

# Az Sql-ledger integrált ügyviteli rendszer

Kabai József

2003. szeptember 22.

## Kivonat

Az Sql-ledger [1] egy olyan integrált ügyviteli rendszer, amellyel cégek könnyen és rugalmasan végezhetik el a megrendelések, számlák feldolgozását illetve azok egyidejű kettős könyvelését, valamint figyelemmel kísérhetik készletmozgásaikat. A program felhasználása rendkívül sokrétű, működik Windows, Linux, Mac operációs rendszerek alatt, valamint a legfejlettebb adatbázismotorokat (Postgres, Oracle) használja. Használható önálló gépeken, intraneten, esetleg igény szerint interneten is.

## Tartalomjegyzék

<b>1. Az általános jelenség</b>	<b>2</b>
1.1. Elszigetelt alkalmazások . . . . .	2
1.2. Integrált rendszerek . . . . .	2
1.3. Nincs rá keret . . . . .	2
<b>2. Reális alternatíva: szabad szofver</b>	<b>2</b>
2.1. Szabadság, szeretem . . . . .	2
<b>3. Az Sql-ledger könnyen módosítható</b>	<b>3</b>
3.1. Jó tervezés . . . . .	3
3.2. Módosított állományok . . . . .	3
3.3. Új állományok létrehozása . . . . .	3
<b>4. Választható felhasználói felületek</b>	<b>4</b>
4.1. Kliensprogram: melyiket válasszam? . . . . .	4
4.2. Más kliensprogramok választása . . . . .	4
4.3. Együtműködés más alkalmazásokkal . . . . .	4
<b>5. A felhasználói kör</b>	<b>5</b>
5.1. Klasszikus felhasználók . . . . .	5
5.2. Könyvelőcégek . . . . .	5
5.3. Rendelési rendszer . . . . .	5
<b>6. A magyar verzió</b>	<b>5</b>
6.1. A program honosítása . . . . .	5
6.2. Ki vállal felelősséget? . . . . .	6
<b>7. Mít hoz a jövő?</b>	<b>6</b>

## 1. Az általános jelenség

### 1.1. Elszigetelt alkalmazások

A vállalkozások működésük során folyamatosan szembesülnek azzal a problémával, hogyan egységesítsék a különböző alkalmazásokon futó adataikat, amelyek szigetként különülnek el a többi adatbázistól. A szállító/vevőszámla-nyilvántartás a pénzügyi osztályon szerepel valamelyik népszerű táblázatkezelőben. Ezeket az adatokat a könyvelés duplikálja a számlák feldolgozásakor, és a raktár is használ (évek óta) valami „ideiglenes” megoldást a raktárnyilvántartásra és a számlázásra. Ezekből az adatokból fáradtságos munkával lehet csak a vezetők számára értékelhető, és a valósághoz közeli jelentéseket létrehozni. Hogyan lehet a legfontosabb információkat, mint például az árrélelemzést elvégezni, amikor a raktármozgások rögzítése elkülönül az árbevétel-nyilvántartástól?

### 1.2. Integrált rendszerek

Adódik a megoldás: a különböző osztályok ugyanabból az adatbázisból dolgozzanak, amelyet az integrált rendszerek biztosítanak. A legismertebb a nagyvállalatok részére készült SAP, amely a piac úttörője, és a legnagyobb piaci szereplője is egyben. Ezek a rendszerek már olyan fejlettségűek, hogy lehetőséget adnak nemcsak a múltbeli teljesítmény értékeléséhez, hanem a vállalati tervezésben is segítséget nyújtanak (ERP rendszerek: Enterprise Resource Planning). Tipikus példa egy termelő vállalat, amely számára létfontosságú a jelenlegi és a jövőbeni megrendelések alapján a termeléshez szükséges anyagbeszerzés ütemezését megtervezni.

### 1.3. Nincs rá keret

Az integrált rendszerek hasznosak, csak drágák, ráadásul a betanítást és a testreszabást is sok pénzért lehet csak megkapni. A rendszerek döntő többsége zárt forráskódú, ezért a felhasználó a program fejlesztőjéhez van láncolva: nincs alternatíva, nincs versenyhelyzet. Mit tehet az ember, ha nincs elég pénze és nem akar kiszolgáltatott helyzetbe kerülni? Körülnéz a *szabad világban*.

