó, ezért nekünk kell „rávarázsolni” a számla űrlapjára. Első feladat annak az adatbázisparancsnak a megkeresése, amely a vevő törzsadatainak kinyerését végzi.

A *menu.ini*-ben láthatjuk, hogy az *Új számla kiállítása* menüpont az *is.pl* szkript *add* funkcióját hívja meg:

```
[AR]
```

```
[AR--Sales Invoice]
module=is.pl
action=add
type=invoice
```

A *bin/mozilla/is.pl add* funkciója meghívja az *invoice_links* funkciót, amely szintén meghívja a *IS->get_customer* funkciót. A *get_customer* funkció felelős a vevőtörzsadatok kinyeréséért, amelyet az *SL/IS.pm* modulban találunk meg:

```
# get customer
$query = qq|SELECT c.name AS customer, c.discount, c.creditlimit, c.terms,
c.shipvia, c.shippingpoint,
        c.email, c.cc, c.bcc, c.taxincluded,
        c.address1, c.address2, c.city, c.state,
        c.zipcode, c.country, c.curr AS currency, c.language_code,
        $duedate AS duedate, c.notes AS customernotes, c.intnotes AS
customerintnotes, c.duebase,
        b.discount AS tradediscount, b.description AS business,
        e.name AS employee, e.id AS employee_id
        , CASE WHEN b.tdij1 IS NOT NULL OR b.tdij2 IS NOT NULL THEN 1 ELSE
0 END AS tdij_van
        FROM customer c
        LEFT JOIN business b ON (b.id = c.business_id)
        LEFT JOIN employee e ON (e.id = c.employee_id)
        WHERE c.id = $form->{customer_id}|;
```

Ezt az SQL lekérdezést kell módosítani úgy, hogy a *c.contact* mezőt is kérjük:

```
# get customer
```

```
# get customer
$query = qq|SELECT c.name AS customer, c.discount, c.creditlimit, c.terms,
c.shipvia, c.shippingpoint, c.contact, .....
```

Néhány sorral arrébb láthatjuk, hogy az összes adat bekerül a *form* változóba:

```
map { $form->{$_} = $ref->{$_} } keys %$ref;
```

Tehát ennél a pontnál a *\$form->{contact}* nevű változóban van a kívánt adat.

Most már csak meg kell jeleníteni az űrlapon, hogy a *LaTeX* sablon megkapja a változó értékét.

A *bin/mozilla/is.pl form_header* funkciója felelős az űrlap első részének kiírásáért, tehát érdemes ebben a szekcióban megoldani a *\$form->{contact}* változó értékének megjelenítését, például így:

```
sub form_header {
...
<input type=hidden name=whded value=$form->{whded}>
<input type=hidden name=inwh value=$form->{inwh}>
<input type=hidden name=contact value="$form->{contact}">
....
|;
...
}
```

Most már az űrlapon van a kívánt érték, már csak a *LaTeX* sablonban kell elhelyezni a *<%contact%>* értéket, a *LaTeX* szabályai szerint.

2.2.5. DirektSQL lekérdezések

A DirektSQL modul segítségével saját lekérdezéseket írhatunk az *SQL* nyelv segítségével, amelyeket elmenthetünk és később újra lefuttathatunk.

A lekérdezések eredményét a mellékelt sablonok segítségével böngészőben, vagy *xls* formátumban táblázatkezelő segítségével jeleníthetünk meg.

Ahhoz, hogy kinyerhessük a minket érdeklő adatokat, ismernünk kell a táblák neveit, valamint a táblában található mezőket.

