

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

NFORMÁCIÓS RENDSZEREK TANSZÉK

Egy lakberendező cég

INFORMÁCIÓS RENDSZERE

(KITERJESZTETT VERZIÓ)

Témavezető: Dr. Nikovits Tibor műszaki tanár Készítette: Ábrahám Róbert Programtervező informatikus (BSc)

Budapest, 2013.

TARTALOMJEGYZÉK

1.Bevezetés	4
1.1.Az elérni kívánt célok	6
1.2.A rendszer felépítése, működése	7
1.2.1.Hatékonyság	10
1.2.2.Kiterjeszthetőség, egyéb jellemzők	11
1.2.3.Alapkoncepció, megkötések	12
1.2.4.Együttműködés más webáruházakkal	14
2.Felhasználói dokumentáció	15
2.1.A vásárlói oldal	15
2.1.1.Termékek megtekintése	17
2.1.2.Fórum használata	19
2.1.3.Bevásárlókosár kezelése és termék megrendelése	20
2.2.Az eladói központ	25
2.2.1.Fiókbeállítások	
2.2.2.Termékek kezelése	30
2.2.2.1.Termékek gyors feltöltése	34
2.2.3.Szállítók kezelése	34
2.2.4.Rendelések kezelése	
2.2.5.Pénzügyi adminisztráció	
2.2.6.Fórum az eladói központban	

2.2.7.Adatok importálása/exportálása	41
2.2.7.1.Grando terméklista letöltése	42
2.2.7.2.Adatforrás (rendelési segédlista) importálása	42
2.2.7.3. Megrendelések importálása/exportálása	44
2.2.8.Különleges akciók	45
3.Fejlesztői dokumentáció	48
3.1.A rendszer telepítése és üzembe helyezése	48
3.1.1.Az adatbázis telepítése	49
3.1.2.A webes felület telepítése, beállítása	51
3.2.A rendszer felépítése	53
3.2.1.Az adatbázis felépítése	53
3.2.1.1.Termékek adatainak kezelése	57
3.2.1.2.Bevásárlókosár és online rendelések kezelése	66
3.2.1.3. Pénzügyi és adminisztrációs adatok kezelése	78
3.2.1.4. Nyilvános adatok és felhasználói függvények	83
3.2.1.5.Egyéb elemek az adatbázisban	99
3.2.2.A webes felület felépítése	107
3.2.2.1.A vásárlói oldal felépítése	110
3.2.2.2.Az eladói központ felépítése	112
3.2.2.3.A definíciós fájl	118
3.2.2.4.Diagramok kirajzolása	125
3.3.A rendszer tesztelése	127

4. Irodalomjegyzék	
4.1.Online segédanyagok	130

1. BEVEZETÉS

Magyarországon az 1990-es évek második felében hódított nagyobb teret az internet, a legtöbb háztartásban ekkorra váltak elérhetővé a különböző weboldalak a világhálón. Hazánkban az internet ekkortájt leginkább egyfajta szórakoztató médiumnak számított, mely a szórakoztatás mellett gyorsan és hatékonyan volt képes interaktívan információkat szolgáltatni. Az egyre szélesebb körű elterjedés magával hozta a kereskedelem fokozatos áthelyeződését a világhálóra, mely mára kezdett igazán látványos méreteket ölteni. Míg a lakosság nagy része korábban a különböző áruházak által terjesztett szórólapokon és újságokon keresztül tájékozódott a termékek aktuális árairól, illetve akcióiról, addig mára már egyre inkább a különböző internetes keresőszolgáltatásokat veszik igénybe a potenciális vásárlók annak érdekében, hogy az általuk megvásárolni kívánt termékekhez minél kedvezőbb áron, minél jobb minőségben, illetve minél gyorsabban és kényelmesebben juthassanak hozzá.

Mostanra a magyar internet előfizetők kb. 80%-a legalább vásárlás előtti tájékozódásra használja az internetet, és folyamatosan növekszik az interneten keresztül leadott rendelések száma.



1. ábra. Magyar online kereskedelmi forgalom 2001 és 2012 között (forrás: KSH)

2011-ben az internetet ténylegesen vásárlásra használó előfizetők száma kb. 13% volt a teljes lakossághoz mérten, azoknak az előfizetőknek száma pedig, akik már vásároltak az interneten, kb. 27% volt.



2. ábra. Internetes vásárlások aránya a teljes lakossághoz mérten (forrás: KSH)

Az online megrendelt termékeket alapvetően kétféleképpen vehetjük át: személyesen vagy futárszolgálattal kiszállíttatjuk. A futárszolgálat igénybevételének lehetősége kényelmessé teszi a vásárlást, mert nem kell messzire mennünk lakásunktól, hogy lebonyolítsunk egy komplett bevásárlást. Ennek köszönhetően nagy méretű és súlyú termékek (pl. komplett hálószoba szettek) szállíttatása nem igényel különösebb szervezést, mindent otthonról elintézhetünk néhány webes űrlap kitöltésével, ami nekünk rengeteg időt spórol meg, az áruházaknak pedig több milliós plusz bevételt jelenthet havonta a nagyobb érdeklődőszám miatt.

Mindemellett az is tény, hogy Magyarországon még nem annyira elterjedt a futárszolgálat igénybe vétele, mint külföldön. Ez egyrészt betudható a magyar fogyasztók árérzékenységének, mivel sok termék házhoz szállíttatása esetén a szállítási költség – ami a fogyasztót terheli – néha bizony nem olcsó, másrészről pedig hazánkban még nem alakult ki megfelelő bizalom a szállítócégekkel kapcsolatban, mivel az e-kereskedelem még viszonylag új

dolognak számít. Éppen ezért törekednünk kell a potenciális vásárlók megfelelő tájékoztatására, és biztosítanunk kell számukra olyan felületet, amin keresztül feltehetik kérdéseiket az eladók felé, és természetesen az online értékesítéseket kezelő információs rendszernek biztonságosan és bizalmasan kell kezelnie a rendelkezésére bocsátott adatokat. Ehhez általában a népszerű, jól bevált adatbázis kezelő rendszereket használjuk, ezekből többféle van, az adott probléma mérete és összetettsége dönti el, hogy melyiket érdemes használni.

1.1. AZ ELÉRNI KÍVÁNT CÉLOK

Szakdolgozatomban egy komplett webáruház elkészítését tűztem ki célul, melyet "Fake¹ Lakberendezési Kft." névre kereszteltem. Ahogyan a név is sugallja, egy kitalált lakberendezési áruházról van szó.

Ennek az áruháznak készítem el az információs rendszerét, beleértve a központi adatbázist (mely a rendszer lelkét adja), a webes felületet, amin keresztül a termékeket megtekinthetjük és megrendelhetjük, kérdéseinket feltehetjük az eladók felé, illetve eladóként kezelhetjük a beérkező megrendeléseket, válaszolhatunk a vásárlók által feltett kérdésekre, módosíthatjuk a termékek és a gyártók adatait, kezelhetjük az áruház pénzügyi adatait, megtekinthetjük az értékesítési statisztikát, illetve – feltéve, hogy rendszergazdai jogosultsággal rendelkezünk – létrehozhatunk vagy törölhetünk más adminisztrátorokat a rendszerben, stb., valamint egy vagy több webes alkalmazást, melyek kiegészítik a felületet olyan szolgáltatásokkal, melyeket hagyományos weboldal-fejlesztési eszközökkel nem lehetne megvalósítani.

További cél, hogy a rendszer a biztonságos működés mellett megfelelően nyílt legyen ahhoz, hogy gyorsan és hatékonyan tudjunk új szolgáltatásokat megvalósítani, akár a korábbiaktól teljesen független feladatok megoldására,

1 Fake (angol, ejtsd: fék) jelentése: kitalált, valótlan, nem igazi, hamis.

ezen felül támogatnia kell a nemzetköziséget, azaz a rendszernek képesnek kell lenni felismerni, ha idegen helyen használják, és annak megfelelő nyelvű felületet ajánlani, vagy ennek hiányában a leggyakrabban használt idegen nyelven (angolul) kommunikálni a felhasználóval. Ennek megfelelően az Angol lesz az alapértelmezett nyelve a felületnek, de benne lesz a magyar fordítás is, tehát magyar helyi beállítások esetén a felület magyar nyelven fog megjelenni a böngészőben.

1.2. A RENDSZER FELÉPÍTÉSE, MŰKÖDÉSE

Szakdolgozat által érintett információs rendszer alapvetően három fő részből fog állni:

- 1 központi adatbázis (mind a vásárlói, mind az eladói oldalt kiszolgálja, az adatok csak a felület, illetve az adatbázis által biztosított függvényeken és eljárásokon keresztül elérhetőek, közvetlenül nem);
- 2 Vásárlói oldal (a felület azon része, melyen keresztül a termékek megtekinthetőek és megrendelhetőek, az eladók részére üzenetek küldhetők);
- 3 Eladói központ (a felület azon része, melyen keresztül fel lehet venni a megrendeléseket, válaszolni lehet a vásárlók kérdéseire, a termékek és a szállítók adatai módosíthatók, további adminisztrátorok adhatók hozzá a rendszerhez, illetve távolíthatók el belőle, a pénzügyi tranzakciók kezelhetők, illetve az értékesítések statisztikái megtekinthetőek);
 - 3.1 Webes alkalmazások az extra szolgáltatásokhoz (az eladói oldalba beépülve futnak, önállóan nem futtathatók, leginkább diagramok dinamikus kirajzolására használatosak. Az értékesítési adatok megváltozása a diagramokban is meg kell, hogy jelenjen).



A 3. ábrán az információs rendszer leegyszerűsített rajzát láthatjuk:

3. ábra. A rendszer alapvető felépítése

Az ábrán kék nyilakkal jelöltük az adatáramlás irányát. Látszik, hogy a webes felület az adatbázisból veszi az adatokat, illetve oda küldi vissza a változtatásokat, ugyanakkor a webes alkalmazások (applet-ek) nem férnek hozzá közvetlenül az adatbázishoz, a felülettől kapják meg a szükséges információkat. Ennek oka az áttekinthetőség és biztonság növelése: az adatbázishoz kizárólag lokálisan, egyetlen ponton keresztül lehet csatlakozni, a felhasználó pedig csak a webes felülettel, illetve alkalmazásokkal tud kommunikálni.

A felület kényelmes és áttekinthető hozzáférést biztosít az adatbázisban tárolt adatokhoz. Egyszerű műveletek esetén (pl. néhány rekord lekérdezése, módosítása vagy törlése egy vagy több adattáblából) a felület közvetlenül az adattáblákon hajtja végre az adott műveletet, amennyiben ez nem zavarja meg az adatbázis konzisztenciáját, azaz a módosított adattáblával kapcsolatban álló más adatbázis elemek képesek érzékelni és helyesen kezelni a megváltozott adatokat. Bonyolultabb műveletek esetén (pl. termékek hozzáadása a bevásárló kosárhoz úgy, hogy a túl régóta inaktív bevásárlókosarak törlődjenek, vagy megrendelések rögzítése, visszaigazolása, állapotainak beállítása és a módosítások alkalmazása az érintett termékek készleteire) a felület az adatbázis által biztosított függvények és eljárások segítségével módosíthatja a tárolt adatokat. Ez többnyire egy egysoros, nagyon egyszerű lekérdezést jelent, mert a felületnek nem kell tudnia, hogy pontosan mi történik az adatbázisban tárolt adatokkal, az adatbázis dolga, hogy a kapott paramétereket megfelelő alapossággal ellenőrizze és helyes paraméterek esetén megfelelően módosítsa az adattáblákban tárolt rekordokat.

Ugyanakkor a felületnek pontosan kell tudnia, hogy a különleges adatokat (pl. használt valuta vagy az áruház[lánc]hoz tartozó telephelyek adatainak lekérdezése) melyik táblázatból tudja kikeresni, mert ez egyszerűbb és gyorsabb programot eredményez.

Összességében tehát ott használunk segédfüggvényeket az adatbázisban, ahol a végrehajtandó műveletek bonyolultsága ezt indokolja. A legtöbb helyen törekszünk alacsony szinten, az adatbázis elemi lehetőségeivel (ún. megszorításokkal, vagy constraint-ekkel) ellenőrizni az újonnan megkapott adatok helyességét, ahol erre lehetőségünk van.

Miután a felhasználó kérésére rendszer megkísérelte módosítani az adatbázisban tárolt adatokat, a rendszer visszajelzést küld a művelet sikerességéről vagy sikertelenségéről (pl. sikeres megrendelés esetén visszaküldi a vásárlóknak a generált rendelési azonosítót, amivel a megrendelt termékek személyes átvétele esetén hivatkozhatnak megrendelésükre az áruház vevőszolgálatán, sikertelen megrendelés esetén pedig hibaüzenetet küld vissza).

A felhasználó megfelelő segítése a munkában a webes felület feladata. Űrlapok kitöltésénél pontosan meg kell jelölni, hogy mely mezők kitöltése kötelező, hogy az űrlap kitöltése minél egyszerűbb legyen. Ergonómiai szempontból arra is figyelni kell, hogy ne legyen túl sok kitöltendő mező az űrlapon, mert ez megzavarhatja a felhasználót. Ha sok adatot kell bekérnünk a felhasználóktól, akkor érdemes ezt léptetéses rendszerrel megvalósítani, megfelelően csoportosítva a kitöltendő mezőket. A felületen feltett kérdéseknek érhetőnek kell lenniük, pontosan meg kell mondanunk, hogy milyen adatokra van szükségünk.

1.2.1. HATÉKONYSÁG

Miután előre lehetetlen megmondani, hogy a jövőben összesen hány felhasználót kell majd kiszolgálnunk, illetve hány termékkel és gyártóval lesz dolgunk az áruházban, nagyon nagy hangsúlyt kell fektetnünk arra, hogy a rendszer gyors legyen! Az adatbázisban és a felületen található rutinoknak hatékonyan kell működni, fel kell készíteni a rendszert arra, hogy egy lekérdezéssel akár több ezer termék adatát is vissza kell adnunk, éppen ezért pontosan tisztáznunk kell, hogy milyen rutinokat milyen rutinokba ágyazunk be! Lehetőség szerint kerülni kell az olyan lekérdezéseket, melyek az elengedhetetlennél több al-lekérdezéseket indítanak, és az olyanokat is, melyek egyszerre túl sok adattáblát vizsgálnak meg vagy kapcsolnak össze!

Nagy mennyiségű adat kezeléséhez olyan algoritmusokat kell találnunk, amelyek lépésekben vizsgálják át a szükséges adatokat: egyszerre nem dolgozhatunk több adattáblával, mint amennyi elengedhetetlenül szükséges, a kapott eredményeket tároljuk, és a következő művelet már a tárolt adatokkal dolgozik tovább, szükség esetén felhasználhat újabb adattáblákat, majd ennek a műveletnek is tároljuk az eredményét és így tovább. Törekednünk kell tehát arra, hogy újabb és újabb adatok esetén a végrehajtási idő csak lineárisan növekedjen, esetleg konstans-szorosára, bár a konstans-szorossal is vigyázni kell, bizonyos esetekben már az is túl nagy ugrást jelenthet futási időben! Néha okosabb elhagyni olyan ellenőrzéseket, melyek megvizsgálják, hogy a módosítás tényleges módosítást jelentett-e, vagy a régi adat íródott vissza még egyszer, mert – ritka esetekben – az ellenőrzés tovább tarthat, mint magának a műveletnek a végrehajtása.

1.2.2. Kiterjeszthetőség, egyéb jellemzők

A rendszernek megfelelően nyitottnak kell lennie a további fejlesztésekhez, új szolgáltatások beépítéséhez. Lehetőség szerint minden szolgáltatást külön egységben kell megvalósítani, és ügyelni kell arra, hogy ezek az egységek minél kevésbé függjenek egymástól. Természetesen a termékekhez kapcsolódó fórum-szolgáltatást nem lehet külön választani az érintett termékektől, de mind a termék tartalmi jellemzőit, mind a termékkel kapcsolatos hozzászólásokat külön-külön egységben kell kezelni. Ugyanez vonatkozik a gyártókra is.

Többször előfordulhat, hogy a rendszerben tárolt adatok egymásra hivatkoznak, egymásra épülnek, egymástól nagymértékben függenek. Ezeket a függéseket tiszteletben kell tartani, mind az adatbázisban, mind a hozzá kapcsolódó felületen megfelelő hierarchiában kell, hogy álljanak, és az egyes rutinok prioritásait ennek megfelelően kell kezelni.

Ugyanakkor, az egymástól teljesen független szolgáltatásokat egymástól teljesen függetlenül kell kezelni, a rendszernek alkalmasnak kell lennie arra, hogy ezek közé újabb szolgáltatásokat lehessen beépíteni, illetve felesleges, nem használt szolgáltatásokat el lehessen távolítani.

Az adatbázis tartalmazhat olyan függvényeket, eljárásokat, nézeteket vagy adattáblákat is, melyeket normál esetben egyáltalán nem használ a rendszer, ám egy komolyabb fejlesztés esetén nagyon hasznosnak bizonyulhatnak. Ilyen lehet például a postai irányítószám-település nyilvántartása és a hozzájuk kapcsolódó lekérdező függvények megléte a megrendelések gyorsabb leadásához, vagy a leggyakrabban használt vezeték- illetve keresztnevek nyilvántartása, ami tesztadatok létrehozását könnyítheti meg. Ezeket nem feltétlenül használja ténylegesen az elkészítésre kerülő információs rendszer, ám a jövőbeli fejlesztési munkákat rendkívül megkönnyíthetik, némely esetekben pl. egy hét fejlesztési munkát lerövidítve egy napra.

1.2.3. ALAPKONCEPCIÓ, MEGKÖTÉSEK

Elsődleges célunk az áruházi forgalom maximalizálása, tehát legyen minél több sikeres értékesítésünk, azaz legyen minél több termék eladva. A szakdolgozatban eltekintünk az értékesítések számát befolyásoló gazdasági tényezőktől, és végig feltesszük, hogy az értékesítések száma tisztán informatikai eszközökkel a végtelenségig növelhető (ami a valóságban természetesen nem így van), tehát minél jobb rendszert alkotunk, annál több terméket tudunk eladni.

Célunk az oldal látogatószámának, az oldalon keresztül leadott megrendelések számának, ezeken keresztül pedig az eladások számának maximalizálása. Minél több mindent tudnak a potenciális vásárlók elintézni oldalunkon keresztül (és azt minél egyszerűbben), annál több megrendelésre számíthatunk. Éppen ezért a vásárlók és az eladók közötti interakciónak tökéletesen kell működnie, az oldalnak könnyen kezelhetőnek kell lennie, ugyanakkor feltételeznünk kell azt is, hogy értékesítőinknek a megrendelések kezelése és a vásárlók kérdéseinek online megválaszolása mellett rengeteg egyéb, adminisztrációs jellegű háttérmunkája is van, tehát törekednünk kell arra, hogy a rendszer minél több feladat elvégzését átvegye tőlük. Sikeres értékesítés esetén például a bevételt a rendszer automatikusan könyvelje le és a tranzakcióra vonatkozó bejegyzést lássa el értelmes leírással.

Arra is figyelni kell, hogy olyan termékekkel is akadhat dolgunk, amikből fizikailag egy darab sincs az áruházban készleten, viszont meg tudjuk rendelni a gyártótól, a termék jó minőségű, esztétikus, így mindenképpen szeretnénk belőle eladni. Éppen ezért támogatni kell a rendelhetőséget is, tehát a készletek kezelésénél figyelembe kell venni azt, hogy az ilyen termékből éppen nincs fizikai készlet, mégis megjelenik az oldalon, a vásárlók meg tudják rendelni, és a megrendelést követően néhány héten belül meg is kaphatják a terméket.

Néha olyan termékekkel is találkozhatunk, melyek külsőre összeillenek,

passzolnak egymáshoz, esetleg kedvező áron el is tudjuk adni mindet. Emiatt az információs rendszernek támogatnia kell szettek létrehozását is: ha tehát valamilyen szempont alapján csoportosítunk kettő vagy több terméket, és azokat szeretnénk egy termékként valamilyen kedvezménnyel eladni, akkor a rendszernek külön termékként is meg kell tudnia jeleníteni őket, összeadva az áraikat, az összegre alkalmazva a kért kedvezményt, a készletet pedig "rendelhető" státuszra beállítva.

Ha egy termék éppen akciósan kapható, akkor a normál árat (vagy listaárat) és az akciós árat is meg kell jeleníteni, az akciós árat külön kiemelve, más színnel, látványosan szemléltetve azt, hogy az adott terméket most a szokásosnál kedvezőbb áron lehet megvásárolni. Ez komoly tényező lehet egy bevásárlólista összeállításánál.

Weboldalunkon a termékeknek dinamikusan, szépen, kategóriákra bontva, rendezetten kell megjelenni, a termékek közötti navigálást a lehető legkényelmesebbé kell tenni, és mivel gyakori jelenség, hogy egy vásárlót kifejezetten csak egyfajta termék érdekel, kötelező implementálnunk egy hatékony keresőmotort is, amivel szövegrészletre lehet keresni – lényegében bárhol a terméklapon – mert a vásárló nem fogja az összes termékünket végignézni, ha neki csak egyféle kell, hanem elnavigál egy konkurens cég oldalára, amennyiben nem biztosítunk számára megfelelő keresőfelületet! A hatékonyabb keresés érdekében úgy valósítjuk meg a keresőmotort, hogy termékkódra is rá lehessen vele keresni.

Mivel célunk az, hogy minél többen látogassák és használják oldalunkat, nagyon fontos, hogy az oldal gyors és megbízható legyen, optimalizáljuk a megjelenítést, csak korlátozott számú listaelemet jeleníthetünk meg egyszerre, azért, hogy a szerver ne omoljon össze a nagy látogatószám miatt, ha esetleg a vártnál jóval többen keresik fel áruházunkat egyszerre.

1.2.4. EGYÜTTMŰKÖDÉS MÁS WEBÁRUHÁZAKKAL

Szem előtt kell tartanunk annak tényét, hogy egy frissen beindított webáruháznak csak viszonylag hosszú idő elteltével lesz megfelelő számú látogatója. Ugyanakkor, ezalatt az idő alatt is szeretnénk megfelelő mennyiségű forgalmat generálni, tehát valószínűleg a saját weboldalunkon kívül valahol máshol is szeretnénk meghirdetni termékeinket.

Éppen ezért információs rendszerünket fel kell készítenünk arra, hogy támogassa más weboldalak formai és tartalmi követelményeit. Pontosabban: termékeink más weboldalakra történő feltöltéséhez biztosítanunk kell információs rendszerünkben olyan kimenetet, ami az adott oldalnak bemenete lehet. Ehhez az eladói központban implementálunk egy ún. Import/Export menüt, melynek segítségével könnyedén tudjuk az összes termékünk adatát különféle formátumokban letölteni.

Emellett, az is gyakran előfordulhat, hogy már más weboldalakon meghirdetett termékeinket a saját weboldalunkra is fel akarjuk tölteni, méghozzá nagy mennyiségben, vagy központi kérésre egyszerre nagyon sok termék adatát kell módosítanunk. Ehhez implementálunk egy ún. adatforrásfeltöltő szolgáltatást is az Import/Export résznél, melynek segítségével – információs rendszerünk legfőbb felhasználóival történt egyeztetést követően – tömegesen frissíthetők termékeink adatai (bizonyos feldolgozási idő árán), illetve néhány nagyon egyszerű alapadat megadásával villámgyorsan lehet új termékeket feltölteni a termékadatok kezelésénél.