## 2. Reális alternatíva: szabad szofver

### 2.1. Szabadság, szeretem

A szabad szoftverek fejlesztése ebben a témában csak az utóbbi években gyorsult fel, és a SourceForge [2] oldalait böngészve szép számmal találhatók szabad ügyviteli rendszerek. Jellemzően egyedi megbízásokból fejlődtek szabaddá ezek a megoldások. Mit jelent számomra tágabb értelemben a szabadság?

- használatához szükséges környezet is ingyenes (programnyelv, adatbázis stb.);
- szabadon letölthető: ingyenes;
- szabadon módosítható, nyílt forráskódú, amely könnyen átlátható, logikus;
- szabadon eldönthetem, hogy energiát és tanulást befektetve magam módosítom az igényeim szerint, vagy mással végeztetem el;

- szabadon eldönthetem melyik operációs rendszeren használom.

Az Sql-ledger maradéktalanul megfelel a fenti elvárásoknak:

- Az adatok a legjobb GPL licenstes adatbázisban a PostgreSQL-ben tárolódnak (a fizetés Oracle is választható);
- A kliens bármelyik böngésző (Internet Explorer, Mozilla, Lynx, Links, W3M) lehet. Javascript kódot nem használ, kompatibilitási problémák nem jelentkeznek;
- az alkalmazás Perl nyelven íródott;
- Apache webszerver kommunikál az adatbázis és a böngésző között;
- Linux, Windows, Macintosh rendszereken is fut.

### 3. Az Sql-ledger könnyen módosítható

#### 3.1. Jó tervezés

Ha valaki úgy dönt, hogy szabad szoftvert ír, az óriási kitarás mellett arra is szükség van, hogy a program alapjaiban jól megtervezett, és a kódja jól olvasható legyen: számos példát láthatunk mennyi fejlesztés jutott zsákutcába a programkód állandó foltozása miatt. Az ügyviteli rendszerek tipikusan olyan szoftverek, amelyeket gyakran kell testre szabni a különböző vállalati igényekből fakadóan. Egy jól olvasható, és érthető kód jelentős mértékben megkönnyíti a fejlesztők munkáját.

#### 3.2. Módosított állományok

Az Sql-ledger készítői a fejlesztőket is megcélözva óriási hangsúlyt fektettek a módosíthatóságra. A módosított Perl állományt két szinten lehet érvényesíteni: globális szinten, valamint felhasználói szinten, amelyet a Perl állomány nevének változtatásával lehet szabályozni. Például ha egy új dátummezőt illesztünk be a vevőszámlák feldolgozása során, a változtatás globális szintű lesz, így az állomány nevét *custom\_ar.pl* névre kell változtatni. Egy másik esetben például dönthet egy könyvelőiroda, hogy a pénzügyi jelentések megtekintésekor az ügyfél még véletlenül se módosíthasson, törölhessen adatot (hiszen az a könyvelők feladata). Tehát a módosítást tiltó állomány neve (az ügyfél bejelentkező neve: *client*) a *client\_ar.pl* lesz, amely tehát csupán a *client* bejelentkezés esetén fut le. A módosított nevű állományok így nem vesznek el új verziók telepítésekor sem, hiszen csak az eredeti *ar.pl* állomány íródik felül. Természetesen az új verzió állományát ismét módosítani kell saját kódrészeinkkel, ha élvezni akarjuk az előnyeit a módosításokkal együtt.

#### 3.3. Új állományok létrehozása

A módosíthatóságon túl, lehetőség van új menüpontok létrehozására és ezek menürendszerhez történő könnyű csatolására. Például, ha egy határidőnaplót csatolunk a rendszerhez, az alapmodulokhoz hasonlóan létre kell hoznunk egy szimbolikus linket a gyökérkönyvtárban, a megjelenítésért felelős állományt a *bin/mozilla* – vagy *lynx* böngésző esetén *bin/lynx* – könyvtárba, és az adatbázis-műveletekért felelős Perl modult az *SL* könyvtárba.