A táblák neveit például a következő *SQL* parancs begépelésével és futtatásával nyerhetjük ki:

```
SELECT relname FROM pg_class WHERE relkind='r' AND relowner>=100;
```

amelynek az eredménye a minket érdeklő táblák neveinek a listája:

```
defaults
tax
dpt_trans
business
sic
warehouse
inventory
yearend
partscustomer
serialnum
cogs
ds
acc_trans
ar
assembly
chart
customer
customertax
employee
exchangerate
gifi
```

```
gl
invoice
makemodel
oe
orderitems
parts
partstax
vendor
vendortax
shipto
project
partsgroup
status
department
partsvendor
pricegroup
audittrail
translation
language
ap
```

A *vendor* tábla mezőit például a következő egyszerű SQL utasítással kapjuk meg (adatokkal együtt):

```
SELECT * FROM vendor;
```

Most, hogy már ismerjük a táblák és mezők neveit, el kezdhetjük gyártani a saját lekérdezéseinket. Lássuk például azt a feladatot, amelyben azt szeretnénk megtudni, mely felhasználóhoz, mely raktárak vannak rendelve!

Az alkalmazottak adatai az *employee* táblában vannak, míg a raktárnevek a *warehouse* táblában. A következő SQL paranccsal a két táblát egyesítve kapjuk meg a minket érdeklő adatokat:

```
SELECT e.name AS felhasználó, w.description AS raktár FROM employee e, warehouse
w WHERE e.warehouse_id = w.id;
```

5 legnagyobb tartozással rendelkező vevők listája:

```
SELECT name, sum(amount)-sum(paid) AS tartozas FROM ar, customer c
WHERE ar.customer_id = c.id GROUP BY name ORDER BY tartozas DESC LIMIT 5;
```

Vevők, amelyek több mint 100.000 Ft-tal tartoznak:

```
SELECT name, sum(amount)-sum(paid) AS tartozas FROM ar, customer c
WHERE ar.customer_id = c.id GROUP BY name HAVING sum(amount)-sum(paid) >
100000;
```

Vevők nevei és a hozzá tartó sales-ek:

```
SELECT c.name AS vevő, e.name AS sales FROM customer c, employee e WHERE
c.employee_id = e.id;
```

A fenti SQL parancsokat egyszerűen be kell másolni *Lekérdezés* mezőbe, ki kell választani a sablon típusát, majd menteni és/vagy futtatni lehet.

2.2.6. További mezők beépítése a lekérdezésekben

Ez a testreszabási igény akkor merül fel, ha olyan információt is szeretnénk látni a lekérdezés oszlopaként, ami jelenleg nem elérhető.

Vegyük például a *Raktár/Cikkek/Cikkek* lekérdezést, ahol a cikkek törzsadatában lévő *Megjegyzés* oszlopot nem lehet kijelölni/megjeleníteni. A *perl* forráskódot több helyen kell módosítani, a fejlesztési lépések a

következők:

1. A szűkítő űrlapon a *Megjegyzés* kapcsoló megjelenítése, hogy kiválasztható legyen
2. A cikkek törzsadatainak kinyerését végző SQL parancs bővítése a *notes* mezővel, és a megjelenítendő lista kibővítése a *Megjegyzés* oszloppal

1. A cikkek törzsadatainak kinyerését végző SQL parancs bővítése a *notes* mezővel

A szűkítő űrlap megjelenítését a *menu.ini* alapján a *bin/mozilla/ic.pl* állomány *search* funkciója végzi. Keressük meg azt a részt, ahol a kapcsolókat írja ki a szkript és egészítsük ki a a *Megjegyzés* kapcsolóval:

```
<tr>
  <td><input name=l_image class=checkbox type=checkbox
value=Y>&nbsp;|. $locale->text ('Image') .qq|</td>
  <td><input name=l_drawing class=checkbox type=checkbox
value=Y>&nbsp;|. $locale->text ('Drawing') .qq|</td>
  <td><input name=l_microfiche class=checkbox type=checkbox
value=Y>&nbsp;|. $locale->text ('Microfiche') .qq|</td>
  <td><input name=l_notes class=checkbox type=checkbox value=Y>&nbsp;|. $locale-
>text ('Notes') .qq|</td>
</tr>
```

2. A cikkek törzsadatainak kinyerését végző SQL parancs bővítése a *notes* mezővel, és a megjelenítendő lista kibővítése a *Megjegyzés* oszloppal