2. Felhasználói dokumentáció

Információs rendszerünket két felületről lehet elérni: a vásárlói oldalról, illetve az eladói központból. Áruházunk adatai ezeken keresztül érhetők el, webböngésző segítségével, miután beírtuk a meglátogatni kívánt oldal címét.

A vásárlói oldalon keresztül megtekinthetjük áruházunk termékeit, a termékekkel kapcsolatban kérdéseket tehetünk fel, illetve megrendelhetjük a termékeket. A vásárlói oldal használata nem igényel semmilyen regisztrációt.

Az eladói központ használata regisztrációhoz kötött. A regisztráció kizárólag meghívásos alapon történhet, új felhasználókat az oldal rendszergazdái vehetnek fel, vagy törölhetnek régi felhasználókat. Az eladói központban módosíthatjuk a termékek és gyártóik adatait, új tételeket vehetünk fel, adatokat importálhatunk/exportálhatunk, kezelhetjük pénzügyeinket, a megrendeléseket, válaszolhatunk az oldal látogatói által feltett kérdésekre, megtekinthetjük a statisztikát, kezelhetjük a felhasználókat és saját adatainkat, stb.

2.1. A VÁSÁRLÓI OLDAL

A vásárlói oldal (index.php) négy fő panelból tevődik össze:

- Címsáv (az oldal tetején helyezkedik el, egyfajta fejlécnek tekinthető. baloldalán az áruház emblémája, közepén az áruház neve, jobboldalán pedig a bevásárlókosár található);
- Kategórialista keresőmezővel (az oldal baloldalán helyezkedik el, segítségével tallózni lehet az egyes kategóriákba tartozó termékek között, illetve konkrét termékekre lehet rákeresni);

- Elérhetőségi adatok (az oldal alján helyezkedik el. Ha a vásárlónak olyan kérdése van, vagy olyan segítségre van szüksége, ami az oldal által nyújtott szolgáltatásokhoz nem tartozik, akkor az itt szereplő elérhetőségi adatok segítségével léphet kapcsolatba áruházunk ügyfélszolgálatával / vevőszolgálatával);
- Tartalom (az oldal jobboldalán helyezkedik el, minden konkrét tartalommal bíró elem itt jelenik meg, beleértve a terméklapokat, a bevásárlókosár tartalmát, a megrendelőlapot, a terméklistát, stb.).

Az esztétikus és áttekinthető megjelenítés érdekében az oldal szigorúan betartja ezt az elrendezést, bármelyik szolgáltatást is vegyük igénybe az oldalon. Mindezt grafikusan a 4. ábrán szemléltetjük:



4. ábra. A vásárlói oldal elrendezése

Mivel az elrendezés az oldal használata közben egyszer sem változik meg, a keresőmező, az egyes kategóriákhoz tartozó termékek listája, a bevásárlókosár tartalma, illetve az elérhetőségi adatokra való hivatkozások egy-egy kattintással elérhetőek, az oldalon való navigálás tehát ennyire egyszerű. Külön visszafele léptető hivatkozások a tartalom-panelben nem is nagyon vannak, mert nincs rájuk szükség. A különféle tartalmakra való hivatkozások benne vannak az URL-ben (kivéve a szenzitív adatokat), így a legtöbb esetben a használt böngésző visszafele léptető gomba tökéletesen elég a visszaléptetéshez, kivéve akkor, amikor valamilyen fontosabb adatot (pl. megrendelést vagy

fórumban hozzászólást) szándékozunk elküldeni az oldalnak. Ekkor ugyanis az adatok elküldése után automatikusan megjelenik egy hivatkozás, ami a megfelelő kezdőlapra mutat, vagy azonnal oda is irányít minket.

A vásárlói oldal főoldalának alapértelmezett, tengerkék témájú megjelenítését az 5. ábrán láthatjuk:



2.1.1. TERMÉKEK MEGTEKINTÉSE

Áruházunk weboldala alapértelmezés szerint mindig ár szerint növekvő sorrendben listázza ki a termékeket. Ha nem kattintunk semmire, miután bejött a főoldal, akkor az összes létező termékünk közül választja ki az alapértelmezés szerinti 10 legolcsóbbat. Ha rákattintunk valamelyik kategóriára, akkor már csak az azon belüli termékek közül láthatjuk a 10 legolcsóbbat. Ha a keresőmezőt használva rákeresünk valamilyen szövegre a termékek között, akkor minden olyan terméket érint a listázás, amelyben valahol megtalálható a keresett szöveg, vagy cikkszáma megegyezik a keresőmezőbe beírt számmal, és erre a szűrt listára alkalmazzuk ismét a rendezést, illetve listában található sorok számának korlátozását.

A lista egy fejléccel kezdődik, melyben lehetőségünk van megváltoztatni a rendezési szempontot, illetve azt is, hogy hány listaelem jelenhet meg egyszerre. A lista fejlécét a 6. ábrán láthatjuk:

Rendezés mint: Ár, Cikkszám, Kulcsszavak, Megnevezés	Sorok megjelenítése: 10, 20, 30, 50, 100
6. ábra. A termékek listájának fejléd	ce

A lista első oszlopában a termék ikonméretű fényképe található, a második oszlopban a termék megnevezése, a hozzá tartozó kulcsszavak és a termék cikkszáma, a harmadik oszlopban a lerövidített leírás, a negyedikben pedig az ár és a bevásárlókosár ikonja található. Ha a termék akciós, akkor a negyedik oszlopban a normál ár (vagy listaár) és az akciós ár is szerepel, az akciós ár külön kiemelve, a normál ár pedig áthúzva.

A lista láblécében találjuk a lapozáshoz szükséges elemeket (7. ábra). A nyilak segítségével az előző/következő oldalra ugorhatunk, a számok segítségével pedig lehetőségünk van oldalakat átugrani, illetve a lista legelejére/legvégére navigálni.



Ha egy terméket meg szeretnénk tekinteni, akkor kattintsunk a terméklistán a termékhez tartozó megnevezésre vagy fényképre. Ekkor megjelenik a kiválasztott termékhez tartozó terméklap (8. ábra).



8. ábra. Egy vizespohárhoz tartozó terméklap

A terméklapon meg van jelenítve a termék fényképe, neve, a hozzá tartozó kulcsszavak, a termék cikkszáma, a termék ára (akció esetén normál ár is és akciós ár is), a termékre adott garancia, a termék gyártója (szállító, vagy beszállító), a termékből rendelkezésre álló készlet, a szállítási díj, a termék súlya és a termék teljes leírása.

Minden termékhez legalább 1, legfeljebb 4 fénykép van feltöltve. Az aktuálisan megnyitott fénykép alatti ikonok segítségével a többi feltöltött képet is megtekinthetjük, amennyiben legalább két fénykép van feltöltve a termékhez.

2.1.2. FÓRUM HASZNÁLATA

Amennyiben egy terméklap nem szolgál nekünk megfelelő mennyiségű és tartalmú információval, a fórum segítségével feltehetjük kérdéseinket az eladók felé. A fórum megnyitásához kattintsunk a terméklapon található "Fórum" fülre. A fórum alapvető megjelenítését a 9. ábrán láthatjuk:

#6 Ábrahám Róbert írta		2012-11-26 12:09:28
Jó kis pohár, én fényképezt	em le :)	
Vezetéknév*	Üzenet szövege*	The Market
Keresztnév*		and the second se
Email*	the parts	Küldés
*Kötelező kitölteni!		

9. ábra. A fórum megjelenítése a vásárlói oldalon

A hozzászólásokat időrendben láthatjuk, csökkenő sorrendben. Minden hozzászólás fejlécében megjelenik a hozzászólás egyedi azonosítója (amire a későbbiekben hivatkozni lehet), a hozzászóló neve, a hozzászólás dátuma és ideje. A hozzászólás törzsében maga a konkrét poszt található.

A hozzászólások alatt található egy űrlap, melynek segítségével új hozzászólást lehet írni. A keresztnév kivételével az űrlap minden mezőjének

Fórum használata

kitöltése kötelező (cégnév esetén nem biztos, hogy megadunk keresztnevet). A hozzászóló email címe a vásárlói oldalon nem jelenik meg. A hozzászólások csak az ábécé betűit, számokat, illetve az általánosan használt, nyelvtanilag elfogadott írásjeleket tartalmazhatják, egyéb, érvénytelen karaktereket vagy tartalmat (például HTML-kódot) nem. Ha a rendszer ilyet talál, azt megkísérli kiszűrni, hogy az oldalon megjelenő hozzászólás már csak közönséges szöveget tartalmazzon. A hozzászólásnak legalább 1 értelmes karaktert tartalmaznia kell.

Bármikor visszatérhetünk a termék leírásához, ha rákattintunk a "Leírás" fülre a terméklapon.

2.1.3. BEVÁSÁRLÓKOSÁR KEZELÉSE ÉS TERMÉK MEGRENDELÉSE

Amennyiben egy termék elnyerte tetszésünket és meg szeretnénk vásárolni online, a terméklapon a kosár alatti mezőbe írjuk be, hogy hány darabot szeretnénk vásárolni a termékből (alapértelmezés szerint 1 db-ot), majd kattintsunk a kosár ikonjára (10. ábra).



10. ábra. Tetszőleges számú termék kosárba helyezése

Ha egyszerre csak egy terméket szeretnénk a bevásárlókosárba helyezni, akkor ezt a termékek listájánál is megtehetjük, ha rákattintunk a bevásárlókosár ikonjára az adott terméknél (11. ábra).



11. ábra. Egy darab termék kosárba helyezése

Ügyeljünk arra, hogy a megvásárolni kívánt mennyiség nem lehet több, mint amennyi a termékből raktáron van! Amennyiben egy termék csak rendelhető készleten van (azaz: nem feltétlenül van belőle tényleges készlet az áruház raktárában, de az áruház tud belőle rendelni szükség esetén), akkor ilyen korlátozás nincs.

Miután egy vagy több terméket behelyeztünk a kosárba, látnunk kell, hogy az oldal jobb felső sarkában található bevásárlókosár alatti szöveg megváltozik: most már nem üres kosarat jelez, hanem kiírja, hogy hányféle termék található benne (12. ábra).



12. ábra. Üres és nem üres bevásárlókosár

A kosárba behelyezett termékek megrendeléséhez kattintsunk erre a bevásárlókosárra. Ekkor az oldal elnavigál minket a bevásárlókosár tartalmához, ahol egy listában láthatjuk a bevásárlókosárba helyezett elemeket. Ebben a listában – az előzőekben bemutatott terméklistához hasonlóan – megtalálható az érintett termékek ikonméretű fényképe, megnevezése, a hozzájuk tartozó kulcsszavak, cikkszámaik, rövidített leírásuk, egységáruk (akció esetén akciós ár és normál ár egyaránt), a rendelési mennyiség, és

termékenként a fizetendő összeg.

A lista alatt megtalálható a teljes fizetendő összeg is. Figyelem: a lista alatt található összesített ár a szállítási díjakat nem tartalmazza! Ennek oka az, hogy az egyes termékek szállítási költségei több tényezőtől függhetnek, mint a listában szereplő termékek adatai! Ilyen tényező lehet például az áruházzal szerződésben álló futárszolgálat díjszabása, a szállítási cím, és az a tény, hogy a kisebb méretű és súlyú termékek egybe is csomagolhatóak, amivel mérsékelhető a szállítási díj, tehát nem áll rendelkezésre olyan díjtáblázat, ami alapján minden esetben helyesen lehet automatikusan kiszámláztatni a rendszerrel a megrendelt termékek kiszállításának díját!

Ha egy terméket el kívánunk távolítani a bevásárlókosárból, akkor kattintsunk az adott termékhez tartozó bejegyzés legvégén található törlés ikonra! Ekkor a teljes termék törlődik a kosárból, nemcsak egyetlen darab!

Fánykán	Mognovozás	Lairás	Equipógán	Monnuicóa	Fizetendő
renykep	Cortina Vizespohár Poharak Cikkszám: 9068752	T9068752 Pohár Vízes 421035998/050401/9998Kedves Online Megrendelő! Jogosult A Megrendeléstől A Termék Átvételét Követően 8 Munkanapo Bel�	211 HUF	l	211 HUF
	<mark>Söröskorsó 6dl</mark> Sörös Poharak Cikkszám: 18597479	Sörös PoharakKedves Online Megrendelő Jogosult A Megrendeléstől A Termék Átvételét Követően 8 Munkanapon Belül Indoklás Nélkül Elálln	332 HUF 229 HUF	1	229 HUF
				Fizetendő:	440 HUF
		A számolt összeg nem tartalmazza a kiszállítási díjat!			
zemélyes a	adatok				
čérjük, töltse	ki a szállítási adatokat!				
Vezetéknév		Keresztnév*	Email*		
Telefonszán		Átvétel típusa*			
		Futárszolgálattal T	Következő		

A bevásárlókosárhoz tartozó oldal megjelenítését a 13. ábrán láthatjuk:

13. ábra. A bevásárlókosárhoz tartozó oldal megjelenítése

Az oldal alján megtalálhatjuk azt az űrlapot, melynek segítségével leadhatjuk rendelésünket a bevásárlókosárban szereplő termékekre. A megrendelés leadása három fő lépésből áll: alapadatok (személyes adatok), szállítási/átvételi adatok és számlázási adatok megadása.

Az alapadatok kitöltése kötelező, jelenleg kivétel ez alól a telefonszám és keresztnév megadása, de a jövőbeli fejlesztésekre vonatkozó tervek alapján (a megrendelők megbízhatóbb azonosíthatósága érdekében) ez is kötelező lehet. Az alapadatokhoz tartozik a vezetéknév, keresztnév, email-cím, telefonszám és az átvétel típusa (jelenleg még csak futárszolgálat, posta ponton történő vagy személyes átvétel).

A szállítási/átvételi adatok az előző lépésben megadott átvételi típustól függenek. Futárszolgálat vagy posta ponton történő átvétel esetén meg kell adni az átvevő vezetéknevét, keresztnevét, a postai irányítószámot, település nevét, az utcanevet és házszámot, valamint a fizetés módját. A "Megjegyzések" rész kitöltése nem kötelező; ide bármilyen különleges kérést be lehet írni. Az ide tartozó űrlapot a 14. ábrán láthatjuk:

térjük, töltse ki a szállítási	i adatokat!		
Vezetéknév*	Keresztnév*	Irányítószám*	Város*
Cím*	Fizetés módja*	Megjegyzések	
	Előreutalás 🔹		Következő

14. ábra. Rendelés leadása futárszolgálat igénybevételével

Személyes átvétel esetén lényegesen kevesebb adatot kell megadni a szállítási adatoknál. Ide tartozik a fizetés módja, amit itt is meg kell adni, az átvétel helye, azaz, hogy melyik telephelyen kívánjuk átvenni a megrendelt termékeket, illetve ide is fűzhetünk megjegyzéseket (15. ábra).

Fizetés módja*	Megjegyzések	Átvétel helye*	
Előreutalás	•	Fake Lakberendezési Kft.	Következő
Kötelező kitölteni!			

15. ábra. Rendelés leadása személyes átvétellel

Az utolsó lépés a számlázási adatok megadása. Ezt az űrlapot csak akkor kell kitölteni, ha máshová kérjük a számlát, mint a szállítást, vagy személyes átvételt kértünk, de meg kívánjuk adni lakcímünket számlázási címként (16. ábra).

Számlázási adatok			
Kérjük, töltse ki a számlázási adat	okat, ha a számlát máshová kéri, mint a szállítást!		
Vezetéknév	Keresztnév	Irányítószám	
Város	Cím		
		Küldés	

16. ábra. Rendelés leadása számlázási adatok megadásával

Ha minden adatot helyesen adtunk meg a megrendelés leadásakor, akkor a "Küldés" gombra való kattintás után a 17. ábrán láthatóhoz hasonló üzenetnek kell megjelennie:

Megrendelése elküldve és rögzítve.	
Az Ön rendelési azonosítója:: 407550.	and the second second
Kérjük, várjon, amíg eladóink felveszik Önnel a kapcsolatot!	
	<u>Vissza a föoldalra</u>

17. ábra. Sikeres megrendelés leadása

Ez az üzenet informál minket arról, hogy a megrendelés leadása sikeres volt, tehát a megrendelésünk beérkezett az eladói központba és rögzítésre került. Beérkezéskor a rendszer automatikusan generált nekünk egy rendelési azonosítót, amit a weboldalon keresztül visszaküldött nekünk. A jövőben ezzel az azonosítószámmal tudunk majd hivatkozni megrendelésünkre az eladókkal történő kapcsolatfelvételt követően.

Ha a megrendelés elküldése sikertelen volt, mert valamit nem jól töltöttünk ki, akkor a 18. ábrán látható hibaüzenetet kapjuk eredményül:

Megrendelés elküldése sikertelen! Kérjük, helyesen töltse ki a megrendelő űrlapot!			
	Vissza	The second designed and the second	

18. ábra. Megrendelés elküldése sikertelen

Ez az üzenet valójában azt jelenti, hogy a rendszer megkapta ugyan a megrendelésünket, de az űrlap helytelen kitöltése miatt a kapott adatokat nem

lehetett feldolgozni. Ilyenkor kattintsunk a "Vissza" hivatkozásra és töltsük ki újra, helyesen az űrlapot, majd próbáljuk meg újra elküldeni. Sikeres megrendelés esetén a "Vissza a főoldalra" hivatkozásra kattintva visszatérhetünk a termékek listájához.

Üres bevásárlókosár esetén a megfelelő üzenet jelenik meg, és az előbb említett "Vissza a főoldalra" hivatkozással visszatérhetünk a termékek listájához.

Figyelem: a bevásárlókosár ideiglenes tárolónak minősül, amely kizárólag azt a célt szolgálja, hogy a rendszer tárolja az általunk megrendelni kívánt termékek listáját addig, amíg leadjuk megrendelésünket az adott termékekre. Sikeres megrendelést követően a bevásárlókosár tartalma automatikusan törlődik.

Ezen kívül – mivel a bevásárlókosár csak egy ideiglenes tároló – a benne tárolt adatokat csak korlátozott ideig őrzi meg a rendszer! Ha csak belehelyezünk termékeket a bevásárlókosárba, de nem rendeljük meg őket, akkor a benne tárolt adatok 3 nap elteltével törölhetővé válnak, és törlődnek is, amikor új felhasználó veszi igénybe a rendszer szolgáltatásait, mivel a rendszer csak korlátozott számú felhasználót képes kiszolgálni!

2.2. Az eladói központ

Az eladói központ (admin.php) használata regisztrációhoz kötött. Amennyiben nem vagyunk belépve a rendszerbe, és így próbáljuk megnyitni az oldalt, a rendszer automatikusan átirányít minket a bejelentkezési oldalra. Az eladói központ a vásárlói oldalon keresztül is megnyitható, ha az oldal jobb alsó sarkában található "Eladói központ" hivatkozásra kattintunk.

Az eladói központba történő belépéshez meg kell adnunk felhasználónevünket és jelszavunkat, amit a weboldal rendszergazdáitól kapunk

meg. Önálló regisztráció nem lehetséges, az eladói központhoz való hozzáférést kizárólag meghívásos alapon kapjuk meg. A bejelentkezési oldalt a 19. ábrán láthatjuk:

Fake Lakberendezési Kft.	
Bejelentkezés A bejelentkezéshez adminisztrációs csoport tagság szükséges! Ethasználónév: Jelszó:	<u>Vissza a föoldalra</u>

19. ábra. Bejelentkezés az eladói központba

A "Vissza a főoldalra" című hivatkozásra kattintva visszatérhetünk a vásárlói oldalra. Amennyiben bejelentkezési kísérletünk sikertelen, hibaüzenet jelenik meg az oldalon, és a bejelentkezést megkísérelhetjük újra.

Közvetlenül bejelentkezés után áttekinthetjük saját felhasználói adatainkat, és elvégezhetjük rajtuk a szükséges módosításokat (20. ábra).

Etál-b «Állík	fach	Tovább az eladói központra
Flokbeann	ASOK	
Vezetéknév*:	- Aller	
Keresztnév*:		
Email*:	The second s	
Felhasználónév:		
ÚJ jelszó:		
Új jelszó mégegyszer.		
Telefonszám:		
Irányítószám:		
Város:		
Cím:		
Születési hely:		
Születési dátum:		
in the second second	(Minta: 1980-01-01)	
Altalanos iskola:		
Kozepiskola:		
Egyetem:		
Liozo munkanelyek:		
Küldés		

20. ábra. Felhasználói adatok gyors beállítása

Ha módosítani kívánjuk valamelyik adatunkat, akkor az új adatot írjuk be a megfelelő mezőbe, majd kattintsunk a "Küldés" gombra. Ekkor megjelennek a módosított adatok. Ha végeztünk a beállításokkal, kattintsunk a "Tovább az eladói központra" feliratú gombra. Ekkor megjelenik az eladói központ főoldala.

Az eladói központ elrendezését a 21. ábrán láthatjuk:



21. ábra. Az eladói központ elrendezése

A vásárlói oldalhoz hasonlóan az eladói központ is szigorúan megtartja az ábrán látható elrendezést. Bármelyik menübe is megyünk bele, mindvégig megmarad az oldal elrendezése. Az eladói központ főoldalát a 22. ábrán láthatjuk:



A főoldalra történő belépés után rögtön a pénzügyi statisztikák fogadnak

minket, melyeket különféle csoportosításokban le is tölthetünk. Ezt a nézetet elérhetjük a főmenüben a "Statisztika" menüre kattintva is bármikor. Egy diagramba belekattintva megváltoztathatjuk a diagram típusát.

Az eladói központban egy új fejléc is megjelenik: a "Login menü". Itt információt kapunk arról, hogy milyen felhasználónéven vagyunk bejelentkezve, felhasználónevünkre kattintva beléphetünk a fiókbeállításokhoz (amit ugyanúgy elérhetünk a főmenüből a "Fiókbeállítások" menüre kattintva), valamint a "Kijelentkezés" hivatkozásra kattintva kiléphetünk az eladói központból és visszatérhetünk a bejelentkezési oldalra.

Ha a főmenüben rákattintunk a "Vásárlói oldal" menüre, akkor a rendszer közvetlenül a vásárlói oldalra navigál minket.

2.2.1. FIÓKBEÁLLÍTÁSOK

Rendszerünkben kétféle fióktípus létezik:

- Normál (a weboldal összes szolgáltatása elérhető, de nem hozhat létre új felhasználókat és nem törölhet más felhasználókat sem);
- 2. Rendszergazdai (teljes hozzáférés a weboldalhoz).

Közvetlenül bejelentkezés után a felhasználói adatok gyors beállításainál nincs különbség normál és rendszergazdai fiók között, mindkét esetben csak a saját adatainkat van lehetőségünk módosítani. A két fióktípus közötti különbség akkor jelenik meg, amikor a "Fiókbeállítások" menüből, vagy a fejlécben a felhasználónévre kattintva belépünk a fiókbeállításokhoz.

Az első oszlopban saját adatainkat láthatjuk ugyanúgy, ahogyan közvetlenül bejelentkezés után, az adataink teljesen ugyanúgy módosíthatók. Amennyiben rendszergazdai fiókkal rendelkezünk, akkor a második oszlopban a többi

felhasználó adatai jelennek meg, illetve egy űrlap, melynek segítségével új felhasználót hozhatunk létre.