A fentiekre konkrét példák találhatók a következő címen:

```
http://www.sql-ledger.org/cgi-bin/nav.pl?page=misc/api.html
&title=Customization
```

## 4. Választható felhasználói felületek

### 4.1. Kliensprogram: melyiket válasszam?

A tervezés során valószínűleg a legjobb választás a szabványos böngésző melletti döntés volt. Böngésző bármelyik gépen található; nem kell vesződni a kliensprogram telepítésével, illetve karbantartásával. Természetesen az alapverzióban nyomokban sem található a webprogramozók életét megkeserítő javascript kód, így minden böngészőben kiválóan működik.

A gyors adatbevitelt megnehezítő egérhasználatot grafikus böngészőkben is ki lehet váltani egyedi testreszabás során, a felhasználó kérésére. Erre az *accesskey* eszköz használható, amely lehetőséget ad egérműveletek helyettesítésére billentyűkombinációkkal, mint például a `<a href= ref.html ACCESSKEY="r">ref</a>` esetében, ahol az *ALT+r* lenyomásával aktívvá tehetjük a hivatkozást.

### 4.2. Más kliensprogramok választása

Ha valakinek nem tetszik a böngészős környezet, akkor a program rugalmasságának köszönhetően más felhasználói felület is húzható a program elé, mivel a folyamatok parancssoros üzemmódban is működnek a változók paraméterként való átadásával. A következő példa a szállítók listájának keresőfelületét jeleníti meg parancssoroson:

```
$ cd /usr/local/sql-ledger
$ ./ct.pl "login=name&path=bin/mozilla&password=&action=search&db=vendor"
```

A fenti példához hasonlóan, ismétlődő könyvelési tételek sokkal gyorsabb bevitelét is el lehet érni: táblázatkezelőben beírt tételeket (szöveges állományként) egy egyszerű szkripttel be lehet tölteni.

### 4.3. Együtműködés más alkalmazásokkal

A testreszabás nem feltétlenül történhet az alapprogram szabályai alapján. A *PostgreSQL* adatbázismotor közvetlenül is elérhető, amely lehetőséget ad nagy tömegű adat közvetlen bevitelére. Egy régi rendszer törzs- és záró adatait (cikktörzsállomány, nyitott vevőszámlák) nagyságrendekkel könnyebb így feltölteni, mint kézzel bepötyögni. *ODBC* adatkapcsolaton keresztül pedig saját egyéni lekérdezéseket írhatunk, amelyek eredményét tovább elemezhetjük kedvenc táblázatkezelőnkben.

Egy hazai nagykereskedelmi cégnél is hasonló megoldás született. A logisztikai osztály dolgozói nehezen váltak volna meg az évek óta használt MS Access felületüktől, így a rendelések feldolgozása, a szállítástervezés, illetve a szállítólevelek kinyomtatása Accessben történik. A megvalósult kiszállításokat az *Sql-ledger ODBC* kapcsolaton keresztül automatikusan megkapja, és inentől kezdve a termék életútját már a pénzügyi osztály kíséri figyelemmel. A feladás a raktárkészletet módosítja, és a megvalósult vevőrendelésekről egy gombnyomásra számla készül.

## 5. A felhasználói kör

### 5.1. Klasszikus felhasználók

Az integrált ügyviteli rendszert gyakorlatilag bármilyen iparágban tevékenykedő cég használhatja, attól függően, mennyi energiát és fejlesztési munkát áldoz a testreszabásra. Mégis a legkevesebb beavatkozással a szolgáltató- és kereskedőcégek tudják az Sql-ledger-t használni. A szoftver logikája a termék/szolgáltatás általános életútját követi:

Értékesítés esetén:

Ajánlat → Vevőrendelés → Kiszállítás → Számlázás → Pénzbefolyás

Beszerezés esetén:

Ajánlatkérés → Beszerzési rendelés → Beszállítás → Beszerzési számla → Pénzki-fizetés

Az alapbeállítás során meg kell határozni a termék/szolgáltatás mozgásainak könyvelési szabályait, így a tranzakciók automatikusan könyvelésre kerülnek. A programot tehát olyan is tudja használni, aki nem ismeri a kettős könyvelés rejtjelmeit.