Amikor megnyomjuk a *Folytatás* gombot, akkor a *bin/mozilla/ic.pl* szkript fut le, ahol a *generate_report* funkció végzi el a lista készítését. Első lépésként hozzuk tudomására a programnak, hogy új oszlop került be:

```
@columns = $form->sort_columns(qw(partnumber description assemblypartnumber
partsgroup make model bin onhand rop unit listprice linetotallistprice sellprice
linetotalsellprice avprice linetotalavprice lastcost linetotallastcost
priceupdate weight image drawing microfiche invnumber ordnumber quonumber name
serialnumber reqdate balance business tdi1 tdi1_total tdi2 tdi2_total
projectnumber notes));
```

Az *IC->all_parts(%myconfig, %\$form)* parancs végzi az adatok kinyerését az adatbázisból, tehát az *SL/IC.pm all_parts* funkciójában kell bővíteni az SQL lekérdezést:

```
my $flds = qq|p.id, p.partnumber, p.description, p.onhand, p.unit, p.notes,
      p.bin, p.sellprice, p.listprice, p.avprice, p.lastcost, p.rop,
      p.weight, p.priceupdate, p.image, p.drawing, p.microfiche,
      p.assembly, pg.partsgroup, pr.projectnumber $makemodelflds
      $assemblyflds
|;
```

Most már a *notes* mező értékét is kinyerjük.

Adjuk meg a megjelenítendő listában az oszlop fejlécét:

```
$column_header{image} = qq|<th class=listheading nowrap>|. $locale-
>text ('Image') .qq|</a></th>|;
$column_header{drawing} = qq|<th nowrap><a class=listheading
href=$href&sort=drawing>|. $locale->text ('Drawing') .qq|</a></th>|;
$column_header{microfiche} = qq|<th nowrap><a class=listheading
href=$href&sort=microfiche>|. $locale->text ('Microfiche') .qq|</a></th>|;
$column_header{notes} = qq|<th nowrap><a class=listheading
href=$href&sort=notes>|. $locale->text ('Notes') .qq|</a></th>|;
```

Majd az adatokat tartalmazó részt is illesszük be:

```
$column_data{image} = ($ref->{image}) ? "<td><a href=$ref->{image}><img  
src=$ref->{image} height=32 border=0></a></td>" : "<td>&nbsp;</td>";  
$column_data{drawing} = ($ref->{drawing}) ? "<td><a href=$ref->  
{drawing}>$ref->{drawing}</a></td>" : "<td>&nbsp;</td>";  
$column_data{microfiche} = ($ref->{microfiche}) ? "<td><a href=$ref->  
{microfiche}>$ref->{microfiche}</a></td>" : "<td>&nbsp;</td>";  
$column_data{notes} = "<td>$ref->{notes}&nbsp;</td>";
```

A fenti módosítások hatására --a *Megjegyzés* kapcsoló bejelölése után-- a listában megjelenik a *Megjegyzés* oszlop, sőt újra is rendezhetjük a listánkat a *Megjegyzés* fejlécre kattintva.

2.2.8. Parancssoros hozzáférés

A Ledger-ben kétféleképpen indíthatunk valamilyen tranzakciót:

- linkre kattintással
- gombra kattintással

Ezeket a tranzakciókat parancssorosan is kiválthatjuk.

Vegyük például azt a helyzetet, hogy a számlák kiállítása egy másik ügyviteli szoftverben történik, de a könyvelése már a Ledger-ben. Szeretnénk, ha nem kellene manuálisan újra bevinni a kinyomtatott számlákat; ehelyett szeretnénk, ha a számlát készítő program bemásolná egyből az adatokat Ledger adatbázisába.

Előbb vizsgáljuk meg, milyen értékeket küldd tovább a *Bevételek/Új átutalásos rögzítése* űrlapon lévő *Rögzítés* gomb (pl. a html oldal forráskódjának a vizsgálatával), és máris megszerkeszthetjük azt a parancssort, amely beküldi az adatokat az adatbázisba.