A "Fiókbeállítások	' menüt a 23.	ábrán	láthatj	uk:
--------------------	---------------	-------	---------	-----

Vezetéknév*:	Vezetéknév Keresztnév Email	Felhasználónév Jelszó Művelet
Email*: Provincial Constant	Új adminisztrátor hozzáada	ása
Uj jelszó: Új jelszó mégegyszer: Telefonszám: Irányítószám: Város: Cím: Születési hely: Születési dátum: (Minta: 1980-01-01) Általános iskola: Középiskola: Egyetem: Előző munkahelvek:	V ezetéknév*: Keresztnév*: Email*: Felhasználónév*: Jelszó*: Rendszergazda*:	Új adminisztráto
Küldés Götelező kitölteni!		

23. ábra. Fiókbeállítások rendszergazdai hozzáférés esetén

Ha el akarunk távolítani egy felhasználót, akkor az "Adminisztrátorok" résznél a "Művelet" oszlopban kattintsunk az "Eltávolítás" hivatkozásra. Saját magunkat nem távolíthatjuk el, eltávolítani csak rendszergazda tud, így mindig marad legalább 1 rendszergazdai fiók. Új felhasználó létrehozásához töltsük ki az "Új adminisztrátor hozzáadása" űrlapot, majd kattintsunk az "Új adminisztrátor" gombra. Sikeres létrehozás esetén az új felhasználó megjelenik az adminisztrátorok között. Sikertelen létrehozási kísérlet esetén a felhasználólista fölött hibaüzenet jelenik meg.

A felhasználónév létrehozás után már nem módosítható. Létrehozáskor a vezetéknév, keresztnév, email-cím, felhasználónév és jelszó megadása kötelező, a "Rendszergazda" jelölőnégyzetet csak akkor pipáljuk ki, ha

rendszergazdai fiókot kívánunk létrehozni.

2.2.2. TERMÉKEK KEZELÉSE

A főmenüben a "Termékek kezelése" menüre kattintva megnyithatjuk a termékek listáját. Ez a lista nagyban hasonlít a vásárlói oldalon található listához, az elrendezése ugyanaz, de itt bevásárlókosár már nincs, helyette szerkesztés-ikon jelenik meg, illetve a keresőmező már nem a vásárlói oldalnál megszokott helyen található, hanem a terméklista fejlécében (24. ábra).



24. ábra. Termékek kezelése

A kilistázott termékek nevére vagy fényképére kattintva megjelenik ugyanaz a terméklap, mint a vásárlói oldalnál, de a bevásárlókosár helyén itt is a szerkesztés-ikon lesz látható.

Új termék feltöltéséhez kattintsunk a "Termék feltöltése" hivatkozásra, új szett² feltöltéséhez pedig kattintsunk a "Szett feltöltése" hivatkozásra. Mindkét

² Összeállítás meglévő termékekből; az egy szetthez tartozó termékek egyszerre is megrendelhetőek, és a szettre alkalmazott kedvezmény mindegyik termékre vonatkozik, de

esetben hasonló űrlapot fogunk látni, melynél "*" karakter jelöli a kötelezően kitöltendő mezőket.

- Normál termékek esetén a cikkszámok 1-99999999 közötti, illetve 20000000 vagy annál nagyobb egész számok lehetnek, szettek esetén pedig 10000000-199999999 közötti egész számok lehetnek;
- A termék megnevezése tetszőleges szöveg lehet, maximális hosszúsága 255 karakter;
- A termékhez tartozó kulcsszavak olyan kifejezések, amelyek megkönnyíthetik a termékre való rákeresést. Együttes maximális hosszuk 255 karakter lehet;
- A termék állapota háromféle lehet: engedélyezett, letiltott és problémás.
 A problémás állapot egyben tiltást is jelent;
- Az "Ár" mező a termék normál árát jelenti, kitöltése kötelező. Az "Akciós ár" mező kitöltése elhagyható, ha a termék nem akciós;
- A termék szállítójának csak a rendszerben létező szállítót lehet megadni. Új szállító esetén a szállítót fel kell venni a szállítók közé;
- Súlyt nem kötelező megadni, mert nem biztos, hogy beérkezéskor minden termék azonnal le van mérve;
- A készlet megadása kötelező (-1 érték esetén csak rendelésre elérhető), a "Lefoglalt" mező értéke a termék készletéből lefoglalt mennyiséget jelenti, megadása kötelező;
- A garancia megadása kötelező, a felkínált lehetőségek közül választhatunk;
- A "Kategória" mező értéke belső kategóriakód lehet, ez tetszőlegesen

csak akkor, ha magát a szettet rendelik meg, nem pedig külön-külön a termékeket.

Termékek kezelése

módosítható, elhagyható. Egész szám kell, hogy legyen;

- Amennyiben a termék csak személyesen vehető át, pipáljuk ki az ehhez tartozó jelölőnégyzetet;
- Egy termékhez összesen 4 fényképet lehet megadni, 1 fénykép megadása kötelező. A fényképhez internetes hivatkozást adunk meg;
- Az "Eladói kategória" értéke tetszőleges szöveg lehet, elhagyható;
- Videó-link megadására is lehetőség van, nem kötelező;
- A termékhez megadható az ISBN13, valamint a GTIN kód is;
- A gyártói termékkód és a raktári kód megadása nem kötelező, alapértelmezés szerint a cikkszám kerülhet ide;
- Megadható a termékhez egy külső terméklap is, amennyiben a termék más webáruházakban is meg van hirdetve;
- A szállítási díj megadása nem kötelező;
- Az "Árucsoport" a termék főkategóriája³, ami alapján a vásárlói oldalon a termékek csoportosítva vannak. Megadása kötelező;
- A "Cikkcsoport" a termék alkategóriája⁴. Egyike a jövőbeli fejlesztéseknek, jelenleg csak főkategória alapján csoportosítunk;
- Megjegyzések is hozzáadhatók a termékekhez, nem kötelező;
- A termék leírása tartalmazhat szöveget is és HTML-kódot is, megadása kötelező.

Szettek esetén az előzőekhez képest annyi különbség van, hogy elhagyjuk a

³ Elsődleges csoportosítási szempont. A legfőbb termékosztályok ide, mint pl. lámpa, nappali, konyha, fürdőszoba, stb.

⁴ Másodlagos csoportosítási szempont. Általában az egyes termékosztályok termékeit csoportosítjuk tovább vele, mint például nappalin belül kanapék, fotelok, ülőkék, stb.

normál árat (ezt a rendszer automatikusan számolja ki a szettben szereplő termékek áraiból), az akciós ár helyett pedig százalékos kedvezményt adunk meg, ezen kívül meg kell adni a szetthez tartozó termékek cikkszámait és a hozzájuk tartozó mennyiségeket vesszővel elválasztva a "Termékek", illetve "Termékmennyiségek" mezőnél, valamint elhagyjuk a szállítót, súlyt, lefoglalt és teljes készletet, mert ezeket az adatokat is az érintett termékek adataiból állapítja meg a rendszer.

Mindezt a 25-26. ábrákon láthatjuk:

Fermék feltöltés	se		
Cikkszám*:	(1-99999999, 20000000-)	Charles A.	Leirás*:
Megnevezés*:			
Kulcsszavak*:	[
Állapot*:	Engedélyezve	•	
Ár*:			
Akciós ár:			
Szállító*:	Aberham KG	7	
Súly (kg):			
Készlet (-1: Csak rendelésre)*:	1		
Lefoglalt*:			
Garancia:		and a	
Kategória:	ſ		the second
Csak személyesen vehető át:	0		as a
Fénykép 1. (link)*:			

25. ábra. Új termék feltöltése (részlet)

zett feltöltése	e		
Cikkszám*:	(100000000-199999999)	Leirás*:	
Megnevezés*:			
Kulcsszavak*:			
Állapot*:	Engedélyezve		
Akció (%):			
Termékek*:			
Termékmennyiségek*:			
Garancia*:	Hat hónap		
Kategória:		and the second se	
Csak személyesen vehető a	it: 🗆		

26. ábra. Új szett feltöltése (részlet)

Ha egy terméknél vagy szettnél rákattintunk a szerkesztés-ikonra, akkor ugyanolyan űrlap fog megnyílni, mint amit az előbb bemutattunk, csak éppen részben vagy egészben ki lesz már töltve. Nem minden mező értéke jelenik meg a terméklapokon, az ilyen mezők a rendszer továbbfejlesztésének megkönnyítését szolgálják, a következő verziókban kerülnek használatba.

2.2.2.1. Termékek gyors feltöltése

Lehetőség van termékek gyors feltöltésére néhány alapadat megadását követően (27. ábra). Ez egy adatforrásként használt, nagyméretű adattábla (ún. rendelési segédlista) segítségével történik. Csak olyan termék tölthető fel ezzel a módszerrel, amit a rendelési segédlista tartalmaz. Az erre szolgáló almenüt a termékek feltöltésénél találjuk a lap alján.

Termék feltöltés	se az adatforrásb	ól (rendelési segédlis	tából)	
Ennél a résznél gyorsan g	enerálhat egy új terméket a	z adatforrás (rendelési segédlista)	használatával.	
Cikkszám*:	Ár*:	Akciós ár:	Fénykép 1. (link)*:	
				Küldés
*Kötelező kitölteni!				
	27	. ábra. Termékek gy	ors feltöltése	

2.2.3. SZÁLLÍTÓK KEZELÉSE

Lehetőség van a rendszerben tárolt szállítók adatainak módosítására és új szállítók felvételére is. Ehhez kattintsunk a "Szállítók kezelése" menüre. Ekkor megnyílik egy lista, melynek első oszlopában a szállítók kódja, második oszlopában a szállítók neve, harmadik oszlopában az egyes szállítókhoz fűzött megjegyzések, negyedik oszlopában pedig a szerkesztés-ikon található.

A lista fejléce és lábléce ugyanolyan, mint a termékek listájánál, csak most a rendezési szempont a szállító kódja, illetve a szállító neve lehet. A keresőmező is ugyanúgy megtalálható a fejlécben. Mindezt a 28. ábrán láthatjuk.


28. ábra. A rendszerben bejegyzett szállítók listája

A szállítók közötti keresés és navigálás teljesen ugyanúgy működik, mint a termékek listájánál. Új szállító felvételéhez kattintsunk az "Új szállító" hivatkozásra. Ekkor megnyílik a szállítókhoz tartozó szerkesztési nézet. Ugyanez a nézet nyílik meg akkor is, amikor egy szállítónál rákattintunk a szerkesztés-ikonra, de abban az esetben már nem lehet átírni a szállító kódját, ezt csak új szállító felvételénél lehet beállítani. Új szállító felvételére és meglévő szállító szerkesztésére láthatunk példát a 29. ábrán:

Új szállító	Szállító szerkesztése	
Szállítókód*:	Szállítókód*: 2	
Megnevezés*:	Megnevezés*:	
Megjegyzések:	Megjegyzések:	
K	ildés	Küldés
*Kötelező kitölteni!	*Kötelező kitölteni!	

29. ábra. Új szállító hozzáadása, illetve meglévő szállító szerkesztése

Ha végeztünk az űrlap kitöltésével, kattintsunk a "Küldés" gombra a

mentéshez. Ezt követően a módosítások láthatóvá válnak a termékeknél, illetve az esetlegesen hozzáadott új szállító kiválasztható lesz a termékek szerkesztésénél.

2.2.4. RENDELÉSEK KEZELÉSE

Ha főmenüben rákattintunk a "Rendelések kezelése" menüre, akkor bejön a megrendeléseket tartalmazó lista, ugyanolyan fejléccel és lábléccel, mint a termékek vagy a szállítók listája. Itt a rendezési szempont a rendelési azonosító, a vásárló neve vagy a rendelés dátuma lehet. Alapértelmezés szerint csökkenő időrendben listázunk, tehát a legfrissebb megrendelés kerül a lista legelejére. Természetesen itt is keresgélhetünk a rendelések között a keresőmezővel.

A lista első oszlopában található a rendelési azonosító, a második oszlopban a megrendelő adatai (név, email-cím, telefon, szállítási cím, ha van), harmadik oszlopban a megrendelés állapota, negyedik oszlopban a fizetési mód, ötödik oszlopban pedig a megrendelés dátuma és ideje. Mindezt a 30. ábrán látjuk.

	Rendezés mint: R	tendelési azonosító, Vásárló neve, Ren-	delés dátuma Sorok m	egjelenítése: <u>10, 20, 30, 50, 100</u>
<u>#407551</u>	Ábrahám Róbert ar@fakemail.com +36301234567 1234 Budapest, Fake street 1.	Feldolgozás alatt	Előreutalás	2012-11-27 02:24:04
<u>#407550</u>	Ábrahám Róbert ar@fakemail.com +36301234567 1234 Budapest, Fake street 1.	Feldolgozás alatt	Előreutalás	2012-11-26 18:03:36
#407549	Ábrahám Róbert	Feladva	Előreutalás	2012-11-22 17:23:05

30. ábra. Megrendelések listája

A megrendelésekben szereplő adatok nem módosíthatók, csak a megrendelés állapota állítható át, az is csak abban az esetben, ha a megrendelés nincs lezárva, azaz állapota: "Feldolgozás alatt".

Nyitott rendelés állapotának módosításához kattintsunk az adott rendelés

azonosítójára. Ekkor megnyílik a megrendeléshez tartozó összes fontos adat (31. ábra).

Személyes adatok	Szállítási adatol	k	Számlázási a	adatok	
Vásárló neve: Ábrah	ám Róbert Átvevő neve:	Ábrahám Róbert	Szerződő fél nev	e: Á	brahám Róbe
Email: ar@fak	email.com Irányitószám:	1234	Irányitószám:		123
Telefonszám: +36	301234567 Város:	Budapest	Város:		Budape
	Cim:	Fake street 1.	Cim:	2	Fake street
Egyéb információk					
Fesztelés					
Átvétel típusa	Fizetés módja	Állaj	pot*		
Futárszolgálattal	Előreutalás	Feldolg	ozās alatt	•	
2012-11-26 18:03:36.935265					
Termékek					
Cilibration Managements		Garancia (hó	nap) Súly (kg)	Ár (HUF)	Mennyiség
Cikkszain Megnevezes				211	1
9068752 Fake - Cortina Vi	respohár - Poharak - 9068752	6			
9068752 Fake - Cortina Viz 18597479 Fake - Söröskorsó	zespohár - Poharak - 9068752 ódl - Sörös Poharak - 18597479	6 6	2:1:	229	1
9068752 Fake - Cortina Vi 18597479 Fake - Söröskorsó Átvétel helye	zespohár - Poharak - 9068752 6dl - Sörös Poharak - 18597479	6 6	2:1	229	
9088752 Fake - Cortina Vi 18597479 Fake - Söröskorsó Átvétel helye	zespohár - Poharak - 9068752 odl - Sörös Poharak - 18597479	6 6	2:14	229	1

31. ábra. Rendelési adatok megtekintése, állapot módosítása

Amennyiben egy megrendelés sikeres volt, a megrendelő a pénzt befizette, akkor átállíthatjuk a megrendelés állapotát "Feladva", "Átvéve", "Teljesítettnek tekintett", "Teljesítve" állapotok valamelyikére, az átvétel módjától függően, majd kattintsunk a "Küldés" gombra. Ekkor a rendszer automatikusan le is könyveli a megrendelésből származó bevételt a pénzügyi adatok közé. Ha az értékesítés nem volt sikeres, akkor a sikertelenség okának megfelelő állapotot beállíthatjuk, majd a "Küldés" gombra kattintva menthetjük a változtatást. Mentés után a rendszer ismét a megrendelések listájához navigál minket, üzenetben jelezve a művelet eredményét.

Amennyiben a megrendelés már lezárt állapotban van, akkor is bármikor megtekinthetjük az adatait, de az állapot megváltoztatásának lehetőségét már nem kínálja fel a rendszer, az rögzítve van.

2.2.5. Pénzügyi adminisztráció

Ha a főmenüben rákattintunk a "Pénzügy" menüre, akkor megnyílik a könyvelés, melyben az áruház pénzügyeit intézhetjük (32. ábra).

#18	Sikeres értékesítés	Megrendelés-azonosító: 407550	440 HUF	2012-11-27 17:34:41	Automatikusan generált bejegyzés
#17	Sikeres értékesítés	Megrendelés-azonosító: 407549	1140 HUF	2012-11-22 17:29:09	Automatikusan generált bejegyzés
#16	Sikeres értékesítés	Megrendelés-azonosító: 406024	39687 HUF	2012-11-21 00:06:37	Automatikusan generált bejegyzés
#15	Sikeres értékesítés	Megrendelés-azonosító: 407548	13865180 HUF	2012-11-19 03:09:55	Automatikusan generált bejegyzés
#14	Sikeres értékesítés	Megrendelés-azonosító: 406030	6076 HUF	2012-11-17 21:48:35	Automatikusan generált bejegyzés
#13	Sikeres értékesítés	Megrendelés-azonosító: 406509	11486 HUF	2012-11-17 21:46:58	Automatikusan generált bejegyzés
#12	Lapozás-teszt	Lapozás tesztelése	120000 HUF	2012-11-17 21:46:11	Tesztelés
#11	Utalás	Most már elvileg lapoznia kell	200000 HUF	2012-11-17 21:45:42	Tesztelés
#10	Szállítási díj	Újabb levonás	-34000 HUF	2012-11-17 21:44:57	Csak teszteléshez
#9	Levonás	Levonás tesztelése	-12000 HUF	2012-11-17 21:44:03	Tesztelés
		1	2		-

32. ábra. Könyvelés - tranzakciók és egyenleg megtekintése

A tranzakciók listája az eddig ismertetett listákhoz nagyon hasonló, a fejléc és a lábléc ugyanolyan funkciót tölt be, mint eddig, az egyetlen különbség az oldal alján található űrlap: ennek segítségével új tranzakciót rögzíthetünk a rendszerben. A tranzakciós lista első oszlopában a tranzakció azonosítószáma található, a másodikban a tranzakció megnevezése, a harmadikban a tranzakció leírása, negyedikben a tranzakció tárgyát képező pénzösszeg, ötödik oszlopban a tranzakció dátuma és ideje, utolsó oszlopában pedig a tranzakcióhoz fűzött megjegyzések találhatók. A keresőmező segítségével bármilyen tranzakcióra rá tudunk keresni.

A tranzakciókat csak egyszer lehet bevinni, lekönyvelés után már nem módosíthatóak. A tranzakciók felvételéhez használt űrlapot a 33. ábrán láthatjuk.



33. ábra. Új tranzakció felvétele, lekönyvelése

Új tranzakció felvételekor a megjegyzések kivételével az űrlap minden mezőjének kitöltése kötelező. Sikeres könyvelés esetén külön értesítés nem jelenik meg az oldalon, hanem az újonnan lekönyvelt tranzakció a lista legelején, illetve az egyenlegben láthatóvá válik. Sikertelen könyvelés esetén az oldal tetején hibaüzenet jelenik meg.

Megrendelések lekönyvelésekor a rendszer automatikusan kitölti nekünk a könyveléshez szükséges adatokat: a tranzakció megnevezése "Sikeres értékesítés" lesz, a leírásba bekerül a megrendelés azonosítószáma, az összeghez a vételár számolt összege kerül, a megjegyzésekbe pedig bekerül, hogy "Automatikusan generált bejegyzés". A többi értéket a rendszer szintén automatikusan generálja.

Figyelem: a korábban említett okok miatt a rendszer külön nem kezeli a szállítási díjakat, ezek automatikusan nem kerülnek lekönyvelésre a megrendelésekkel együtt, nekünk kell gondoskodnunk arról, hogy a szállítási díjakból származó bevétel megfelelően le legyen könyvelve!

2.2.6. FÓRUM AZ ELADÓI KÖZPONTBAN

Az eladói központból a fórumot kétféleképpen nyithatjuk meg: a főmenüben

rákattintunk a "Fórum" menüre, vagy keresünk egy olyan terméket, amelyhez a vásárlói oldalon keresztül érkezett hozzászólás és ott rákattintunk a "Fórum" fülre.

A vásárlói oldali fórumhoz képest az első szembetűnő különbség az lehet, hogy az eladói központból önálló hozzászólást nem lehet írni, kizárólag más hozzászólásokra lehet válaszolni. Ez a korlátozás fogyasztóvédelmi szempontok alapján született, az esetleges visszaélések elkerülése végett: ha az eladók új információt akarnak megosztani az oldal látogatóival egy adott terméket illetően, akkor azt a termék leírásához kell hozzáfűzni, ahol mindenki láthatja.

A második szembetűnő különbség a vásárlói oldalhoz képest az, hogy az eladói központban már megjelenik az egyes hozzászólók email-címe is. Ezáltal az eladók egyszerűbben fel tudják venni az érdeklődőkkel a kapcsolatot, amennyiben valamilyen különleges kéréssel fordulnak az áruház felé.

Ha a főmenüből nyitjuk meg a fórumot, akkor nemcsak 1-1 termékhez tartozó hozzászólásokat láthatunk, hanem időrendben csökkenő sorrendben az összes eddigi hozzászólást is, ami jelentősen leegyszerűsíti az oldal látogatói által feltett kérdések megválaszolását. A főmenüből elérhető fórumot a 34. ábrán láthatjuk.

Cikkszám: 18596702		
Másik válasz az eladói központból a termékla	ipon keresztül	
Chip*Sortiert*15x18cm Mosdókesztyű Cikkszám: 18596702	#2 Fake Lakberendezési Kft. Budapest válaszolta #1:	2012-11-15 23:39:20
Válasz az eladói központból a fórum-menün	keresztül	
Chip*Sortiert*15x18cm Mosdókesztyű Cikkszám: 18596702 Továbbfejlesztett főrum tesztelése	#1 Ábrahám Róbert íria:	2012-11-15 23:37:51
Áruház*	Üzenet szövege*	
Fake Lakberendezési Kft. Válasz erre* #6 (Abraham Robert)		Kuldes
*Kötelező kitölteni!		

34. ábra. Fórum elérése az eladói központból

Egy hozzászólásra történő reagáláskor az ehhez szükséges űrlap minden mezőjének kitöltése kötelező, bár itt kicsit kevesebb mezővel kell foglalkoznunk, ugyanis az áruház adatai megtalálhatók a rendszerben, így azokat nem kell kézzel beírni, csak arra kell ügyelni, hogy a megfelelő telephely azonosítójával válaszoljunk a megfelelő hozzászólásra, amiket a hozzájuk tartozó legördülő menükből választhatunk ki. Az üzenet szövegének itt is legalább 1 értelmes karaktert tartalmaznia kell, és nem tartalmazhat speciális karaktereket vagy HTML-kódot.

Nem megfelelő kitöltöttség esetén a vásárlói oldalhoz hasonlóan hibaüzenetet kapunk. Sikeres küldés esetén nincs külön visszajelzés, hanem a hozzászólásunk megjelenik a fórumon a legelső sorban.

2.2.7. Adatok importálása/exportálása

Információs rendszerünk lehetőséget biztosít különféle adatok importálására és exportálására, ami más webáruházakkal történő együttműködést hivatott megkönnyíteni. Ezt az "Import/Export" menüben tehetjük meg (35. ábra).