### 5.2. Könyvelőcégek

A program ideális lehet könyvelőcégek számára is. Korlátlan ügyfél könyvelhető, minden ügyféladatbázishoz korlátlan felhasználó kapcsolható különböző jogosultságokkal. A tranzakció-kezelés megakadályozza az adatvesztést, és a nap bármelyik pillanatában biztonsági másolat készíthető a lekönyvelt tételekről. Megfelelő biztonsági beállítások mellett Interneten keresztül távmunkások alkalmazása is lehetséges, illetve az ügyfelek Interneten keresztül megtekinthetik cégük pénzügyi jelentéseit. Egyedi javascript alkalmazásokkal a könyvelőcég alkalmazottainak munkája automatikusan ellenőrizhető, illetve a hibás könyvelések száma jelentősen csökkenthető.

### 5.3. Rendelési rendszer

A [http://www.sql-ledger.com/cgi-bin/nav.pl?page=feature/nhc.html](http://www.sql-ledger.com/cgi-bin/nav.pl?page=feature/nhc.html&title=Application%20Interface%20Example) címen a program olyan alkalmazása látható, amely során az Interneten keresztül a franchise partnerek továbbítják a rendeléseket a központi szerver felé, ahol a beküldött rendelések alapján a számlázás történik. Hasonló elven egy webáruház mögöttes adminisztrációjára is fel lehet készíteni a programot.

## 6. A magyar verzió

### 6.1. A program honosítása

Egy idegen nyelven fejlesztett szoftver honosításának első lépése a magyarra történő fordítás. A szoftver készítője könnyen használható eszközt kínál a fordításra, csupán

egy hash változóba kell beírni (a *locale/hu/all* állományban) a szó illetve kifejezéspárokat. A *locale/hu/locale.pl* elindításával a szövegpárok a megfelelő helyre kerülnek, és a szoftver közben folyamatosan be tudja olvasni a magyar fordításokat.

Egy könyvelőprogram használatánál a leggyakrabban felmerülő kérdés, hogy vajon megfelel-e a törvényi szabályozásnak. Az erre vonatkozó passzusok a számlázás mikéntjét szabályozzák, szó nincs arról, hogy az APEH-nál regisztráltatni kell a könyveléshez használt szoftvert (vagy hasonló tévhitet). A helyes számlakibocsátás követelményeinek megteremtéséhez – mint például kihagyás és ismétlés nélküli sorszámozás, számla kötelező adatai – módosítani kell a programot, amelyet külön csomagként lehet letölteni az éppen aktuális verzióhoz.

## 6.2. Ki vállal felelősséget?

Nehezebb probléma a törvény által megkövetelt felelősségvállalás kérdése, amelyben a program készítőjének vagy terjesztőjének nyilatkoznia kell a vonatkozó rendeletek betartásáról. Egy szabad forráskódú programnál ez első ránézésre megoldhatatlan feladat, hiszen felelősséget vállalni egy könnyen módosítható programért – enyhén szólva – felelőtlenység.

Mint minden ez sem lehetetlen: felelősséget lehet vállalni egy programért, amelyhez digitális ujjlenyomatot mellékelnek. Mindenféle módosítás a felelősségvállalás elvesztését vonja maga után, amelyet az ujjlenyomat igazol.

## 7. Mit hoz a jövő?

Idehaza még nincs köztudatban, mi is az a szabad szoftver. Néhányan már összefüggésbe hozzák a Linux-szal, de a szabad szoftver mozgalom lényegét kevesen értik, illetve gyanakvással méregetik. A szoftver elterjedését jelentősen elősegítheti a szabad szoftver filozófiájának terjesztése, (amelynek érdekében jó néhány egyesület már évek óta kampányol pl. LME, fsf.hu), mégis a döntő lépés a megfelelő szakembergárda kialakulása lenne. Szükség van (döntően Linux) rendszergazdákra, programozókra, valamint alkalmazásban jártas tanácsadókra, akikre biztonsággal támaszkodhat a felhasználó. Ezeknek az embereknek a megtalálása lesz a közeljövő legfontosabb feladata.

## Hivatkozások

- [1] <http://www.sql-ledger.org/>: a szoftver oldala
- [2] <http://www.sourceforge.net/>: szabad szoftverek gyűjteménye
- [3] <http://www.investor.hu/>: a honosított verzió oldala