Például a *1002* számú *Vevő 1. Kft*-nek készült bruttó *15.600 Ft*-os számla parancssora:

```
<gyökér>/ar.pl  
"AR=3110&AR_amount_1=9110&AR_base_1=0.25&AR_tax_4671=4671&action=rögzítés&amount  
_1 = 12.480&crdate=2010-09-15&currency=HUF&customer=Vevő%201.%20Kft.--  
10311&customer_id=10311&duedate=2010-09-15&employee=08%20Devel--10153  
&fxgain_accno=9700&fxloss_accno=8700&invnumber=1002&login=08devel&path=bin/mozil  
la&rowcount=2&script=ar.pl&password=&tax_4671=3.120&taxaccounts=4671&transdate=2  
010-09-15&defaultcurrency=HUF"
```

Tehát ezt a parancssort futtatva a Ledger megkerülésével bekerül a számla, úgy mintha kézzel rögzítettük volna.

3. Távügyvitel Kft. által nyújtott szolgáltatások

Cégünk a következő szolgáltatást nyújtja :

- Fórum
A felhasználók egymás között megbeszélhetik problémáikat, kérdéseket tehetnek fel:
<http://www.hup.hu/ledger-forum>
- Előadás
Egy 2-3 órás INGYENES előadáson, amelyet cégünk havonta rendez, átfogó képet kaphat a Ledger programról, valamint kérdéseit felteheti közvetlenül nekünk.
- Ledger KLUB
PDF formátumú felhasználói kézikönyv
Hasznos tippeket tartalmazó kézikönyv
Példákkal illusztrált útmutató a helyes könyvelési technikákhoz
Példákkal illusztrált útmutató a program saját igényekhez történő alakításához
A legújabb prémium verzió
Felelősségvállalási Nyilatkozat (megfelelőség a jogszabályoknak)
Extra segédprogramok
Bejelentett programhibák azonnali javítása
Jogszabálykövetés
Hírlevél
Félévente Klubnap előadásokkal
20% kedvezmény a hivatalos árlistából
Egyszerűbb kérdésekre válasz
- Bevezetési támogatás
Az első perctől kezdve ott vagyunk és segítünk a bevezetésben, amely során a következő feladatokat látjuk el:
-Telepítés
-Törzsadatok feltöltése
-az ügyviteli lépések kialakítása a cég folyamatai alapján
-betanítás
-kisebbségi módosítás elvégzése: logó a számlán, alapbeállítások megváltoztatása, sablonok kisebb módosításai stb.
- Óradíjas támogatás
A tanácsadás lehet személyes, telefonos vagy email-en keresztül.
- Programozás egyedi funkciók kialakításához
Amennyiben olyan funkcióra van szüksége, amely az aktuális verzióban nem található meg, egyedi árajánlat alapján a szükséges programozást elvégezzük Önnek.
- A Ledger bérlése
A többszörösen védett és karbantartott szerverünkön futó Ledger-t folyamatosan használhatja havi bérleti díjért, amely a következő előnyökkel jár:
-Nem kell telepíteni semmilyen szoftvert, ezáltal nem kell bajlódni üzemeltetési problémákkal; a használathoz mindössze egy böngészőre és internetkapcsolatra van szüksége
-Csak azokat a funkciókat használhatja, amelyre valójában szüksége van, így csökkentheti költségeit, valamint saját illetve kollégái munkaidejét
-Nem kell törődnie adatai mentésével, illetve a biztonsági kérdésekkel
-Állandóan élvezheti a szoftver frissítéséből adódó könnyebb használatot, illetve kibővített funkciókat
-A kérdéseire mindig választ kap
- Képviseleti könyvelés
Amennyiben Ön vállalja a bejövő (költség) számlák, bankkivonatok rögzítését, akkor minden bizonylat (bevétel- és költség számlák, bankkivonatok) Önnél maradhat. Önnek csak a bevételek elkészítésével és az APEH felé történő képviselettel kell megbízni a könyvelőjét. Így jelentősen csökkentheti könyvelési költségeit, ráadásul naprakész információkkal rendelkezhet cége üzleti helyzetét illetően.
Ezen képviseleti könyvelési szolgáltatással minket is megbízhat, amelyet az alvállalkozó *könyvelőirodánkkal* együttműködésben nyújtunk.



<http://www.tavugyvitel.hu>
tel: (061) 266-1039