Import/Expor	·t		
Ezen az oldalon lehetősége va	n adatok feltöltésére és letöltés	ére az adatbázisban.	
Grando terméklist	a letöltése		
Ennél a résznél letöltheti a ter	méklistát a Grando webáruház	hoz, feltőltésre kész formátumban.	
Grando terméklista letőltése			
A date and a formed at	(
Adatiorras (rende	test segeonsta) impo	ortalasa	
Ennél a résznél importálhatja	az adatforrást, mely segítségév	vel az összes termékadat automatikusan friss	al.
Ennél a résznél importálhatja Fájlnév: /public/	az adatforrást, mely segítségév	el az összes termékadat automatikusan friss Karakterkódolás: 180-8859-2	ül.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztó: Tabulátor	az adatforrást, mely segítségév • Szöveghatároló: *	vel az összes termékadat automatikusan friss Karakterkódolás: 180-8859-2 Kifutó termékek jelölése: +	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztó: Tabulátor Nagyméretű objektumok: 🗹	az adatforrást, mely segítségév • Szöveghatároló: * Adatforrás (rendelési segédlista)	vel az összes termékadat automatikusan friss Karakterkódolás: 180-8859-2 Kifutó termékek jelölése: +) importálása	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztó: Tabulátor Nagyméretű objektumok: S	az adatforrást, mely segítségév • Szöveghatároló: • Adatforrás (rendelési segédiista)	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: + jmpottátása	al.
Ennél a résznél importálhatja Fájlnév: [public/ Elválasztó: Tabulator Nagyméretű objektumok: & Megrendelések im	az adatforrást, mely segítségév • Szöveghatároló: Adatforrás (rendelési segédiista) portálása/exportálá	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: +) importálása Ása	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztó: Tabulator Nagyméretű objektumok: I Megrendelések im Ennél a résznél importálhatja	az adatforrást, mely segítségév • Szöveghatároló: Adatforrás (rendelési segédilsta) portálása/exportálá /exportálhatia a Grando vebár	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: +) importálása ÁSA	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztó: Tabulator Nagyméretű objektumok: S Megrendelések im Ennél a résznél importálhatja	az adatforrást, mely segűségév Szöveghatároló: Adatforrás (rendelési segédilsta) portállása/exportállá (exportálhatja a Grando webár	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: +) importálasa ása uuházzal kompatibilis megrendelőlistát.	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztő: [Tabulátor Nagyméretű objektumok: S Megrendelések im Ennél a résznél importálhatja. Megrendelések letöttése	az adatforrást, mely segűségév Szöveghatároló: Adatforrás (rendelési segédilsta) portálása/exportálł /exportálhatja a Grando webár	el az összes termékadat automatikusan friss Karakterkódolás: 150-8659-2 Kifutó termékek jelölése: +) importálása ÁSA	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztő: Tabulator Nagyméretű objektumok: I Megrendelések im Ennél a résznél importálhatja. Megrendelések ietöttése Megrendelések feltőltése	az adatforrást, mely segűségév • Szöveghatároló: Adattorrás (rendelesi segedilsta) portálása/exportálá /exportálhatja a Grando webár	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: +) Importátása Ása ruházzal kompatibilis megrendelőlistát.	al.
Ennél a résznél importálhatja Fájlnév: /public/ Elválasztő: Tabulator Megrendelések im Ennél a résznél importálhatja Megrendelések letöltése Fájlnév: /public/	az adatforrást, mely segűségév • Szöveghatároló: Adattorrás (rendelesi segédista) portálása/exportálá /exportálhatja a Grando webár	el az összes termékadat automatikusan friss Karakterkódolás: 150-8859-2 Kifutó termékek jelölése: +) importálása ása ruházzal kompatibilis megrendelőlistát.	al.

35. ábra. Import/Export lehetőségek áttekintése

2.2.7.1. Grando terméklista letöltése

Lehetőség van olyan formátumú terméklista letöltésére, amely bemenete lehet a Grando webáruháznak, így az összes termékünket könnyedén fel tudjuk tölteni oda. Ennek a terméklistának az exportálását a "Grando terméklista letöltése" gombra kattintva tehetjük meg az ide vonatkozó résznél.

2.2.7.2. Adatforrás (rendelési segédlista) importálása

Importálhatjuk az adatforrásként szolgáló állományt (rendelési segédlistát), ami egy olyan CSV táblázat, melyben rengeteg termék adata található meg – előre egyeztetett koncepció alapján. Ennek az adatforrásnak segítségével tömeges frissítést hajthatunk végre termékeinken, illetve ennek az állománynak segítségével gyors feltöltést is végezhetünk az eladói központban. Az állomány feltöltését az "Adatforrás (rendelési segédlista) importálása" résznél végezhetjük el. A fájlnév résznél meg kell adni az állomány nevét, megfelelően be kell állítani a fájl tulajdonságait, majd az "Adatforrás (rendelési segédlista) importálása" gombra kattintva pedig elindíthatjuk az importálást és a tömeges frissítést. Alapértelmezés szerint a leggyakrabban előforduló tulajdonságok vannak beállítva, de ezeket nem árt feltöltés előtt ellenőrizni. A következőkre kell figyelni a frissítés során:

- A fájlnév csak valódi, létező fájl lehet, ellenkező esetben hibaüzenetet kapunk. Ezt a fájlt a rendszergazdák által megadott helyre kell feltölteni és az így kapott elérési útvonalat kell megadni a fájlnévnél;
- A karakterkódolást is ellenőrizni kell. Ha nem megfelelő karakterkódolást használunk, hibás karakterek jelenhetnek meg a termékek adatainál! Alapértelmezés szerint ISO-8859-2-es karakterkódolás van beállítva, ez a legtöbb esetben megfelelő, de előfordulhat, hogy más

karakterkódolással kapjuk meg az állományt;

- Az elválasztó alapértelmezés szerint a tabulátor, a legtöbb esetben ez a beállítás is helyes, de itt is elképzelhető, hogy más karakterrel tagolt a kapott állomány, így ezt is meg kell nézni. Ennek elmulasztása sikertelen feltöltést eredményezhet;
- A szöveghatárolóra is ugyanaz vonatkozik, mint az előbbiekre, erre is figyelni kell;
- A kifutó termékeket általában a cikkszámok után írt "+" karakter jelzi. Ezt szinte soha nem kell átállítani, mivel azonban a jövőben ez sem mindig garantálható, így lehetőség van más karakter megadására is. Ha rossz karaktert adunk meg, akkor a kifutó termékek nem kerülnek megkülönböztetésre;
- Nagyméretű objektumok: általában egy megfelelően részletes, teljes rendelési segédlista 40-50 MB körüli méretekkel rendelkezik. Ekkora állomány feldolgozása rengeteg időt venne igénybe normál módszerrel, ezért ennél a jelölőnégyzetnél jelezhetjük a rendszernek, hogy sok adattal kell dolgoznia, így ennek megfelelő módszerrel fogja az állományt feldolgozni, hogy elfogadható idő alatt végezzen. Felhívjuk a figyelmet, hogy a nagyméretű állományok importálására szolgáló módszer több, nagyméretű ideiglenes fájlt generál, amiket a frissítés lefuttatása után nekünk kell eltávolítani, ezen kívül, а jelölőnégyzetet kizárólag teljes rendelési segédlistánál szabad használni, különben a rendelési segédlistában nem szereplő termékek eltűnnek az adatbázisban tárolt adatforrásból, ami helytelen frissítést eredményezhet! Nem teljes, vagy kisméretű rendelési segédlista esetén vegyük ki a pipát a jelölőnégyzetből, ekkor a rendszer egyszer megy végig a kapott állomány minden egyes során és csak az abban szereplő termékek adatait fogja átírni az adatbázisban, a többi termék adata változtatás nélkül megmarad. Itt viszont arra kell

felhívnunk a figyelmet, hogy ez a módszer nem alkalmas nagyméretű állományok kezelésére, mert ez rengeteg időt venne igénybe! Ha túllépjük az időkeretet (ami alapértelmezés szerint 1 óra), akkor ez sikertelen frissítést eredményez! Ügyelnünk kell tehát az importáláskor megadott módszer gondos megválasztására a sikeres frissítés érdekében!

A frissítési folyamatot követően a rendszer kiírja a frissítés eredményét a futási idővel együtt. Egy frissítés lehet sikeres, részben sikeres vagy sikertelen. Részben sikeres frissítés esetén a frissítés ugyan lefutott, de a frissítéskor bizonyos termékeknél problémák léptek fel, így ezek a termékek letiltásra kerültek. Általában árkülönbségek miatt fordul ilyen elő, tehát ellenőrizni kell ezeknek a termékeknek az árait, korrigálni őket, szükség esetén pedig újra lefuttatni a frissítést. Az így letiltott termékek adatai a frissítés befejezésekor kilistázásra kerülnek.

2.2.7.3. Megrendelések importálása/exportálása

A "Megrendelések letöltése" gombra kattintva letölthetjük CSV állományként az adatbázisban tárolt megrendeléseket. Ezen kívül, importálhatunk is a rendszerbe korábbi megrendeléseket. Az importálandó megrendeléseket a Grando webáruházban letölthető megrendelések formátumában kell megadni, az állomány tulajdonságait pedig a rendelési segédlistánál látottakhoz hasonlóan ellenőrizni kell, szükség esetén átállítva az űrlap megfelelő elemeit.

Ha ezzel készen vagyunk, akkor kattintsunk a "Megrendelések feltöltése" gombra az importálás elindításához. Sikeres importálás esetén a statisztikai adatoknál is meg kell jelennie a változásoknak. Ez a szolgáltatás főleg a havi forgalommal kapcsolatos kimutatások elkészítésének megkönnyítésére lett implementálva.

Míg importáláskor főleg ISO-8859-2-es karakterkódolással találkozunk,

exportáláskor a kimeneti állományok kivétel nélkül mindig Unicode UTF-8-as formátumban kerülnek mentésre. Erre azért van szükség, mert ez a nemzetközi szabvány, és így biztosak lehetünk abban, hogy exportáláskor megfelelően kerülnek mentésre a speciális karakterek, illetve ékezetes betűk is.

2.2.8. KÜLÖNLEGES AKCIÓK

Beállíthatunk különleges akciókat az adatbázisban tárolt termékekre. Ennek lényege, hogy az adatforrásban tárolt, illetve onnan frissítésre kerülő adatok hatálytalanítva lesznek, és az általunk megadott akciók kerülnek bele a terméklistába frissítéskor. Különleges akciók definiálására a "Különleges akciók" menüben van lehetőségünk (36. ábra).

			Rendezės mint:	Azonositó, <u>Cikkszám</u>	. <u>Akciós ár</u> . <u>Készl</u>	et. <u>Kezdés dituma</u> . <u>Befejezés dituma</u> S	orok megjelenitë	se: <u>10, 20, 30</u> ,
Azonositó	Cikkszám	Akciós ár	Készlet	Kezdés dátuma	Befejezés dátuma	Megjegyzések	Promóció	Művelet
11266	18735699	20 HUF	-1	2013-04-19	2013-04-22	teszt	Igen	Ettivolitis
1265	18735699		-1	2012-09-10	2012-11-12	Valódi akció volt, véletlenszerüsítve lett.	Nem	Ehivolitis
1264	19450535	23405 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítve lett.	Nem	Eltivolitis
1262	19384065	18152 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítve lett.	Nem	Ehivolitis
1261	19371484	29487 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítve lett.	Nem	Eluvolitis
1260	19370618	46605 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítve lett.	Nem	Eltivolitis
1259	19370569	9584 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítye lett.	Nem	Ehivolitis
11238	19370535	5887 HUF		2012-09-03	2012-10-14	Valódi akció volt, véletlenszerűsítve lett.	Nem	Ehivolitis
¥1258 ¥1257	19370535 19370527	5887 HUF 41808 HUF		2012-09-03	2012-10-14 2012-10-14	Valódi akció volt, véletlenszerűsítve lett. Valódi akció volt, véletlenszerűsítve lett.	Nem Nem	Ehivolitis Ehivolitis
4258 41257 41256	19370535 19370527 19370478	5887 HUF 41808 HUF 24599 HUF		2012-09-03 2012-09-03 2012-09-03 12	2012-10-14 2012-10-14 2012-10-14 2012-10-14 <u>3 - 127</u>	Vakki akcis vok veletlenzerinitwe ket. Vakki akcis vok veletlenzerinitwe ket. Vakki akcis vok, veletlenzerinitwe ket.	Nem Nem	Ehivolitis Ehivolitis
vi258 vi257 vi256 Ú J akcló innél a résznél	19370535 19370527 19370478	5887 HUF 41808 HUF 24599 HUF	vagy menny liežget kök	2012-09-03 2012-09-03 2012-09-03 12 12 cle25 megadni.	2012-10-14 2012-10-14 2012-10-14 2012-10-14 3 - 127	Vakidi akcisi voli, veletlenarerinine lett Vakidi akcisi voli, veletlenarerinine lett Vakidi akcisi voli, veletlenarerinine lett	Nem Nem Nem	Ehivolitis Ehivolitis Ehivolitis
11258 11257 11256 Jj akcló mel a résztél	19370335 19370527 19370478	5887 HUF 41908 HUF 24599 HUF	vagy menny inéget kök Megjegyzések	2012-09-03 2012-09-03 2012-09-03 1 2 1 2 cle25 megadni	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2 2 - 127	Vakki akcis volt, veletlenarenistve ket. Vakki akcis volt, veletlenarenistve ket. Vakki akcis volt, veletlenarenistve ket.	Nem Nem	Elivolitis Elivolitis Elivolitis
11258 11257 11256 Jjakcló innél a résznél Cikkozám* Akciós ár (Hl	19370335 19370527 19370478 Herekochat egr új ka	SSST HUF 415005 HUF 24599 HUF	vogy mennyiniget kö	2012-09-03 2012-09-03 2012-09-03 12 12 clcző megadni	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2 - 127	Vakki akcis volt, veletlenarenistre ket. Vakki akcis volt, veletlenarenistre ket. Vakki akcis volt, veletlenarenistre len.	Nem Nem	Elivolitis Elivolitis Elivolitis
11238 11257 11256 Új akcló innél a résznél Cidocsim* Akciós ár (HI Készlet	19370335 19370527 19370478 Härekashat ege äj ki	SSST HUF 415005 HUF 24599 HUF	vege menny beiget hör	2012.09.03 2012.09.03 2012.09.03 1 <u>2</u> elező megadni	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2-127	Vakki akcis vali, veledenarerinine lett Vakki akcis vali, veledenarerinine lett Vakki akcis vali, veledenarerinine lett	Nem Nem	Elixolitis Elixolitis Elixolitis
VI255 VI257 VI256 ÚJ akcló innél a résztél Cidoxim* Akciós ár (HI Készlet (-1: Csak rend	19370335 19370527 19370478 letrehochat ego ilj ka UEP	SS7 HUF 41508 HUF 24599 HUF	vegy mennytelget kö	2012.09.03 2012.09.03 2012.09.03 2012.09.03 12 clező megadni	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2-127	Vakki akcis vali, veletlemerelinine lett Vakki akcis vali, veletlemerelinine lett Vakki akcis vali, veletlemerelinine lett	Nem Nem	Ehivolinis Ehivolinis Ehivolinis
11238 11257 11256 111 akcló 111 akcl	9370535 19370527 19370527 19370678 	SST HUF 41808 HUF 24599 HUF 24599 HUF	vago mennyiséget kö Megjegyzések	2012-09-03 2012-09-03 2012-09-03 2012-09-03 12 12 elezó megaditi	2012-10-14 2012-10-14 2012-10-14 2012-10-14 21-127	Vakidi akciš voli, veletlenarerinime lett Vakidi akciš voli, veletlenarerinime lett Vakidi akciš voli, veletlenarerinime lett	Nem Nem Nem	Ehivolinis Ehivolinis Ehivolinis
11238 11257 11257 11256 113 114 114 114 114 114 114 114	9370335 19370357 19370478 	SST HUF 41808 HUF 24599 HUF 26599 HUF	vage menny iniget his	2012-09-03 2012-09-03 2012-09-03 2012-09-03 12 12 cleast megaditi	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2-122	Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett.	Nem Nem Nem	Ehivolitis Ehivolitis
11258 11256 11	9370355 1937057 1937057 19370478 <i>Hétrohochat ege új ká</i> 19770478 19370478 19370478	SST HUF 41808 HUF 24599 HUF	vage menny beiget his	2012-09-03 2012-09-03 2012-09-03 2012-09-03 12 12 clead megadhi	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2-127	Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett.	Nem Nem	Ehivolitis Ehivolitis
11258 11256 11	9370355 9370355 9370478 9370478 9370478 9370478 9370478 9370478 9370478 9370478 93704578 937055 93705 93705 	SST HUF 41508 HUF 24599 HUF	vage menny brigget his	2012-09-03 2012-09-03 2012-09-03 2012-09-03 12 12 clead megadhi	2012-10-14 2012-10-14 2012-10-14 2012-10-14 2-127	Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett. Vakidi akciš voli, veletlenarerinime lett.	Nem Nem	Ehivolitis Ehivolitis Ehivolitis
1238 1257 1256 Új akcló Smel a résznél Cikkerám* Akciós ár (Hl Készlet (Alins: 1980). Refejezés dá (Mins: 1980).	9370555 1937057 1937057 1937057 19370478 19	SST HUF 41008 HUF 24599 HUT	vagr menny helget kös	2012-00-03 2012-00-03 2012-00-03 2012-00-03 12 ctcs5 megadai	2012.10.14 2012.10.14 2012.10.14 2012.10.14 2012.10.14 3 - 127	Vakki akcis vali, veledenarerinine lett. Vakki akcis vali, veledenarerinine lett. Vakki akcis vali, veledenarerinine lett.	Nem Nem Nem	Ebivolitis Privolitis Ebivolitis

36. ábra. Különleges akciók kezelése

A lap tetején a szokásos szűrhető, rendezhető, lapozható lista formátumában

jelennek meg a különleges akciók által érintett termékek. A lista első oszlopában az akció azonosítószáma, a második oszlopban az érintett termék cikkszáma, a harmadikban az általunk meghatározott akciós ár, a negyedikben a kívánt készlet, az ötödikben a kezdés dátuma, a hatodikban a befejezés dátuma, a hetedikben az akcióhoz fűzött megjegyzések, a nyolcadikban a promóciós jelzés, végül pedig az adott akciókon elvégezhető művelet jelenik meg.

Akciók létrehozhatók, törölhetők, de utólag nem módosíthatók, hibás akció esetén tehát törölnünk kell az érintett rekordot és újra létrehozni az akciót. Akciót törölni az érintett tételhez tartozó "Eltávolítás" hivatkozásra kattintva lehet.

Új akció létrehozásakor mindig meg kell adni az akció által érintett termék cikkszámát és a kezdés dátumát, továbbá a következőkre is figyelni kell:

- Az akciós ár és a készlet közül az egyiket mindenképpen meg kell adni, mert csak így van értelme akciót létrehozni. Készletet általában akkor adunk itt meg, ha bizonyos termékek készletét rendelhetőre akarjuk beállítani;
- A befejezés dátumát megadni nem kötelező, viszont ha nem adjuk meg, akkor a létrehozott akció határozatlan időre fog szólni, ellenkező esetben csak a megadott dátumig;
- Ha a promóciós jelölőnégyzet ki van pipálva, akkor a megjegyzés-részt kötelező kitölteni. Ilyenkor az a szöveg, amit a megjegyzés részbe beleírunk, az belekerül a Grando-s terméklista megnevezéseibe is, annak érdekében, hogy az érintett termékekhez gyűjtő linkeket lehessen létrehozni. Az akció lejárta után, ha frissítjük a termékeket, ez a szöveg automatikusan ki is kerül a Grando-s terméklista megnevezéseiből a listában. Az oldal hagyományos használata során ez nem jelenik meg a termékeknél, csak a Grando-s exportot érinti;

 Ha az akció nem promóciós, akkor a megjegyzéseket nem kötelező kitölteni.

Ha rendesen kitöltöttük az akcióhoz tartozó adatokat, kattintsunk a "Küldés" gombra a tétel rögzítéséhez.

Ahhoz, hogy a különleges akcióknál létrehozott tételek megjelenjenek a termékek adatainál, a termékeket frissíteni kell. Ezt az oldal alján megtalálható "Termékek frissítése" hivatkozásra kattintva megtehetjük.

3. FEJLESZTŐI DOKUMENTÁCIÓ

A következőkben részletesen ismertetjük információs rendszerünk felépítését, működését, illetve kitérünk a rendszer továbbfejlesztésének lehetőségeire, illetve arra, hogy a könnyebb fejleszthetőség érdekében milyen extra szolgáltatásokat építettünk be a rendszerbe.

A megfelelő áttekinthetőség érdekében először a rendszer egészére adunk betekintést, majd ezt követően részletezzük a rendszert alkotó alrendszerek felépítését, működését. Legelőször azonban tisztázzuk, hogy a rendszer megfelelő működésének milyen előfeltételei vannak, és hogyan kell beállítani a rendszert.

3.1. A RENDSZER TELEPÍTÉSE ÉS ÜZEMBE HELYEZÉSE

Információs rendszerünket Ubuntu Linux 10.10-es operációs rendszerre fogjuk telepíteni, Apache II. webszerverre (PHP és PostgreSQL támogatással) és PostgreSQL 8.4-es adatbázisszerverre. Ezen kívül szükséges még egy Java futtatókörnyezet telepítése is (mi a Sun-féle Java kiadást javasoljuk), és az, hogy a webszerver és az eléréshez használt böngésző támogassa a Java-applet-ek futtatását. Természetesen újabb verziókon is működik a rendszer, de mivel a fejlesztést ebben a környezetben végeztük, így ebben a környezetben is mutatjuk be. A leírásban eltekintünk az előbb említett szolgáltatások telepítésének részletezésétől, végig feltételezzük, hogy ezek a szolgáltatások megfelelően vannak telepítve, beállítva és tökéletesen tudnak egymással kommunikálni.

Először a rendszer magját képező adatbázis telepítését részletezzük, majd ezt követően kitérünk a webes felület üzembe helyezésére és beállítására.

3.1.1. AZ ADATBÁZIS TELEPÍTÉSE

Az adatbázist a dokumentációhoz mellékelt legfrissebb biztonsági mentésből fogjuk telepíteni, mert abban 100%-ig minden benne van, ami az adatbázis megfelelő működéséhez szükséges. Ennek az állománynak a neve a következőképpen néz ki: CommercialDatabase_YYYY_MM_DD.backup, ahol YYYY_MM_DD a mentés készítésének dátumát jelöli (YYYY az évet, MM a hónapot, DD pedig a napot).

A biztonsági mentés készítésekor már meglévő adatbázist és tulajdonost feltételeztünk, így nem került bele az ún. "CREATE DATABASE statement", azaz, ha az adatbázis nem létezik, akkor az adatbázisszerver nem fogja megpróbálni létrehozni azt! Továbbá, ha nincs létrehozva az "admin" nevű rendszergazdai szerepkör az adatbázisszerveren, akkor egyáltalán nem garantálható a rendszer megfelelő működése!

Az adatbázis telepítéséhez használjuk a PostgreSQL szerverhez telepíthető PgAdmin III. nevű, kényelmes, grafikus adminisztrációs felületet.

Amennyiben a nulláról telepítjük adatbázisunkat, akkor legelőször hozzunk létre egy "admin" nevű "login role"-t (szerepkört) a PostgreSQL adatbázisszerverünkön és ruházzuk fel az összes létező jogosultsággal! Ezt követően hozzunk létre egy "CommercialDatabase" nevű (üres) adatbázist, ahol tulajdonosnak állítsuk be az előbb létrehozott "admin" nevű felhasználót!

Ha a fentebb említett két adatbáziselem létrehozása megtörtént, akkor PgAdmin III. alól nyissuk meg a "CommercialDatabase" nevű adatbázist, a jobboldali egérgombbal kattintsunk az adatbázisra és a "Restore..." parancs segítségével nyissuk meg a helyreállító ablakot! Itt válasszuk ki a telepíteni kívánt adatbázist tartalmazó biztonsági mentést, majd kattintsunk az "OK" gombra a telepítéshez (37. ábra).



37. ábra. Az adatbázis telepítése

Amennyiben az említett felhasználó és adatbázis már létezik és a leírtaknak megfelelő kapcsolatban állanak egymással, akkor a helyreállító ablakban jelöljük be a "Clean before restore" jelölőnégyzetet. Ekkor tényleg helyreállítást csinálunk, a szerver kitakarítja a már meglévő adatbázist és a nekünk szükséges adatokkal tölti fel.

Amennyiben 8.4-es PostgreSQL szervert használunk a hozzá tartozó PgAdmin III. felülettel, akkor javasolt újracsatlakozni az adatbázishoz a telepítés/helyreállítás után, hogy biztosan jól jelenjenek meg a módosítások.

Ha nem kaptunk hibaüzenetet, akkor most már egy tökéletesen működő adatbázissal kell rendelkeznünk. Ha a művelet hibás volt, akkor korrigáljuk a beállításokat és próbáljuk meg újra, bejelölve a "Clean before restore" jelölőnégyzetet, hogy hibás vagy hiányos adat ne legyen az adatbázisban.

Figyelem: mivel a rendszert 8.4-es PostgreSQL szerver alatt készítettük, újabb verzión valószínűleg akkor is hibát fog jelezni a rendszer, ha egyébként minden beállítás megfelelő volt! Ennek oka az, hogy PostgreSQL 8.4 alatt külön paranccsal kell engedélyezni a PL/SQL nyelvet (pontosabban PLPGSQL nyelvet, ami a PostgreSQL saját implementációjú PL/SQL adatbázis-programozási nyelve), viszont a legújabb verziókban ez a szolgáltatás már alapból engedélyezve van, emiatt sem törölni, sem újraengedélyeztetni a helyreállításkor nem lehet. Ez nem is jelent igazi problémát, ha más hiba nem

50

lépett fel a művelet során, akkor ezt nyugodtan figyelmen kívül hagyhatjuk és bezárhatjuk a helyreállító-ablakot, a rendszer akkor is működni fog, feltéve, hogy a "Clean before restore" és "Verbose messages" jelölőnégyzeteken kívül más nincs bejelölve.

3.1.2. A webes felület telepítése, beállítása

Az adatbázis eléréséhez használt weboldal a commercial.zip fájlban található összecsomagolva (könyvtárral együtt csomagoltuk össze). Ha ez lesz az egyetlen weboldal a szerverünkön, akkor a főkönyvtár nem kell, ha csak egy al-oldal lesz, akkor igen. A mi szerverünkön más weboldalak is vannak, tehát mi megőrizzük a főkönyvtárat, és a /var/www könyvtárba helyezzük el azt a teljes tartalmával együtt. Ezen kívül – az elérést megkönnyítendő – létrehoztunk egy /homepage nevű szimbolikus linket is, ami az előbb említett könyvtárra mutat, tehát a továbbiakban Így is hivatkozhatunk információs rendszerünk weboldalainak helyére: /homepage/commercial.

Amikor a böngészőből akarjuk megnyitni a weboldalt, akkor a következőt kell beírnunk az URL-mezőbe: <u>http://localhost/commercial/</u>. De mielőtt ezt még megtennénk, be kell állítani a csatlakozási pontot (ún. "connection string"et) is.

A csatlakozási pont beállításához nyissuk meg a weboldalunk könyvtárában található config/connection.php fájlt szerkesztése. Ekkor a következő tartalmat kell látnunk:

```
<?php
```

```
//DATABASE CONFIGURATION FILE
//This file stores all connection data
$host="localhost";
$port="5432";
$dbname="CommercialDatabase";
$user="admin";
$password="admin";
```

```
//Import/Export directories
//All directories must exist and have read/write permissions
$export_directory="/public";
$import_directory="/public";
?>
```

Itt a következő módosításokat kell elvégeznünk:

- \$host="adatbázisszerver elérési címe";
- \$port="adatbázisszerver portja";
- \$password="admin felhasználó jelszava";
- \$export directory="adattáblák exportálási könyvtára";
- \$import directory="adattáblák importálási könyvtára".

Az export és import könyvtárak csak létező könyvtárak lehetnek és mindkettőre rendelkeznie kell az adatbázisszervernek írási és olvasási jogosultsággal! Javasolt, hogy a két könyvtár ugyanaz legyen (az egyszerűbb kezelés érdekében).

Ha adatbázisszerverünk telepítésekor elfogadtuk az alapértelmezett beállításokat, akkor az elérési cím: localhost, a port: 5432, ezeket nem kell akkor átírni. A jelszót viszont át kell írni arra, amit az "admin" szerepkör (vagy felhasználó) létrehozásakor megadtunk, az adatbázisnév (\$dbname) és felhasználónév (\$user) jól van beállítva.

Miután megfelelően módosítottuk a fájlt, mentsük el, zárjuk be és nyissuk meg webböngészőnkben a fentebb említett webcímet (amennyiben más könyvtárba tettük weboldalunkat, akkor azt írjuk be a localhost-rész után). Ekkor megnyílik böngészőnkben a vásárlói oldal (lásd: Felhasználói dokumentáció).

3.2. A RENDSZER FELÉPÍTÉSE

Információs rendszerünk alapvető felépítését nagyon nagy vonalakban a 3. ábrán láthattuk, ebben a fejezetben tovább részletezzük a felépítést, kitérve legfontosabb egységek működésére, szerepére. Először megnézzük az adatbázis felépítését, kitérve a legfontosabb táblázatokban használt mezők adattípusaira, tartalmára, az adatbázisban használt nézetekre és függvényekre, melyek összekapcsolják a lazábban egymáshoz kapcsolódó elemeket, illetve megnézzük a többi adatbázis elemet is. Ezt követően rátérünk a felület tárgyalására, a főbb szerkezeti elemekre és arra, ahogyan közvetíti az adatbázisban tárolt adatokat a felhasználó felé. A forráskódok részletes tárgyalását mellőzzük, mivel nagyon sok elem tartozik a rendszerhez és a konkrét forráskód megtalálható a dokumentumhoz mellékelt adathordozón. Inkább az egyes elemek szerepére fogunk részletesen kitérni.

3.2.1. AZ ADATBÁZIS FELÉPÍTÉSE

A rendszer lelkét alkotó adatbázisban négy fő sémát különböztetünk meg:

- 1. financial (adminisztrációs és pénzügyi jellegű adatok kezelése);
- 2. orders (rendelések, illetve azokkal kapcsolatos adatok kezelése);
- products (áruházlánccal, telephelyekkel és az ott kapható termékekkel kapcsolatos adatok kezelése);
- 4. public (egyéb, nyilvánosan megosztható adatok kezelése, illetve itt található az összes függvény, amit az adatbázisban használunk).

Helyenként "kis átfedések" előfordulhatnak az egyes sémák között, amikor két adatbázis elem logikailag különböző sémákhoz tartozna, de szerkezetileg

erősen összefüggenek vagy a működésük erősen egymásra épül. Ekkor inkább közvetlenül egymás mellett tároljuk őket.

Olyan is előfordul az adatbázisban, hogy egy adattábla egyik mezője egy másik adattáblának elemeire hivatkozik, még sincs idegen kulcsokkal megjelölve, hogy melyik elem melyik elemre hivatkozik. Ez két esetben fordulhat elő:

- Nem szabad idegen kulcsokat használni, mert az adattábla, ami a másik adattáblára hivatkozik, egyéb azonosítókat is tartalmazhat, amik ugyanannyira fontosak, mint a hivatkozott elemek, és logikailag ahhoz az adattáblához tartoznak, ami a másik adattáblára hivatkozik;
- 2. Nem tudunk idegen kulcsokat használni, mert az adattípusok eltérőek: egyik skalár, a másik pedig egy ugyanolyan adattípusú skalárokat tartalmazó tömb. Itt felhasználói függvények segítségével a tartalmazás ugyan ellenőrizhető, de az ilyen függvények megszorításokban történő használata kiszámíthatatlanná és instabillá teheti a rendszert, mert ha megszorításokban használjuk őket, akkor nem mindig adnak vissza jó eredményt az érintett adattábla változásai miatt!

Az adatbázis karakterkódolása legyen UTF-8, a használt táblatér (a könnyű "költöztetéshez") legyen pg_default, a területi beállítás legyen hu_HU.utf8. Ekkor – az eddigiek fényében – az adatbázist létrehozó SQL utasítássorozat lényegében a következő:

```
CREATE DATABASE "CommercialDatabase"

WITH OWNER = "admin"

ENCODING = 'UTF8'

TABLESPACE = pg_default

LC_COLLATE = 'hu_HU.utf8'

LC_CTYPE = 'hu_HU.utf8'

CONNECTION LIMIT = -1;

GRANT CONNECT, TEMPORARY ON DATABASE "CommercialDatabase" TO public;

GRANT ALL ON DATABASE "CommercialDatabase" TO "admin";

COMMENT ON DATABASE "CommercialDatabase" IS 'General commercial database';
```

A fenti beállítások tükrében már semmilyen problémát nem okozhat az ékezetes és különleges karakterek kezelése, de mivel a PHP felület helyenként egy az egyben visszaadja az adatbázisban tárolt adatokat, ezért érdemes ügyelni a tárolni kívánt adatok megfelelő formátumára vagy adattípusára, mert PHP alatt sok esetben a különleges karaktereknek és a HTML-kódoknak jelentése van, ami befolyásolja az oldal működését (a termékek leírásának megjelenítésénél például kihasználjuk ezt a tulajdonságot).

Mint korábban említettük, törekszünk arra, hogy a rendszerünk nemzetközi szinten használható legyen, így az adatbázis nyelve az Angol, ami természetesen független az adatbázisban tárolt adatok nyelvétől.

Az adatbázis teljes relációs adatmodelljét a 38-40. ábrákon láthatjuk.



38. ábra. Az adatbázis relációs adatmodellje (1)



39. ábra. Az adatbázis relációs adatmodellje (2)

2 M	products.grando_productlist
title	btrinv(((((products global text variables value *-*/.Text) products productist title. Text) r-*/.Text) products productist.teywords.Text) *-*.Text) products productist.tey words.Text) *-*.Text) *-*.Text) products productist.tey words.Text) *-*.Text) *Text) *Text)
seller_product_id	INTEGER
status	INTEGER
price	ROUND(products.productist.price)::Integer
discount price	ROUND(products productist discount price)::Integer
warranty	INTEGER
manufacturer	VARCHAR2 (255 CHAR)
weight	ROUND/products.product@st.weight)::Integer
category	INTEGER
disable_shipping	INTEGER
photo_url_1	VARCHAR2
photo_url_2	VARCHAR2
photo_url_3	VARCHAR2
photo_url_4	VARCHAR2
seller_category	VARCHAR2 (255 CHAR)
video_link	VARCHAR2
isbn13	INTEGER
gtin	INTEGER
product_code	INTEGER
warehouse_code	INTEGER
page_link	VARCHAR2
availability_283	ROUND(products productist, availability_283):/integer
shipping_fee_4_1	ROUND[products.productist.shipping_fee_4_1):-Integer
shipping fee 64 1	.mpl. shipping, enabled/products.productist.idl::Character Varying
description	CLOB
global_text_variab productist manufacturers virtualproducts	kes

40. ábra. Az adatbázis relációs adatmodellje (3)

3.2.1.1. Termékek adatainak kezelése

Az adatbázis legfontosabb adattábla- és nézetcsoportja a termékek adatait tároló adattáblák és a hozzájuk tartozó nézetek. Az ide tartozó adatbáziselemeket a 41-42. ábrákon láthatjuk.



41. ábra. A termékek adatainak relációs adatmodellje (1)

check_mpl_ship	ping_fee (min_weight, max_weight, recipient_point)		• email VARCHAR2 (255 0
			phone VARCHAR2 (20 C
		products.webadminlist	a houseid (id)
	products.webiist	id INTEGER	houseid_unique (id)
ad and a	INTEGER WARCHARD (OSS CHAR)	title VARCHAR2 (255 CHAR)	Checkhouses [btle, zp, cty, address
kon morde	VARCHAR2 (20 CHAR)	keywords VARCHAR2 (255 CHAR)	
status	INTEGER	status VARCHAR2 (255 CHAR)	
rice	ROUND(products.product(st.price)::Integer	price RUUND(products.productist.price)::integer	Laboratory ()
discount price	ROUND(products.productist.discount_price)::Integer	www.anhy VARCHAR2 (255 CHAR)	Integer
warranty	VARCHAR2 (255 CHAR)	manufacturar VARCHAR2 (255 CHAR)	products.not outgoing
manufacturer	VARCHAR2 (255 CHAR)	weight ROUND(products productist weight) Intege	H INTEGER
weight	ROUND(products.productlist.weight)::Integer	category INTEGER	IN THE DET
category	INTEGER	disable_shipping_INTEGER	datasource
disable_shipping	INTEGER	photo_url_1 VARCHAR2	ourgoing
photo_url_1	VARCHAR2	photo_url_2 VARCHAR2	
photo_url_2	VADCHAD2	photo_url_3 VARCHAR2	
photo url 4	VARCHAR2	photo_url_4 VARCHAR2	
seller category	VARCHAR2 (255 CHAR)	seller_category VARCHAR2 (255 CHAR)	
video link	VARCHAR2	VICEO_INK VAKUHAK2	products.update_product
isbn13	INTEGER	atia INTEGER	id INTEGER
gtin	INTEGER	nrocket code INTEGER	title VARCHAR2 (255
product_code	INTEGER	warehouse code INTEGER	keywords VARCHAR2 (255
warehouse_code	INTEGER	page ink VARCHAR2	status INTEGER
page_link	VARCHAR2	availability_283 ROUND(products.productlist.availability_283	Integer NUMBER
availability_283	ROUND(count_free_availability(products.productist.av	shipping fee_4_1 ROUND(products.productist.shipping_fee_4	1):Integer discount_price NUMBER
shipping_tee_4_1	ROUND(products.productist.shipping_tee_4_1)::Integer	description CLOB	availability_203 NUMBER
description	CEO8	product_group INTEGER	shipping_tee_4_LNOMBER
article aroun	INTEGER	article_group INTEGER	attide group INTEGER
aroos_groop	integer.	productist	reserved NUMBER
productist		status_informations	production.
manufacturars		warranties	producting
virtualproducts		manufacturers	
		virtuaproducts	
		products.grar	do_productlist
tie	btrim((((((products.global_text_variables.value ' - '::Tex	t) products.productist.title::Text) ' - '::Text) products.productis	: keywords: Text) " - "::Text) products.productlist.id) " "::Text) nul_to_empt
eller_product_id	INTEGER		
tatus	INTEGER		
rice irrough price	ROUND (products productist price)::Integer ROUND (products productist discount, price)::Integer		
arranty	INTEGER		
anufacturer	VARCHAR2 (255 CHAR)		
eight	ROUND(products productlist, weight)::Integer		
ategory	INTEGER		
isable_shipping	INTEGER		
noto_url_1	VARCHAR2		
noto_url_2	VARCHAR2		
noto_url_3	VARCHAR2		
hoto_url_4	VARCHAR2		
eler_category	VARCHAR2 (255 CHAR)		
hol 2	INTEGED		
tin	INTEGER		
rocket rocks	INTEGER		
arehouse code	INTEGER		
age link	VARCHAR2		
wailability_283	ROUND (products productlist availability_283)::Integer		
shipping fee 4 1	ROUND(products productlist shipping_fee_4_1)::Integer		
hipping_fee_64_1	mpl_shipping_enabled(products.productist.id)::Character V	(arying	
escription	CLOB		
to the second			
obal_text_variab	res		
oductist	R5		
oba_text_variab oductlist anufacturers	R5 Art		
poba_cext_variab productist manufacturers virtualproducts			

42. ábra. A termékek adatainak relációs adatmodellje (2)

Ezeknek az adattábláknak és nézeteknek mindegyike a products sémának része. Talán az egész adatbázis legfontosabb táblázata az önálló (vagy normál) termékek adatait tartalmazó adattábla, melynek neve productlist. Ebben megtalálható az áruházban kapható összes önálló termék, de a belőlük alkotott szettek már nem. Az adattábla a következő mezőkből épül föl:

Név	PostgreSQL adattípus	Leírás
id	INTEGER	A termék cikkszáma. Értéke 1 és 199999999 között, illetve 200000000 fölött lehet. Megadása kötelező. Elsődleges kulcs.
title	CHARACTER VARYING(255)	A termék megnevezése, fantázianeve. Megadása kötelező.
keywords	CHARACTER VARYING(255)	A keresés megkönnyítésére szolgáló kulcsszavak listája. Általában vesszővel vagy "/" jellel elválasztva szerepelnek. Megadásuk kötelező.

Egy lakberendező cég információs rendszere

Név	PostgreSQL adattípus	Leírás
status	INTEGER	 A termék állapota. Lehetséges értékei: -1: a termékkel probléma van, nem jelenik meg a vásárlói oldalon, a termék letiltott; 1: a termék megjelenik a vásárlói oldalon, ha van belőle szabad készlet vagy rendelhető, a termék engedélyezett; 2: a termék nem jelenik meg a vásárlói oldalon, letiltott. Megadása kötelező. Idegen kulcs, a status_informations adattábla id mezőjére hivatkozik. 1 az alapértelmezett értéke.
price	DOUBLE PRECISION	A termék normál ára. Nullánál nagyobb szám lehet. Megadása kötelező.
discount_price	DOUBLE PRECISION	A termék akciós ára. Ha üres, akkor a termék nem akciós, ha ki van töltve, akkor csak nullánál nagyobb, illetve a normál árnál kisebb szám lehet.
warranty	INTEGER	Garancia hónapokban. Megadása kötelező, alapértelmezés szerint 6.
manufacturer	INTEGER	A termék szállítójának/gyártójának a kódja. Megadása kötelező. Idegen kulcs, a manufacturers adattábla id mezőjére hivatkozik.
weight	DOUBLE PRECISION	A termék súlya kilogramban.
category	INTEGER	Belső kategóriakód. Egész szám lehet.
disable_shipping	INTEGER	Csak személyesen vehető-e át a termék. Ha üres, akkor nem, ha értéke 1, akkor igen.
photo_url_1	CHARACTER VARYING	Elsődleges kép. Értéke webcím lehet. Megadása kötelező.
photo_url_2	CHARACTER VARYING	Másodlagos kép. Értéke webcím lehet.
photo_url_3	CHARACTER VARYING	Másodlagos kép. Értéke webcím lehet.
photo_url_4	CHARACTER VARYING	Másodlagos kép. Értéke webcím lehet.
seller_category	CHARACTER VARYING(255)	Eladói kategória. Értéke beszédes kategórianév kell, hogy legyen. Elhagyható.
video_link	CHARACTER VARYING	A termékről készült videó linkje. Értéke webcím lehet.
isbn13	BIGINT	Hagyományos ISBN13-as azonosító. Megadása folytatólagosan történik.
gtin	BIGINT	Újfajta GTIN azonosító. Megadása folytatólagosan történik.

Név	PostgreSQL adattipus	Leírás
product_code	BIGINT	Termékkód. Megadása kötelező.
warehouse_code	BIGINT	Raktári azonosító. Megadása kötelező.
page_link	CHARACTER VARYING	Külső terméklap. Értéke webcím lehet.
availability_283	DOUBLE PRECISION	 Teljes készlet. Értéke a következő lehet: -1: a termék csak rendelésre van; Legalább nulla szám: tényleges raktári készlet. Megadása kötelező.
shipping_fee_4_1	DOUBLE PRECISION	Szállítási díj. Értéke nullánál nagyobb szám lehet. Megadása kötelező, alapértelmezett értéke 1750.
description	TEXT	A termék leírása. Lehet közönséges szöveg is, általában HTML kódot tartalmaz. Megadása kötelező.
product_group	INTEGER	Árucsoport (vagy főkategória). Értéke nullánál nagyobb egész szám lehet. Megadása kötelező. Idegen kulcs, a main_categories adattábla id mezőjére hivatkozik.
article_group	INTEGER	Cikkcsoport (vagy alkategória). Értéke nullánál nagyobb egész szám lehet. Megadása kötelező. Idegen kulcs, a sub_categories adattábla id mezőjére hivatkozik.
comments	TEXT	A termékkel kapcsolatos megjegyzések. Bármi lehet. A vásárlói oldalon nem jelenik meg.
reserved	DOUBLE PRECISION	Lefoglalt készlet. Megadása kötelező. Alapértelmezett értéke 0.
promotion_sign	CHARACTER VARYING(63)	Gyűjtő linkek létrehozásához plusz szöveget fűz hozzá a megnevezésekhez a Grando-s terméklistában, a termékek rendes megnevezéseiben ez nem jelenik meg. Automatikusan kezeli a program a különleges akciók függvényében.

1. táblázat. A terméklista felépítése

A terméklistára épül a virtuális termékek adattáblája, mely a virtualproducts névre hallgat, és terméklistában szereplő termékekből alkotott szetteket tartalmazza. Ennek az adattáblának a szerkezete teljesen hasonló a terméklista szerkezetéhez, de azért vannak különbségek, a következő táblázatban a virtuális termékek adattáblájához tartozó plusz mezőket foglaltuk össze:

Név	PostgreSQL adattípus	Leírás
discount	DOUBLE PRECISION	A szetthez kínált kedvezmény százalékban. Elhagyható.
products	INTEGER[]	A szetthez tartozó termékek cikkszámait tartalmazó tömb. Megadása kötelező.
product_quantities	DOUBLE PRECISION[]	A szetthez tartozó termékek mennyiségeit tartalmazó tömb. Megadása kötelező.

2. táblázat. A virtuális termékek adattáblájához tartozó egyéb mezők

A virtuális termékek adattáblájából hiányoznak a következő mezők:

- price: nincs rá szükség, a normál ár kiszámítható a szetthez tartozó termékek normál árainak összegzésével, sőt, így biztonságosabb;
- discount_price: nincs rá szükség, mert kiszámítható, ha a szetthez tartozó normál árra alkalmazzuk a discount mezőben megadott kedvezményt;
- manufacturer: fölösleges, mert a szetteket mi állítjuk össze tetszőleges termékekből, nincs külön szállítója. A szállítók közül a rendszer azt tekinti a szetthez tartozó szállítónak, akinek a legtöbb terméke tartozik a szetthez;
- weight: nincs rá szükség, mert kiszámítható a szetthez tartozó termékek súlyainak összegzésével;
- availability_283: nem használjuk, mert a meghirdetett szettek automatikusan rendelhető státuszt kapnak, így semmi szükség rá;
- reserved: mivel az availability_283 mezőt sem használjuk, erre sincs szükség;
- promotion_sign: szettekhez nem adunk meg külön promóciót, mert alapból be szoktunk rá állítani valamilyen akciós árat. Szettek esetében a rendszer nem foglalkozik a különleges akciók adattáblájával.

Mivel a szetthez tartozó termékek mennyiségeit a product_quantities mező tartalmazza, értelemszerűen ez a mező nem lehet üres, és a tömb elemszámának meg kell egyeznie a products tömb elemszámával. Ügyelnünk kell az elemek sorrendjére is, hogy a rendszer el tudja dönteni, hogy melyik termékből mennyit rendeltünk hozzá az adott szetthez. Bár valóban hasznos lenne, hogy idegen kulcsokat hozzunk létre a szettekhez tartozó cikkszámoknál, melyek a productlist adattáblára hivatkoznak, ezt sajnos nem tudjuk megtenni, mert eltérőek az adattípusok (lásd: 3.2.1 bevezető). A productlist és virtualproducts adattáblák között tehát nincs közvetlen relációs kapcsolat, a hozzájuk tartozó adatok konzisztenciájának megőrzéséért az adattáblák a helyzet a virtualproducts és manufacturers adattáblákkal is: itt is felhasználói függvények felelősek azért, hogy a weboldalon megfelelő szállító is megjelenjen a szetteknél.

A productlist és virtualproducts adattáblák által hivatkozott manufacturers, status_informations, main_categories és sub_categories adattáblák teljesen egyforma felépítésűek, szerkezetüket a következő táblázatban láthatjuk összefoglalva:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Azonosítószám. Megadása kötelező. Elsődleges kulcs.
title	CHARACTER VARYING(255)	Megnevezés. Megadása kötelező. Legalább 1 értelmes karaktert tartalmaznia kell.
comments	TEXT	Megjegyzések. Bármi jöhet ide, vásárlói oldalon nem jelenik meg.

3. táblázat. A terméklistákkal kapcsolatos segédtáblázatok felépítése

A táblák közötti egyetlen különbség, hogy míg a status_informations adattábla az azonosítószámnál bármilyen egész számot megenged, addig a másik három táblázatnál az azonosítószám értéke legalább 1 kell, hogy legyen.

A webes felületen a termékeknél nem kódokat, hanem értelmes adatokat

szeretnénk megjeleníteni, hogy az oldal esztétikus legyen. Ugyanakkor nem akarjuk azt sem, hogy emiatt túl bonyolult legyen az oldal kódja, és duplikálni sem akarjuk az adatbázisban tárolt elemeket ehhez, tehát létrehozunk két nézetet az adatok olvasható megjelenítéséhez: egyet a vásárlói oldalnak, korlátozott láthatósággal, egyet pedig az eladói központnak, teljes láthatósággal.

A vásárlói oldalon megjelenő terméklista (illetve terméklapok) a weblist nevű nézetből veszik az adataikat, melyet a következő SQL parancsok kiadásával hozhatunk létre:

```
CREATE OR REPLACE VIEW products.weblist AS
((SELECT
 productlist.id AS id,
 productlist.title AS title,
 productlist.keywords AS keywords,
 productlist.status AS status,
 ROUND(productlist.price)::INTEGER AS price,
 ROUND (productlist.discount price) :: INTEGER AS discount price,
  warranties.title AS warranty,
 manufacturers.title AS manufacturer,
 ROUND(productlist.weight)::INTEGER AS weight,
 productlist.category AS category,
 productlist.disable shipping AS disable shipping,
 productlist.photo_url_1 AS photo_url_1,
productlist.photo_url_2 AS photo_url_2,
 productlist.photo_url_3 AS photo_url_3,
 productlist.photo url 4 AS photo url
 productlist.seller category AS seller category,
 productlist.video_link AS video_link,
productlist.isbn13 AS isbn13,
 productlist.gtin AS gtin,
 productlist.product code AS product code,
 productlist.warehouse code AS warehouse code,
 productlist.page_link AS page_link,
 ROUND (COUNT FREE AVAILABILITY (productlist.availability 283,
   productlist.reserved))::INTEGER AS availability 283,
 ROUND (productlist.shipping_fee_4_1)::INTEGER AS shipping_fee_4_1, productlist.description AS description,
 productlist.product_group AS product_group,
 productlist.article_group AS article_group
FROM
 products.productlist,
 products.manufacturers,
 products.warranties
WHERE
 (productlist.manufacturer = manufacturers.id AND
 productlist.warranty = warranties.id) AND
  (productlist.status>=0 AND productlist.status<=1) AND
  (COUNT FREE AVAILABILITY (productlist.availability 283, productlist.reserved) >0 OR
 COUNT FREE AVAILABILITY (productlist.availability 283, productlist.reserved)=-1))
UNION
(SELECT
 virtualproducts.id AS id,
 virtualproducts.title AS title,
 virtualproducts.keywords AS keywords,
 virtualproducts.status AS status,
 ROUND(GETVIRTUALPRICE(virtualproducts.id))::INTEGER AS price,
 ROUND (GETVIRTUALDISCOUNTPRICE (virtualproducts.id))::INTEGER AS discount price,
```

```
warranties.title AS warranty,
  GETVIRTUALMANUFACTURER(virtualproducts.id)::CHARACTER VARYING AS manufacturer,
  ROUND (GETVIRTUALWEIGHT (virtualproducts.id))::INTEGER AS weight,
  virtualproducts.category AS category,
  virtualproducts.disable_shipping AS disable_shipping,
 virtualproducts.photo_url_1 AS photo_url_1,
virtualproducts.photo_url_2 AS photo_url_2,
 virtualproducts.photo_url_3 AS photo_url_3,
virtualproducts.photo_url_4 AS photo_url_4,
 virtualproducts.seller_category AS seller_category,
virtualproducts.video_link AS video_link,
  virtualproducts.isbn1\overline{3} AS isbn13,
  virtualproducts.gtin AS gtin,
 virtualproducts.product_code AS product code,
  virtualproducts.warehouse_code AS warehouse_code,
  virtualproducts.page link AS page link,
  -1::INTEGER AS availability 283,
  ROUND (virtualproducts.shipping fee 4 1)::INTEGER AS shipping fee 4 1,
  TRIM(GETVIRTUALDESCRIPTION(virtualproducts.id)||'<P>'||
virtualproducts.description || '</P>') :: TEXT AS description,
 virtualproducts.product group AS product group,
  virtualproducts.article group AS article group
FROM
 products.virtualproducts,
 products.warranties
WHERE
 virtualproducts.warranty = warranties.id AND
 virtualproducts.status>=0 AND virtualproducts.status<=1));</pre>
ALTER TABLE products.weblist OWNER TO "admin";
GRANT ALL ON TABLE products.weblist TO "admin";
GRANT SELECT ON TABLE products.weblist TO public;
COMMENT ON VIEW products.weblist IS 'List of all available products';
```

Az eladói központban megjelenő terméklista (illetve terméklapok) a webadminlist nevű nézetből veszik az adataikat, melyet pedig a következő SQL parancsok kiadásával hozhatunk létre:

```
CREATE OR REPLACE VIEW products.webadminlist AS
((SELECT
  productlist.id AS id,
  productlist.title AS title,
  productlist.keywords AS keywords,
  status informations.title AS status,
  ROUND (productlist.price) :: INTEGER AS price,
  ROUND (productlist.discount price) :: INTEGER AS discount price,
  warranties.title AS warranty,
  manufacturers.title AS manufacturer,
  ROUND (productlist.weight) :: INTEGER AS weight,
  productlist.category AS category,
  productlist.disable_shipping AS disable_shipping,
  productlist.photo_url_1 AS photo_url_1,
productlist.photo_url_2 AS photo_url_2,
productlist.photo_url_3 AS photo_url_3,
  productlist.photo_url_4 AS photo_url_4,
  productlist.seller_category AS seller_category,
  productlist.video link AS video link,
  productlist.isbn13 AS isbn13,
  productlist.gtin AS gtin,
  productlist.product code AS product code,
  productlist.warehouse_code AS warehouse_code,
  productlist.page link AS page link,
  ROUND (productlist.availability_283)::INTEGER AS availability_283,
ROUND (productlist.shipping_fee_4_1)::INTEGER AS shipping_fee_4_1,
  productlist.description AS description,
  productlist.product group AS product group,
  productlist.article group AS article group
```

```
FROM
  products.productlist,
  products.manufacturers,
  products.warranties,
  products.status_informations
WHERE
  productlist.manufacturer = manufacturers.id AND
  productlist.warranty = warranties.id AND
  productlist.status = status informations.id)
UNTON
(SELECT
  virtualproducts.id AS id,
  virtualproducts.title AS title,
  virtualproducts.keywords AS keywords,
  status_informations.title AS status,
  ROUND (GETVIRTUALPRICE (virtualproducts.id))::INTEGER AS price,
  ROUND (GETVIRTUALDISCOUNTPRICE (virtualproducts.id))::INTEGER AS discount price,
  warranties.title AS warranty,
  GETVIRTUALMANUFACTURER(virtualproducts.id)::CHARACTER VARYING AS manufacturer,
  ROUND(GETVIRTUALWEIGHT(virtualproducts.id))::INTEGER AS weight,
  virtualproducts.category AS category,
  virtualproducts.disable_shipping AS disable shipping,
  virtualproducts.photo_url_1 AS photo_url_1,
virtualproducts.photo_url_2 AS photo_url_2,
  virtualproducts.photo_url_3 AS photo_url_3,
  virtualproducts.photo url 4 AS photo url 4,
  virtualproducts.seller category AS seller category,
  virtualproducts.video_link AS video_link,
  virtualproducts.isbn13 AS isbn13,
  virtualproducts.gtin AS gtin,
  virtualproducts.product code AS product code,
  virtualproducts.warehouse_code AS warehouse_code,
  virtualproducts.page link AS page link,
  -1::INTEGER AS availability 283,
  ROUND (virtualproducts.shipping fee 4 1)::INTEGER AS shipping fee 4 1,
  TRIM (GETVIRTUALDESCRIPTION (virtualproducts.id) || '<P>' ||
virtualproducts.description || '</P>')::TEXT AS description,
  virtualproducts.product_group AS product_group,
  virtualproducts.article group AS article group
FROM
  products.virtualproducts,
  products.warranties,
  products.status informations
WHERE
  virtualproducts.warranty = warranties.id AND
  virtualproducts.status = status_informations.id));
ALTER TABLE products.webadminlist OWNER TO "admin";
GRANT ALL ON TABLE products.webadminlist TO "admin";
GRANT SELECT ON TABLE products.webadminlist TO public;
COMMENT ON VIEW products.webadminlist IS
  'United list of all available products in database';
```

Mindkét nézet lekérdezi a terméklistában és a szettek adattáblájában található adatokat, ahol idegen kulcsokat használunk, ott a hivatkozott kód helyett a hivatkozott rekordban található megnevezést adjuk vissza, majd a szetteket tartalmazó adattáblát az adatbázisban található felhasználói függvények segítségével a terméklistának megfelelő formátumban adjuk vissza, egyesítve azzal.

A különbség az, hogy míg a webadminlist nézet az összes terméket és

szettet visszaadja, addig a weblist csak azokat a tételeket adja vissza, amelyekből van szabad készlet vagy rendelhetők.

Attól függetlenül, hogy az eladói központban még szerkesztésre is nézetből vesszük az adatokat, a termékek frissítésénél és új termék felöltésénél ez egyáltalán nem jelent problémát, mert ezekhez a műveletekhez is felhasználói függvényt használunk, ami pontosan tudja, hogy hol kell módosítani az adatokat, a felület tehát nem közvetlenül ír bele az adattáblákba, hanem csak meghív egy-egy függvényt, mely new product () (normál termékeknél), illetve new virtual product() (szetteknél) névre hallgat. Ezek a függvények ellenőrzik, hogy a paraméterben adott tétel létezik-e már az adatbázisban, és ha nem, akkor beszúrják új termékként (vagy szettként), ha pedig igen, akkor frissítik a meglévő termék (vagy szett) adatait. Ha az adatforrásból (vagy más nevén: rendelési segédlistából) szeretnénk egy-egy termékre gyors feltöltést kezdeményezni, akkor new product from datasource() ezt а függvénnyel intézi nekünk az adatbázis, ahol összesen négy adatot kell megadnunk paraméterként (lásd: felhasználói dokumentáció).

3.2.1.2. Bevásárlókosár és online rendelések kezelése

Az adatbázis második legfontosabb adattábla- és nézetcsoportja a megrendelések kezeléséhez szükséges adattáblák és nézetek csoportja, mely az orders és products sémában található. A Grando webáruházzal megőrizendő kompatibilitás érdekében elkerülhetetlen a redundancia a megrendelési adatok tárolásánál.

Szintén ebben az adattábla- és nézetcsoportban találhatjuk a különféle forgalmi kimutatások elkészítéséhez szükséges nézeteket. Ezek a nézetek a megrendelések adattábláját használják, és különféle szűrések, összegzések és csoportosítások szerint jelenítik meg az adatokat, melyeket a felhasználói felületen keresztül exportálhatunk CSV táblázatként (lásd: felhasználói dokumentáció), illetve Java-applet-ek segítéségével megtekinthetünk különféle formátumú diagramként.

Az előbb említett adattábla- és nézetcsoport relációs adatmodelljét a 43-44. ábrákon láthatjuk.



43. ábra. A megrendelésekkel kapcsolatos adatok táblázatai és nézetei (1)

Bevásárlókosár és online rendelések kezelése

number_of_orders	Count(source_table.order_jd)
orderlist	
	orders.order_monthly_income_detailed
year_and_month	T0_CHAR(source_table.date_ordered::TimeStamp With Time Zone, 'YYYY'. month'::Text)::Character Varying
order_year	date_part('year\::Text, source_table.date_ordered)::Integer
order_month	date_part('month::Text, source_table.date_ordered):Integer
number of orders	Count(source_table.income)
orderlist	
orderstatus	
	orders enders not special detailed
waar and month	TO CHAP(source table date and end times the Web may a the Web and the address that the source table date and the web the Web and the Web a
order year	To_CHAR(source_table.date_protects.thmestamp with time zone, TTTT, month.trext)Character varying date part/typer/Text source table date ordered."Integer
order_month	date_part('month'::Text, source_table.date_ordered)::Integer
income	Sum(source_table.income)
number_ot_orders	Count(source_table.order_id)
orderlist	
orderstatus	
-	reduce order weblict
and an id the	orders.order_webist
customer name er	Income. moty to null(htrim((null to empty(orders orderlist family, name:Text) II ""Text) II null to empty(orders orderlist given name:Text)))=Character Varying
email V/	ARCHAR2 (255 CHAR)
phone V/	ARCHAR2 (20 CHAR)
address bt	rrm((((orders.orderlist.shipping_zip ' '::Text) orders.orderlist.shipping_city::Text) ', '::Text) orders.orderlist.shipping_address::Text)::Character Varying
status V/	ARCHAR2 (255 CHAR)
date ordered da	tate trunc('seconds':Text, orders.orderlist.date_ordered)
orderlist	
orderstatus	
paymethods	
	orders.order detailed weblist
order id	orders.order_detailed_weblist
order_id customer_name	orders.order_detailed_weblist INTEGER Intrin((null_to_empty(orders.orderlist.family_name::Text) ''':Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying
order_id customer_name email	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR)
order_id customer_name email phone	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist_family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (25 CHAR)
order_id customer_name email phone shipping_name shipping_zip	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (20 CHAR) btrim((null_to_empty(orders.orderlist.shipping_family_name::Text) ' '::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying INTEGER
order_id customer_name email phone shipping_name shipping_zip shipping_city	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) 1 ''::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (20 CHAR) btrim((null_to_empty(orders.orderlist.shipping_family_name::Text) ''::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying INTEGER VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_city shipping_city shipping_address	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (20 CHAR) btrim((null_ce_empty(orders.orderlist.shipping_given_name::Text))::Character Varying INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_city shipping_address billing_name billing_name	orders.order_detailed_weblist INTEGER UNTEGER UVARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) UNTEGER VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_city shipping_city shipping_city shipping_city billing_rame billing_rip billing_tity	INTEGER UNCLAR2 (255 CHAR) VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_zip shipping_city shipping_city shipping_address billing_name billing_city billing_city	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_zip shipping_city shipping_city shipping_address billing_rame billing_city billing_city billing_city billing_address info	orders.order_detailed_weblist INTEGER UNTEGER UNTEGER VARCHAR2 (255 CHAR) UNTEGER VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cty shipping_cty shipping_cty shipping_cty billing_rip billing_rip billing_cty billing_address info shipping_type any method	Orders.order_detailed_weblist INTEGER UARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
order_id customer_name email shipping_name shipping_city shipping_city shipping_city billing_rity billing_city billing_city billing_type pay_method status	Orders.order_detailed_weblist INTEGER UNTEGER UNTEGER UNTEGER VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_ddress billing_rip billing_rip billing_rip billing_cip billing_cip billing_typ shipping_type shipping_type status flag	orders.order_detailed_weblist INTEGER INTEGER INTEGER Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) Introduction VARCHAR2 (255 CHAR) Int
order_id customer_name email phone shipping_name shipping_cty shipping_cty shipping_address billing_rame billing_ray billing_cty billing_address info shipping_type pay_method status flag date_ordered	orders.order_detailed_weblist INTEGER INTEGER btrim((null_t_c_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) NTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) <
order_id customer_name email phone shipping_name shipping_cip shipping_cip shipping_ddress billing_address billing_cip billing_cip billing_cip billing_cip billing_cip shipping_type pay_method status flag order_tipin	orders.order_detailed_weblist INTEGER INTEGER VARCHAR2 (255 CHAR) VARCHAR2
order_id customer_name email phone shipping_name shipping_cip shipping_ddress billing_rip billing_rip billing_city billing	orders.order_detailed_weblist INTEGER INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CHA
order_id customer_name email phone shipping_name shipping_zip shipping_zip biling_rame biling_rame biling_rip biling_dtress info shipping_type pay_method status date_ordered product_id product_id product_itel product_weight	orders.order_detailed_weblist INTEGER INTEGER btrim((null_t_c_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CH
order_id customer_name email phone shipping_name shipping_cip shipping_dtress billing_address billing_address info billing_city billing_city billing_ddress info address flag product_tide product_tide product_tide product_tide product_warranty product_warranty	orders.order_detailed_weblist INTEGER INTEGER btrim((null_tc_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (225 CHAR) VARCHAR2 (255 CHAR
order_id customer_name email phone shipping_name shipping_cip shipping_dity shipping_dity shipping_dity billing_cip billing_rip billing_ci	orders.order_detailed_weblist INTEGER INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CHA
order_id customer_name email phone shipping_name shipping_zip shipping_zip biling_rame biling_zip biling_dtress biling_dtress biling_dtress biling_dtress biling_address info shipping_type pay_method status flag date_ordered product_id product_id product_title product_price product_group article group	orders.order_detailed_weblist INTEGER UNTEGER VARCHAR2 (255 CHAR) VARCHARZ (255 CHAR)
order_id customer_name email phone shipping_name shipping_zip shipping_dtress billing_address billing_address billing_address info billing_address info date_ordered product_id product_title product_weight product_weight product_warranty product_auantity product_auantity product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup product_aroup	orders.order_detailed_weblist INTEGER UNTEGER UVARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_different billing_cip billing_c	orders.order_detailed_weblist INTEGER INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CHA
order_id customer_name email phone shipping_name shipping_zip shipping_tcty shipping_tcty shipping_tcty billing_try billing_try billing_address info shipping_type pay_method status flag date_ordered product_id product_tid product_tid product_tid product_tid product_trice product_group takeover_park bickpack_sbg_code pickpack_sbg_code	orders.order_detailed_weblist INTEGER INTEGER btrim((null_to_empty/orders.orderlist.family_name::Text) ' '::Text) null_to_empty/orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (25 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_dty shipping_dty shipping_dty billing_address billing_ty billing_ddress billing_ddress billing_ddress billing_ddress info shipping_type pay_method status flag date_ordered product_id product_id product_id product_weight product_weight product_group article_group article_group atkeover_park pickpack_stop_code pickpack_dty_code	orders.order_detailed_weblist INTEGER INTEGER btrim((null_tc_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR
order_id customer_name email phone shipping_name shipping_izip shipping_dtress billing_tip billing_tip billing_tip billing_dtress info shipping_type adjusted shipping_type adjusted shipping_type pay_method status flag date_ordered product_id product_id product_id product_invie product_group atklee_ordered product_group takeover_park takeover_park takeover_park takeover_park takeover_park takeover_park	orders.order_detailed_weblist INTEGER INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (25 CHAR)
order_id customer_name email shipping_name shipping_tip shipping_tip shipping_tip shipping_tip shipping_tip billing_tip billing_tip billing_address info shipping_type pay_method status flag date_ordered product_id product_id product_title product_title product_price product_price product_group takeover_park pickpack_shop_code pickpack_caddress pay_mpaid	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (25 CHAR) <
order_id customer_name email phone shipping_name shipping_cip shipping_dty shipping_dty shipping_dty billing_address billing_address billing_typ pay_method status flag date_ordered product_id product_id product_id product_id product_id product_id product_warranty product_group article_group article_group atkeover_park pickpack_shop_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code pickpack_cip_code	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_ddress billing_tip billing_tip billing_ddress info shipping_type billing_ddress info shipping_type pay_method status flag date_ordered product_id product_id product_id product_id product_id product_group atricle_group takeover_park takeover_park takeover_park park_code pickpack_shop_code pickpack_shop_code pickpack_ci	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ' '::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_ddress billing_address billing_address info billing_address info pay_method status flag product_type product_tidle product_tidle product_weight product_price product_group article_group pictpack_address pay_u_paid order!ist shippingtype paymethods	orders.order_detailed_weblist INTEGER btrim(inull_to_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (225 CHAR) btrim(inull_to_empty(orders.orderlist.shipping_family_name::Text) ''::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) btrim(inull_to_empty(orders.orderlist.billing_given_name::Text) ''::Text) null_to_empty(orders.orderlist.billing_given_name::Text))::Character Varying INTEGER VARCHAR2 (255 CHAR)
order_id customer_name email phone shipping_name shipping_cip shipping_cip shipping_ddress billing_rame billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip billing_cip pay_method status flag date_ordered product_tid product_tid product_tid product_weight product_weight product_weight product_group article_group article_group atticke_group	orders.order_detailed_weblist INTEGER btrim((null_to_empty(orders.orderlist.family_name::Text) ''::Text) null_to_empty(orders.orderlist.shipping_given_name::Text))::Character Varying VARCHAR2 (255 CHAR)

44. ábra. A megrendelésekkel kapcsolatos adatok táblázatai és nézetei (2)

Mint korábban említettük, egy vagy több termék megrendeléséhez a megrendelni kívánt termékeket a bevásárlókosárba kell helyezni. A bevásárlókosár adattáblája tartalmaz egy elsődleges azonosítót, az adott bevásárlókosár létrehozásának dátumát, a kliens gép IP címét (Ipv4 vagy Ipv6), a megrendelni kívánt termékek cikkszámainak tömbjét és a hozzájuk tartozó mennyiségeket, esetleg még néhány megjegyzést a bevásárlókosárhoz fűzve (ami a weboldalon nem jelenik meg). Az adattáblában tárolt bevásárlókosarakat az őket létrehozó kliens gépekhez rendeljük, vagyis elvileg az IP cím is elég

lenne elsődleges azonosítónak, de számokhoz jobban szeretünk kulcsokat rendelni, mint szöveghez. A weboldal azonban végig az IP-címre hivatkozik.

A bevásárlókosár szigorúan csak IP címhez kötött, tehát, ha például egy felhasználó notebook (vagy egyéb mobil eszköz) segítségével létrehoz egy bevásárlókosarat a weboldalon keresztül, de nem rendeli meg a benne található termékeket, majd kilép a hálózatról, ezt követően pedig egy másik felhasználó csatlakozik a rendszerhez, és a hálózat véletlenül pont az előző felhasználó IPcímét osztja ki neki, akkor az új felhasználó számára láthatóvá válnak az előző felhasználó által hátrahagyott tételek a bevásárlókosárban. Ezt mi nem tekintjük problémának, ugyanis – mint korábban említettük – a bevásárlókosarat csak egy ideiglenes tárolónak terveztük, amelynek csak addig kell tárolnia a benne található elemeket, amíg a felhasználó befejezi a termékek közötti keresgélést és megrendeli a bevásárlókosárba helyezett elemeket. Bizonyos idő elteltével az adatbázis automatikusan törölheti a bevásárlókosarat, annak teljes tartalmával együtt azért, hogy felszabadítsa az IP-címet a következő (új) vásárló számára.

A bevásárlókosár adattáblájának neve carts, és az orders sémához tartozik. Az adattábla felépítését a következő táblázatban láthatjuk:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	A bevásárlókosár azonosítója. Értéke nullánál nagyobb egész szám lehet. Megadása kötelező. Elsődleges kulcs.
date_created	TIMESTAMP WITHOUT TIME ZONE	A bevásárlókosár létrehozásának dátuma. Érvényes rendszeridő lehet. Megadása kötelező.
ip_address	CHARACTER VARYING(40)	A bevásárlókosár létrehozását kezdeményező kliens gép IP címe. Lehet Ipv4 vagy Ipv6 érték, szövegként tároljuk. Megadása kötelező.
products	INTEGER[]	A megrendelni kívánt termékek cikkszámait tartalmazó tömb. Megadása kötelező.

Név	PostgreSQL adattípus	Leírás
quantities	DOUBLE PRECISION[]	A megrendelni kívánt termékek mennyiségei. Nullánál nagyobb számok lehetnek. Megadásuk kötelező. A tömb darabszámának meg kell egyeznie a cikkszámokat tartalmazó tömb darabszámával.
comments	TEXT	Megjegyzések. Bármi jöhet ide, nem kerül megjelenítésre a felületen.

4. táblázat. A bevásárlókosár adattáblája

A bevásárlókosár esetében is ugyanaz a helyzet, mint a virtuális termékek (szettek) táblázatánál: hasznos lenne, ha idegen kulcsokkal tudnánk ellenőrizni, hogy a cikkszámok érvényesek-e, azaz valóban léteznek-e a megrendelni kívánt termékek a terméklistában, vagy a szettek listájában, de az eltérő adattípusok miatt erre sajnos nincs lehetőségünk, így a bevásárlókosarat kezelő felhasználói függvényeknek kell elvégezniük ezt az ellenőrzést. Az egyes termékek cikkszámaihoz rendelt mennyiségeknek itt is meg kell egyezniük a cikkszámok számával és itt is számít a sorrend. Természetesen közvetlenül a felületről nem módosíthatjuk a bevásárlókosár adattábláját sem.

A megrendeléseket tartalmazó adattábla az orders séma orderlist adattáblája. Ez az adattábla tartalmazza a megrendelők összes adatát és a megrendelni kívánt termékek legfontosabb adatait (beleértve az árucsoportot és cikkcsoportot a forgalmi statisztikák készítéséhez). A megrendeléseket tartalmazó adattábla nem áll összeköttetésben a bevásárlókosár adattáblájával az adatbázisban, a bevásárlókosár tartalmát a webes felület lekérdezi, majd megrendeléskor a felület biztosítja a megrendelés rögzítéséhez szükséges összes adatot az adatbázis számára, majd a megrendelés leadása után ugyanígy a felület kérvényezi az adatbázistól az így fölöslegessé vált bevásárlókosár törlését. A tényleges adatmódosításokat természetesen az adatbázis végzi el.

A megrendeléseket tartalmazó részletes adattábla felépítését az alábbi
táblázatban foglaltuk össze.

Név	PostgreSQL adattipus	Leírás
transaction_id	INTEGER	Tétel azonosítója. Elsődleges kulcs. Megadása kötelező.
order_id	INTEGER	Megrendelés azonosítója. Több tételes rendelés esetén nem egyedi érték. Megadása kötelező.
family_name	CHARACTER VARYING(255)	A megrendelő vezetékneve. Ennek megadása kötelező.
given_name	CHARACTER VARYING(255)	Megrendelő keresztneve. Rövid cégnév esetén nem feltétlenül van kitöltve, így elhagyható.
email	CHARACTER VARYING(255)	Megrendelő email-címe. Megadása kötelező.
phone	CHARACTER VARYING(20)	Megrendelő telefonszáma
shipping_family_name	CHARACTER VARYING(255)	Átvevő vezetékneve
shipping_given_name	CHARACTER VARYING(255)	Átvevő keresztneve
shipping_zip	INTEGER	Szállítási irányítószám
shipping_city	CHARACTER VARYING(255)	Szállítási település
shipping_address	CHARACTER VARYING(255)	Szállítási cím
billing_family_name	CHARACTER VARYING(255)	Szerződő fél vezetékneve
billing_given_name	CHARACTER VARYING(255)	Szerződő fél keresztneve
billing_zip	INTEGER	Számlázási irányítószám
billing_city	CHARACTER VARYING(255)	Számlázási település
billing_address	CHARACTER VARYING(255)	Számlázási cím
info	TEXT	Megjegyzések
shipping_type	INTEGER	Átvétel típusa. Kétféle lehet: • 4: futárszolgálat; • 8: személyes átvétel. Megadása kötelező. Idegen kulcs, a shippingtype adattábla id mezőjére hivatkozik.
pay_method	INTEGER	Fizetés módja. Kétféle lehet: • 1: Előreutalás; • 4: Készpénz. Megadása kötelező. Idegen kulcs, a paymethods adattábla id mezőjére hivatkozik.

Név	PostgreSQL adattipus	Leírás
status	INTEGER	Megrendelés állapota. A következő értékeket veheti fel: • 0: Feldolgozás alatt (alapértelmezett); • 1: Feladva; • 3: Nem teljesíthető; • 4: Nem utalt; • 5: Téves rendelés; • 6: Átvéve; • 8: Nem vette át; • 9: Teljesítettnek tekintett; • 10: Teljesítve. Megadása kötelező. Idegen kulcs, az orderstatus adattábla id mezőjére hivatkozik.
flag	INTEGER	 Kétféle érték fordul elő: 0: Nincs megadva számlázási cím; 1: Meg van adva számlázási cím. Megadása kötelező, de nem lényeges az értéke. Még nem teljesen tisztázott a szerepe és a lehetséges értéktartománya, ezért a rendszer jelenleg nem is használja, nem épül rá semmi.
date_ordered	TIMESTAMP WITHOUT TIME ZONE	Megrendelés dátuma és ideje. Érvényes rendszeridőnek kell lennie. Megadása kötelező.
product_id	INTEGER	Megrendelt termék cikkszáma. Megadása kötelező. Idegen kulcs egyelőre nincs hozzá definiálva, mert egyszere két adattábla id mezőjét is felveheti értéknek (productlist és virtualproducts).
product_title	CHARACTER VARYING(255)	Megrendelt termék megnevezése. Megadása kötelező.
product_warranty	INTEGER	Megrendelt termékra adott garancia hónapokban. Megadása kötelező. Alapértelmezett értéke 6.
product_weight	DOUBLE PRECISION	Megrendelt termék súlya
product_price	DOUBLE PRECISION	Megrendelt termék eladási ára. Megadása kötelező.
product_quantity	DOUBLE PRECISION	Megrendelt termék mennyisége. Megadása kötelező. Értéke alapértelmezés szerint 1.
takeover_park	CHARACTER VARYING(255)	Átvétel helye (település)

Egy lakberendező cég információs rendszere

Név	PostgreSQL adattipus	Leírás
pickpack_shop_code	CHARACTER VARYING(127)	Pickpack-pont-átvevőhely kódja (még nem használjuk)
pickpack_zip_code	CHARACTER VARYING(127)	Pickpack-pont-átvevőhely irányítószám-kódja (még nem használjuk)
pickpack_city	CHARACTER VARYING(255)	Pickpackpont-átvevőhely (település; még nem használjuk)
pickpack_address	CHARACTER VARYING(255)	Pickpackpont-átvevőhely címe (még nem használjuk)
pay_u_paid	CHARACTER VARYING(127)	Fizetéssel kapcsolatos információ. Szerepe nem tisztázott, így még nem is használjuk.
takeover_park_id	INTEGER	Átvétel helye. Megadása nem kötelező. Idegen kulcs, a products.houses adattábla id mezőjére hivatkozik.
product_group	INTEGER	A megrendelt termék árucsoportja. A rendszer automatikusan tölti ki megrendeléskor és importáláskor, ha lehetséges.
article_group	INTEGER	A megrendelt termék cikkcsoportja. A rendszer automatikusan tölti ki megrendeléskor és importáláskor, ha lehetséges.

5. táblázat. Megrendelések adattáblája

A szállítás/átvétel módját tartalmazó adattábla az orders séma shippingtype adattáblája. Ennek felépítését foglalja össze a következő táblázat:

Név	PostgreSQL adattípus	Leírás
id	INTEGER	Szállítási/átvételi mód azonosítója. Megadása kötelező. Elsődleges kulcs.
title	CHARACTER VARYING(255)	A szállítási/átvételi mód megnevezése. Megadása kötelező.
comments	TEXT	Bármilyen megjegyzés a szállítási/átvételi móddal kapcsolatban. Nem jelenik meg az oldalon.
delivery	BOOLEAN	Az adott szállítási/átvételi mód jelent-e kiszállítást? Megadása kötelező, a felület ez alapján tudja eldönteni, hogy milyen űrlapot jelenítsen meg a rendeléseknél.

6. táblázat. A szállítási módokat tartalmazó adattábla

A lehetséges fizetési módokat tartalmazó adattábla az orders sémához tartozó paymethods tábla, melynek felépítését az alábbi táblázatban láthatjuk:

Név	PostgreSQL adattípus	Leírás
id	INTEGER	Fizetési mód azonosítója. Megadása kötelező. Elsődleges kulcs.
title	CHARACTER VARYING(255)	Fizetési mód megnevezése. Megadása kötelező.
comments	TEXT	Megjegyzések az adott fizetési módhoz kapcsolódóan. A weboldalon nem jelenik meg.

^{7.} táblázat. A lehetséges fizetési módokat tartalmazó adattábla

A megrendelések állapotát és sikerességét az orders séma orderstatus adattáblája írja le. Az adattábla felépítése a következő:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Állapot azonosítója. Elsődleges kulcs, megadása kötelező.
title	CHARACTER VARYING(255)	Állapot megnevezése. Megadása kötelező.
comments	TEXT	Megjegyzések az adott állapottal kapcsolatban. A felületen nem jelenik meg.
success	BOOLEAN	Sikeres értékesítés esetén értéke TRUE (igaz), sikertelen értékesítés esetén FALSE (hamis). Megadása kötelező.
canceled	BOOLEAN	Sikeres értékesítés esetén értéke FALSE, sikertelen értékesítés esetén TRUE. Megadása kötelező.

8. táblázat. A megrendelések állapotait leíró adattábla

Előfordulhat olyan állapot, amely se nem sikeres, se nem sikertelen (például: feldolgozás alatt), de olyan nem, amelyik egyszerre sikeres és sikertelen. Ezt a következő megszorítással ellenőrizhetjük:

```
CONSTRAINT check_status_success CHECK((success=TRUE AND canceled=TRUE)=FALSE)
```

A telephelyek adatait a products séma houses adattáblája tartalmazza. Bár ez a tábla a termékek sémájához van bejegyezve, megrendelések rögzítésekor az átvétel helyét is tároljuk, amit innen olvasunk ki. Az adattábla

Név	PostgreSQL adattípus	Leírás
id	INTEGER	Telephely azonosítója. Elsődleges kulcs, megadása kötelező.
title	CHARACTER VARYING(255)	Telephely megnevezése. Megadása kötelező.
zip	INTEGER	Telephely irányítószáma. Megadása kötelező.
city	CHARACTER VARYING(255)	Telephely székhelye. Megadása kötelező.
address	CHARACTER VARYING(255)	Telephely címe. Megadása kötelező.
description	TEXT	Telephely leírása. Megadása kötelező.
comments	TEXT	Megjegyzések az adott telephellyel kapcsolatban.
email	CHARACTER VARYING(255)	Telephely email-címe. Megadása kötelező.
phone	CHARACTER VARYING(20)	Telephely telefonszáma

felépítését az alábbi táblázatban láthatjuk.

9. táblázat. Telephelyek adattáblája

A telephelyek adattáblájának tartalma a felületen is megjelenik, a legtöbb adat nyilvánosan látható a vásárlói oldalon az elérhetőségeknél.

A megrendelések adattáblájából nem közvetlenül olvassuk ki az adatokat a felületen, mert nem szeretnénk kódokat megjeleníteni az eladók számára, hanem beszédes, érthető és áttekinthető megrendelőlistát szeretnénk megjeleníteni az eladói központban. Ennek érdekében két nézetet is létrehoztunk az orders sémában.

Az order_weblist nevű nézet egy egyszerűsített listát jelenít meg az eladói központ "Rendelések kezelése" menüjében, amelyben csak a legfontosabb adatok láthatók. Ezt a nézetet a következő SQL utasítások generálják:

```
CREATE OR REPLACE VIEW orders.order_weblist AS

SELECT DISTINCT

orderlist.order_id,

EMPTY_TO_NULL(TRIM(NULL_TO_EMPTY(orderlist.family_name)||' '||

NULL_TO_EMPTY(orderlist.given_name)))::CHARACTER VARYING AS customer_name,

orderlist.email,

orderlist.phone,

TRIM(orderlist.shipping_zip||' '||orderlist.shipping_city||', '||

orderlist.shipping_address)::CHARACTER VARYING AS address,

orderstatus.title AS status,

paymethods.title AS pay_method,

DATE_TRUNC('seconds', orderlist.date_ordered) AS date_ordered

FROM orders.orderlist, orders.orderstatus, orders.paymethods
```

Bevásárlókosár és online rendelések kezelése

```
WHERE orderlist.status=orderstatus.id AND orderlist.pay_method=paymethods.id;
ALTER TABLE orders.order_weblist OWNER TO "admin";
COMMENT ON VIEW orders.order weblist IS 'List of all orders in web-friendly format';
```

Az order_detailed_weblist nézet a teljes megrendelési listát megjeleníti, de az adattáblához képest az összes adatot áttekinthető, olvasható formátumban. Ennek a nézetnek azonban csak egy-egy rekordját láthatjuk egyszerre, mivel egy megrendelés kiválasztása után jelenik csak meg a weboldalon. Ezt a nézetet a következő SQL utasítások generálják:

```
CREATE OR REPLACE VIEW orders.order detailed weblist AS
(SELECT
 orderlist.order id AS order_id,
 TRIM(NULL_TO_EMPTY(orderlist.family_name)||' '||
   NULL_TO_EMPTY(orderlist.given_name))::CHARACTER VARYING AS customer_name,
 orderlist.email AS email,
 orderlist.phone AS phone,
 TRIM(NULL_TO_EMPTY(orderlist.shipping_family_name)||' '||
   NULL TO EMPTY (orderlist.shipping given name))::CHARACTER VARYING AS shipping name,
 orderlist.shipping_zip AS shipping_zip,
 orderlist.shipping_city AS shipping_city,
orderlist.shipping_address AS shipping_address,
 TRIM(NULL_TO_EMPTY(orderlist.billing_family_name)||' '||
   NULL TO EMPTY(orderlist.billing given name))::CHARACTER VARYING AS billing name,
 orderlist.billing zip AS billing zip,
 orderlist.billing_city AS billing_city,
 orderlist.billing_address AS billing_address,
 orderlist.info AS info,
 shippingtype.title AS shipping type,
 paymethods.title AS pay method,
 orderlist.status AS status,
 orderlist.flag AS flag,
 DATE_TRUNC('seconds', orderlist.date_ordered) AS date_ordered,
 orderlist.product_id AS product_id,
orderlist.product_title AS product_title,
 warranties.title AS product_warranty,
 orderlist.product weight AS product weight,
 orderlist.product price AS product price,
 orderlist.product_quantity AS product_quantity,
 orderlist.product_group AS product_group,
 orderlist.article_group AS article_group,
 orderlist.takeover park AS takeover park,
 orderlist.pickpack_shop_code AS pickpack_shop_code,
 orderlist.pickpack_zip_code AS pickpack_zip_code,
 orderlist.pickpack_city AS pickpack_city,
 orderlist.pickpack_address AS pickpack_address,
 orderlist.pay u paid AS pay u paid,
 ORDER CLOSED (order id) AS order closed
FROM
 orders.orderlist,
 orders.paymethods,
 orders.shippingtype,
 products.warranties
WHERE
 orderlist.pay_method = paymethods.id AND
 orderlist.shipping type = shippingtype.id AND
 orderlist.product_warranty = warranties.id);
ALTER TABLE orders.order detailed weblist OWNER TO "admin";
COMMENT ON VIEW orders.order detailed weblist IS
  'Show order list in browser-friendly format. Status by code only (for HTML forms).';
```

Ennél a nézetnél a megrendelés állapotát kóddal jelenítjük meg, mivel erre a

kódra szükség lesz a webes felületen az esetlegesen ide tartozó űrlapelem megfelelő elkészítéséhez.

A sémához tartozó többi nézet lényegében a forgalmi adatok statisztikáihoz szükségesek. Valamennyi havi bontásban mutatja a bevételeket, folyamatban lévő megrendeléseket és sztornókat. Szerepüket az alábbiak szerint foglalhatjuk össze:

- incoming_orders: havi bontásban megjeleníti a beérkező megrendeléseket (sikeres, törölt és feldolgozás alatti);
- incoming_orders_detailed: ugyanaz, mint az előző, de részletesebben mutatja az adatokat;
- order_monthly_income: havi forgalmi kimutatás nagy vonalakban;
- order_monthly_income_detailed: havi forgalmi kimutatás részletesebben kifejtve;
- orders_not_canceled: sikeres és feldolgozandó megrendelések havi bontású forgalmi kimutatása nagy vonalakban;
- orders_not_canceled_detailed: az előbbi kimutatás tovább részletezett változata.

Kivételt képez a grando_orderlist nézet, mely lehetővé teszi a megrendelések letöltését a Grando webáruházzal kompatibilis formátumban.

Bár nem kifejezetten a megrendelésekhez tartozik, de azért valamilyen szinten mégis kapcsolódik hozzájuk a fórum adattáblája, melyet az orders sémában hoztunk létre forum néven. Azért került ide, mert – egyrészt – nincs külön séma a fórumhoz, új sémát pedig legalább két tábla esetén szoktunk létrehozni, másrészt pedig a potenciális vásárlók vélhetően előbb érdeklődnek a fórumon az egyes termékekkel kapcsolatban, ha pedig megkapták a választ a

kérdéseikre, akkor megrendelik az adott termékeket, így tartalmilag lényegében a megrendelésekhez kapcsolódik.

Név	POSTGRESQL ADATTIPUS	Leírás
id	INTEGER	Hozzászólás azonosítója. Elsődleges kulcs, megadása kötelező.
product_id	INTEGER	A hozzászólás által érintett termék cikkszáma. Megadása kötelező.
family_name	CHARACTER VARYING(255)	A hozzászóló vezetékneve vagy rövid cégnév. Megadása kötelező.
given_name	CHARACTER VARYING(255)	A hozzászóló keresztneve vagy cég típusa
email	CHARACTER VARYING(255)	A hozzászóló email-címe. Megadása kötelező.
post	TEXT	A hozzászólás tartalma. Megadása kötelező, legalább 1 értelmes karaktert tartalmaznia kell.
date_posted	TIMESTAMP WITHOUT TIME ZONE	A hozzászólás dátuma. Megadása kötelező, érvényes rendszeridő lehet.
answer_to	INTEGER	Melyik hozzászólásra válasz az aktuális hozzászólás? Ha üres, akkor önálló hozzászólás, egyébként pedig az érintett hozzászólás azonosítója lehet.
comments	TEXT	Belső láthatóságú megjegyzések az adott hozzászólást illetően. A felületen nem jelenik meg, csak az adatbázis rendszergazdái láthatják.

A fórum adattáblájának felépítését a következő táblázatban tekinthetjük meg:

10. táblázat. A fórum adattáblájának felépítése

A webes felületen a konkrét hozzászólást a felhasználónak csak tartalmilag kell kitöltenie, minden más értéket a hozzászólás létrehozásához szükséges felhasználói függvény generál.

3.2.1.3. Pénzügyi és adminisztrációs adatok kezelése

Pénzügyi és adminisztrációs adatoknak a könyvelést, illetve az eladói központban regisztrált felhasználók adatait tekintjük. Ez két adattáblát jelent a

financial sémában: az accountancy és az admins adattáblát. Előbbi a könyvelést, utóbbi a regisztrált felhasználók adatait tartalmazza. Két, egymástól teljesen független adattábláról beszélünk, de tartalmilag ezek is összetartoznak valamilyen szinten, így ugyanabban a sémában lettek bejegyezve.

A webes felületen keresztül az accountancy tábla utólag már nem módosítható, kizárólag beszúrni lehet rekordokat a NEW_TRANSACTION() felhasználói függvény segítségével, az admins adattáblában módosíthatjuk az adatokat, törölhetünk belőle, de azt is csak a felület által ellenőrzött keretek között. Új felhasználó hozzáadása a NEW_ADMIN() felhasználói függvény segítségével történik.

Sikeres értékesítés esetén a SET_ORDER_STATUS () felhasználói függvény – amivel a felületen keresztül a megrendelés állapotát beállítjuk – automatikusan meghívja a NEW_TRANSACTION () felhasználói függvényt, felparaméterezve azt az adott megrendelésből vett adatokkal, így a könyvelésben automatikusan bejegyzés keletkezik minden megrendelés sikeres státuszra történő állításakor.



Az ide tartozó relációs adatmodellt a 45. ábrán láthatjuk:

45. ábra. Pénzügyi és adminisztrációs adatok relációs adatmodellje

az accountancy adattábla szerkezetét az alábbi táblázat írja le.

Név	POSTGRESQL ADATTIPUS	Leírás
id	SERIAL	A tranzakciós bejegyzés azonosítószáma. Elsődleges kulcs, megadása kötelező. Az accountancy_id_seq szekvencia generálja az értékeit.
title	CHARACTER VARYING(255)	A tranzakció megnevezése, jogcíme. Megadása kötelező.
description	TEXT	Tranzakció szöveges leírása. Megadása kötelező.
ammount	DOUBLE PRECISION	A tranzakció tárgyát képező pénzösszeg. Megadása kötelező.
transaction_date	TIMESTAMP WITHOUT TIME ZONE	Tranzakció dátuma. Megadása kötelező.
comments	TEXT	A tranzakcióhoz fűzött megjegyzések.

11. táblázat. A könyvelést tartalmazó adattábla

Mint korábban említettük, új tranzakciós bejegyzést a NEW_TRANSACTION() felhasználói függvény segítségével hozhatunk létre. Itt meg kell adni a tranzakció megnevezését vagy jogcímét, leírását, a tranzakció tárgyösszegét, illetve megjegyzést is fűzhetünk a bejegyzéshez. Az összes többi mező értékét a függvény (illetve szekvencia) automatikusan generálja, éppen ezért nem is kell megadni őket. A tranzakció dátuma a függvény meghívásának dátuma és ideje lesz.

A regisztrált felhasználókat tartalmazó admins adattábla felépítését a következő táblázatban láthatjuk:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Felhasználó azonosítószáma. Nullánál nagyobb egész szám lehet. Elsődleges kulcs, megadása kötelező, de a felület nem használja.
family_name	CHARACTER VARYING(255)	Felhasználó vezetékneve. Megadása kötelező.
given_name	CHARACTER VARYING(255)	Felhasználó keresztneve. Megadása kötelező.
email	CHARACTER VARYING(255)	Felhasználó email-címe. Megadása kötelező.

Név	PostgreSQL adattipus	Leírás
login_name	CHARACTER VARYING(255)	Felhasználó bejelentkezési neve. Megadása kötelező. Egyedi értékeknek kell lenniük!
password	CHARACTER VARYING(255)	Felhasználó jelszava. Megadása kötelező.
phone	CHARACTER VARYING(20)	Felhasználó telefonszáma
zip	INTEGER	Felhasználó lakóhelyének irányítószáma
city	CHARACTER VARYING(255)	Felhasználó lakóhelye (település)
address	CHARACTER VARYING(255)	Felhasználó lakcíme
birthplace	CHARACTER VARYING(255)	Születési hely
birthdate	DATE	Születési dátum
elementary_school	CHARACTER VARYING(255)	Általános iskola
highschool	CHARACTER VARYING(255)	Középiskola
university	CHARACTER VARYING(255)	Főiskola/egyetem
previous_employments	TEXT	Előző munkahelyek
comments	TEXT	Megjegyzések
superuser	BOOLEAN	Az adott felhasználó rendszergazda-e? Megadása kötelező, alapértelmezett értéke: TRUE.

12. táblázat. A regisztrált felhasználókat tartalmazó adattábla

Szerveroldalon, közvetlenül az adatbázisba történő bejelentkezés esetén a regisztrált felhasználók adattáblájának semmilyen jelentősége nincs, mert nyilván a telepítéskor létrehozott admin felhasználónévvel jelentkezünk be, ami az adatbázis tulajdonosa, és amit felruháztunk az összes létező jogosultsággal, tehát az összes adatbáziselemet el kell tudnunk érni, és bármilyen függvényt meghívhatunk az adatbázisban.

A korábbi alapelvekhez hasonlóan a könyvelés adattáblájában tárolt adatokat sem közvetlenül jelenítjük meg, hanem erre is létrehozunk egy nézetet, melyet accountancy_weblist-ként nevezünk el, ami lényegében ugyanazt adja vissza, mint ami az adattáblában van tárolva. Erre azért van szükség, hogy az adattáblában tárolt dátum értéket másodpercre kerekítve jeleníthessük meg (az adatbázisszerver ugyanis a másodperceket is tizedestörtszámként kezeli, ami helyenként igen csúnya megjelenést eredményezne a webes felületen).

A könyvelés megjelenítéséhez szükséges nézetet a következő SQL parancssorozat generálja:

```
CREATE OR REPLACE VIEW financial.accountancy_weblist AS
SELECT
    accountancy.id,
    accountancy.title,
    accountancy.description,
    accountancy.description,
    accountancy.ammount,
    DATE_TRUNC('seconds', accountancy.transaction_date) AS transaction_date,
    accountancy.comments
    FROM financial.accountancy;
ALTER TABLE financial.accountancy_weblist OWNER TO "admin";
COMMENT ON VIEW financial.accountancy_weblist IS
    'Show accountancy in browser-friendly format';
```

Ezenkívül szükségünk van egy másik nézetre is, ami kiszámolja az aktuális egyenleget és elküldi a webes felületnek megjelenítésre. Ezt a nézetet accountancy_total_ammount-nak nevezzük el, és a következő, az eddigiek tükrében nem túl bonyolult SQL parancssorozattal hozhatjuk létre:

```
CREATE OR REPLACE VIEW financial.accountancy_total_ammount AS
SELECT SUM(accountancy.ammount) AS total_ammount
FROM financial.accountancy;
ALTER TABLE financial.accountancy_total_ammount OWNER TO "admin";
COMMENT ON VIEW financial.accountancy_total_ammount IS
'Show total ammount of money in table accountancy';
```

Csak megemlítés szintjén, tekintsük a könyvelés adattáblájában használt szekvenciát, melyet a következő SQL parancsok generálnak:

```
CREATE SEQUENCE financial.accountancy_id_seq
INCREMENT 1
MINVALUE 0
MAXVALUE 9223372036854775807
START 1
CACHE 1;
ALTER TABLE financial.accountancy id seq OWNER TO "admin";
```

Bár a kódból nem látszódik, van a szekvenciának egy nem elhanyagolható adattagja: called. A fenti kód esetén ennek nincs nagy jelentősége, de amikor valamiért közvetlenül be kell lépnünk az adatbázisba és törölnünk kell egy vagy több rekordot a könyvelés adattáblájából (esetleg az összeset), akkor szükségünk lehet ennek az adattagnak a módosítására. Ez az adattag azt határozza meg, hogy egy szekvencia értékének visszaadása és a tárolt érték

megnövelése milyen sorrendben történjen: előbb adjuk vissza az aktuális értéket, majd utána növeljük azt meg az INCREMENT adattagnál megadott értékkel, vagy éppen fordítva, előbb növeljük meg, és a megnövelt értéket adjuk-e vissza. A rendszerben már tárolt szekvenciánál ez az adattag TRUE-ra van állítva, és minimum értéknek nullát adtunk meg. Vagyis, ha töröljük a könyvelésből az összes bejegyzést, akkor a szekvencia értékét utána nullára kell állítanunk. Ezután új bejegyzés generálásakor a szekvencia jó azonosítót fog visszaadni az adattáblában.

Sok esetben szekvenciát is használhattunk volna az azonosítók megadásánál, de amikor az adattáblákat hoztuk létre, mérlegeltük, hogy a jövőben vélhetően hányszor kell majd közvetlenül belepiszkálni az adattáblákba és rekordokat törölni belőle. Ennek megfelelően arra jutottunk, hogy csak azoknak az adattábláknak az azonosítóit fogjuk szekvenciákkal generálni, amelyekből – a rendszer rendeltetésszerű használata esetén – nagy valószínűséggel soha nem kell majd adatot törölnünk (így nem kell túl gyakran két helyen is ellenőrizni és módosítani ugyanazokat az azonosítókat). A többi adattáblában felhasználói függvényekkel generáljuk az azonosítókat, amennyiben az adattábla azonosítóját nem kívánjuk a webes felületen keresztül módosíthatóvá tenni.

3.2.1.4. Nyilvános adatok és felhasználói függvények

A következőkben azokat az adattáblákat tárgyaljuk, melyek elvben nem tartalmaznak olyan adatot, ami a láthatóság korlátozását igényli, ezt követően pedig rátérünk az adatbázisban használt felhasználói függvények tárgyalására.

A nyilvános adatokat tároló adattáblák a public sémában találhatók. Független adattáblákról van szó, mindegyikük más és más szerepet tölthet be. Egy részük a tesztadatok generálásához lett felhasználva, más részük a helyesírás-ellenőrzéskor kerül felhasználásra, illetve van olyan is, amelyiknek a jövőbeli fejlesztéseknél is haszna lesz, amikor például a weboldalon a rendszer egy megadott irányítószámhoz megpróbálja hozzárendelni a települést.

A public séma relációs adatmodelljét a 46. ábrán láthatjuk.

public.zipcodes	public.familynames	public.givennames
P * id INTEGER * city VARCHAR2 (255 CHAR) * zip INTEGER * state VARCHAR2 (255 CHAR)	P * id INTEGER * title VARCHAR2 (255 CHAR) * country VARCHAR2 (255 CHAR) comments CLOB	P * id INTEGER * title VARCHAR2 (255 CHAR) * country VARCHAR2 (255 CHAR) comments CLOB
⇒ zipcode_id (id) ↓ zipcode_id_unique (id)	→ familyname_id (id)	givenname_id (id) givenname_id_uniq (id)
public.keywordlist	public.test_table	
P * id INTEGER * word_entry VARCHAR2 (255 CHAR) > word_id (id) > word_id_unique (id)	Kitle VARCHAR2 (255 CHAR) Kitle VARCHAR2 (255 CHAR) Kouble_value NUMBER boolean_value CHAR (1) comments CLOB	
	🖙 test_table_PK (id)	

46. ábra. A nyilvános adatok relációs adatmodellje

Elsőként tekintsük a gyakran használt vezeték- és keresztnevek adattábláját. Ezek a familynames és givennames névre hallgatnak. Ezeket az adattáblákat véletlenszerű megrendelési adatok generálásához használtuk fel. Felépítésüket az alábbi táblázatban találjuk:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Azonosítószám. Elsődleges kulcs. Megadása kötelező. "Tradicionális" szerepe van, soha nem használjuk.
title	CHARACTER VARYING(255)	Vezeték- vagy keresztnév. Megadása kötelező.
country	CHARACTER VARYING(255)	Melyik országban használják leggyakrabban? Megadása kötelező.
comments	TEXT	Megjegyzések

13. táblázat. A familynames és givennames adattáblák felépítése

Következő adattáblánk a kulcsszavak listája, amely a keywordlist nevet viseli. Helyesírás-ellenőrzéshez készítettük, többségében nyelvtanilag korrekt keresőszavakat tartalmaz. A helyesírás-ellenőrzés a SPELLCHECK() felhasználói függvény meghívásával történik meg. A függvény az első paraméterként adott, feltehetően hibás keresőszóhoz megkeresi a hozzá legjobban illeszkedő korrekt keresőszavat a keywordlist táblázatban a

második paraméterként adott tűréshatáron belül. Ha a megadott tűréshatáron belül nem talál egyezést, akkor nincs javítás.

A keresőszavakat tartalmazó adattábla felépítését az alábbi táblázatban foglaltuk össze:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Azonosítószám. Elsődleges kulcs, megadása kötelező.
word_entry	CHARACTER VARYING(255)	A konkrét keresőszó. Megadása kötelező.

14. táblázat. A javított keresőszavakat tartalmazó adattábla felépítése

Véletlenszerű megrendelés-adatok generálásához is felhasználtuk azt az adattáblát, amely települések nevét tartalmazza a hozzájuk tartozó postai irányítószámokkal együtt. Ennek az adattáblának a neve zipcodes. A jövőben lehet nagyobb jelentősége a megrendelések összeállításánál, ugyanis megkönnyítheti a megrendelők számára az űrlapok kitöltését. Mivel azonban postai irányítószám és a hozzá tartozó település sok esetben nem rendelhető egymáshoz egyértelműen, a megrendelések űrlapjának kitöltésekor semmit nem helyettesít, csak kényelmesebbé teheti az oldal használatát. Az adattábla felépítése az alábbi táblázatban látható:

Név	PostgreSQL adattípus	Leírás
id	INTEGER	Bejegyzés azonosítószáma. Elsődleges kulcs, megadása kötelező.
city	CHARACTER VARYING(255)	Település neve. Megadása kötelező.
zip	INTEGER	Postai irányítószám. Megadása kötelező.
state	CHARACTER VARYING(255)	Ország/állam/megye megnevezése
comments	TEXT	Megjegyzések

15. táblázat. Postai irányítószámok és települések adattáblája

Táblázatok importálásának és exportálásának teszteléséhez készítettük a test_table nevű adattáblát. Különféle adattípusok import/export formátumát ellenőrizzük vele, hogy a webes felület megfelelően tudja kezelni az élesben használt adatok konverzióját. Az adattábla felépítését az alábbi táblázatban

foglaltuk össze részletesen:

Név	PostgreSQL adattipus	Leírás
id	INTEGER	Azonosítószám, megadása kötelező. Egész szám lehet.
title	CHARACTER VARYING(255)	Megnevezés, megadása kötelező. Tetszőleges szöveg lehet.
description	TEXT	Leírás, megadása kötelező. Tetszőleges szöveg lehet.
double_value	DOUBLE PRECISION	Lebegőpontos számtípus teszteléséhez használt mező. Megadása kötelező. Tetszőleges valós szám lehet.
boolean_value	BOOLEAN	Logikai adatok teszteléséhez használt mező. Megadása kötelező. IGAZ/HAMIS (TRUE/FALSE) érték lehet.
comments	TEXT	Megjegyzések. Elhagyható. Értéke tetszőleges szöveg lehet.

16. táblázat. A tesztelő táblázat felépítése

Az adatbázisban található függvények a public sémában kaptak helyet, így meghívásuk gyors és egyszerű. Természetesen az, hogy egy függvény a public sémában található, önmagában még nem garantálja, hogy mindenki számára futtatható lesz, ehhez külön meg kell adni a jogosultságot minden függvénynél.

Az adatházichan	dofiniált függyány	vokat az alábbi	táblázatban	foolaltuk	öcc70.
Az aualbazisbari	uennan nugyveng	yerel az alabbi	labiazalbari	iogiailur	03320.

Függvény	Visszatérési érték típusa	Leírás
<pre>add_to_cart(input_id integer, input_quantity double precision, input_address character varying, validity_timeout integer)</pre>	INTEGER	*Termék kosárba helyezése, érvénytelenítési idő megadásával (milyen régi bevásárlókosarak törölhetők). Nyilvános.
<pre>add_to_cart(input_id integer, input_quantity double precision, input_address character varying)</pre>	INTEGER	*Termék kosárba helyezése, érvénytelenítési idő alapértelmezett. Nyilvános.
<pre>cart_product_number(input_address character varying)</pre>	INTEGER	*Hányféle termék található a bevásárlókosárban? Nyilvános.
<pre>count_free_availability(availability double precision, reserved double precision)</pre>	DOUBLE PRECISION	*Szabad készlet kiszámítása a raktári- és lefoglalt készlet megadásával. Nyilvános.
<pre>delete_cart(input_address character varying)</pre>	INTEGER	*Bevásárlókosár törlése. Nyilvános.

Függvény	Visszatérési érték típusa	Leírás
discountisvalid(discountid integer)	BOOLEAN	*Érvényes-e még a paraméterként megadott központi akció?
<pre>empty_to_null(inputstring text)</pre>	TEXT	*Üres szöveget NULL értékre konvertál. Nyilvános.
<pre>export_grando_productlist(filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Terméklista exportálása a Grando webáruház formátumában, megadva minden szükséges paramétert.
<pre>export_grando_productlist(filename character varying, table_delimiter character varying)</pre>	INTEGER	*Terméklista exportálása a Grando webáruház formátumában, csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_incoming_orders(filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Bejövő megrendelések statisztikai adatainak exportálása az összes szükséges paramétert megadva.
<pre>export_incoming_orders(filename character varying, table_delimiter character varying)</pre>	INTEGER	*Bejövő megrendelések statisztikai adatainak exportálása csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_monthly_income(filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása az összes szükséges paramétert megadva.
<pre>export_monthly_income(filename character varying, table_delimiter character varying)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_monthly_income_by_product_gr oups(order_successful boolean, order_canceled boolean, filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása árucsoportonkénti bontásban, az összes szükséges paramétert megadva.

Függvény	Visszatérési érték típusa	Leírás
<pre>export_monthly_income_by_product_gr oups(order_successful boolean, order_canceled boolean, filename character varying, table_delimiter character varying)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása árucsoportonkénti bontásban, csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_monthly_income_for_product_g roups(input_product_groups integer[], filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása a megadott árucsoportokra összegezve, az összes szükséges paramétert megadva.
<pre>export_monthly_income_for_product_g roups(input_product_groups integer[], filename character varying, table_delimiter character varying)</pre>	INTEGER	*Havi forgalmi statisztika adatainak exportálása a megadott árucsoportokra összegezve, csak a szükséges paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_not_canceled_orders(filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Sikeres és folyamatban lévő megrendelések havi forgalmi adatainak exportálása az összes szükséges paramétert megadva.
<pre>export_not_canceled_orders(filename character varying, table_delimiter character varying)</pre>	INTEGER	*Sikeres és folyamatban lévő megrendelések havi forgalmi adatainak exportálása, csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_orderlist(filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Megrendelések exportálása a Grando webáruházzal kompatibilis formátumban, az összes szükséges paramétert megadva.
<pre>export_orderlist(filename character varying, table_delimiter character varying)</pre>	INTEGER	*Megrendelések exportálása a Grando webáruházzal kompatibilis formátumban, csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>export_table(tablename character varying, filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Táblázat exportálása az összes szükséges paramétert megadva. Gyakorlatilag az összes export- függvény közvetve vagy közvetlenül ezt a függvényt hívja meg.

Függvény	Visszatérési érték típusa	Leírás
<pre>export_table(tablename character varying, filename character varying, table_delimiter character varying)</pre>	INTEGER	*Táblázat exportálása csak az alapvető paramétereket megadva, a többihez alapértelmezett értékeket használunk.
<pre>get_city_by_zip(zipcode_in integer)</pre>	TEXT	Visszaadja a paraméterként adott irányítószámhoz tartozó települést, ha egyértelmű. NULL-t ad vissza egyébként. Nyilvános.
<pre>get_default_weight(input_id integer)</pre>	DOUBLE PRECISION	*Lekérdezi az alapértelmezett súlyt a products.default_weights táblázatból a megadott termékhez.
<pre>get_free_availability(input_id integer)</pre>	DOUBLE PRECISION	Szabad készlet lekérdezése a paraméterként megadott termékhez. Nyilvános, de nem használjuk, mert sok feldolgozandó adat esetén lassú.
<pre>get_hdt_category_by_load_data(input_size double precision, input_weight double precision)</pre>	INTEGER	*Lekérdezi a megadott körmérethez és súlyhoz tartozó HDT-s szállítási árkategóriát.
get_hdt_category_by_load_data(input_weight double precision)	INTEGER	*Lekérdezi a megadott súlyhoz tartozó HDT-s szállítási árkategóriát. A hiányzó körmérethez alapértelmezés szerint veszi a legkisebb bejegyzett körméretet.
<pre>get_manufacturer_by_id(manufacturerid integer)</pre>	TEXT	Visszaadja a paraméterként adott kódú szállító nevét.
<pre>get_manufacturerid_by_name(manufacturername text)</pre>	INTEGER	Visszaadja a paraméterként adott nevű szállító kódját.
<pre>get_product_weight(input_id integer)</pre>	DOUBLE PRECISION	*Lekérdezi a megadott termék súlyát a products.weights adattáblából, ha nem találja, akkor az alapértelmezett súlyt adja vissza, ha van, egyébként NULL-t.
<pre>get_random_city(input_zip integer)</pre>	TEXT	A megadott irányítószámhoz tartozó települések közül véletlenszerűen visszaad egyet. Nyilvános.
<pre>get_random_family_name()</pre>	TEXT	Véletlenszerűen visszaad egy gyakran használt vezetéknevet. Nyilvános.
<pre>get_random_given_name()</pre>	TEXT	Véletlenszerűen visszaad egy gyakran használt keresztnevet. Nyilvános.
get_random_zip()	INTEGER	Véletlenszerűen visszaad egy irányítószámot. Nyilvános.

Függvény	Visszatérési érték típusa	Leírás
<pre>get_zip_by_city(cityname text)</pre>	INTEGER	Visszaadja az adott településhez tartozó postai irányítószámot, ha egyértelmű. NULL-t ad vissza egyébként. Nyilvános.
getavailability(inputid integer)	DOUBLE PRECISION	*Lekérdezi az adatforrásból a megadott termék készletét. Összeadja a raktári készletet a kiállításban lévő készlettel, ha abból van legalább 2 darab, raktári készlet egyébként. Ha raktári készlet sincs, akkor ellenőrzi, hogy a termék rendelhető-e. Ha igen, akkor -1 lesz a darabszám, ha nem, akkor 0. Nem a terméklistából dolgozik, hanem külsőleg megkapott és importált adatforrásból (más néven: rendelési segédlistából)!
getcorrectstatus(inputid integer)	INTEGER	 *Visszaadja a paraméterben adott termék korrekt státuszát, azaz: Ha van legalább 1 db fénykép a termékhez és van belőle készlet, vagy rendelhető, akkor 1; Ha vagy nincs fényképe, vagy nincs készlete (és nem is rendelhető), akkor 2.
getcurrentdiscounts(inputid integer)	INTEGER	*Lekérdezi a paraméterben adott termékhez tartozó érvényes központi akciók számát.
<pre>getlonger(inputstring1 text, inputstring2 text)</pre>	TEXT	*A paraméterekben adott szövegek közül visszaadja a hosszabbat. Nyilvános.
getrecentdiscount(inputid integer)	INTEGER	*A paraméterben adott termékhez visszaadja az éppen érvényesnek tekinthető központi akciót, ha van ilyen. NULL-t ad vissza egyébként.
getshippingfee(inputid integer)	DOUBLE PRECISION	*Az áruház által meghatározott alapelvek alapján kiszámítja egy termékhez a szállítási díjat.
getvirtualdescription(inputid integer)	TEXT	*Visszaadja a paraméterben adott virtuális termék (szett) elemeit olvasható, áttekinthető HTML- kódként. Nyilvános.
getvirtualdiscountprice(inputid integer)	DOUBLE PRECISION	*Kiszámítja a paraméterben adott virtuális termék akciós árát. Nyilvános.

Függvény	Visszatérési érték típusa	Leírás
getvirtualmanufacturer(inputid integer)	TEXT	*Meghatározza a paraméterben adott virtuális termék szállítóját a mennyiségi adatok alapján. Nyilvános.
getvirtualprice(inputid integer)	DOUBLE PRECISION	*Kiszámítja a paraméterben adott virtuális termék normál árát (listaárát). Nyilvános.
getvirtualweight(inputid integer)	DOUBLE PRECISION	*Kiszámítja a paraméterben adott virtuális termék súlyát. Nyilvános.
greater(value1 double precision, value2 double precision)	DOUBLE PRECISION	*Két valós szám közül visszaadja a nagyobbat. Ha egyik paraméter NULL, akkor a visszatérési érték a másik szám lesz. Ha mindkettő NULL, akkor a visszatérési érték is NULL. Nyilvános.
handleavailability(warehouse double precision, ordered double precision)	DOUBLE PRECISION	 *Meghatározza a paraméterként adott készletek közül a dominánsat: Ha van az áruházban készlet, akkor ez lesz a visszatérési érték; Ha nincs, de a termék rendelhető, akkor -1; Egyébként 0.
<pre>import_datasource(datasource_filename character varying, outgoing_filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Importálja az első paraméterben adott adatforrás-fájlt és a második paraméterben adott, kifutó termékek listáját a harmadik paraméterben adott mezőelválasztóval tagolva, a negyedik paraméterben megadott idézőjelet figyelembe véve, az ötödik paraméter szerint eldöntve, hogy van-e a táblázatnak fejléce. Röviden: beolvas két darab CSV fájlt és tartalmukat rögzíti az adatbázisban.
<pre>import_datasource(datasource_filename character varying, outgoing_filename character varying, table_delimiter character varying)</pre>	INTEGER	*Ugyanazt csinálja, mint az előbbi függvény, de itt alapértelmezett értékek helyettesítik a hiányzó paramétereket.
<pre>import_orderlist(orderlist_filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	Importálja a Grando webáruházból letöltött megrendelések listáját az összes szükséges paramétert megadva. Már nem ezt használjuk erre a webes felületen.

Függvény	Visszatérési érték típusa	Leírás
<pre>import_orderlist(orderlist_filename character varying, table_delimiter character varying)</pre>	INTEGER	Ugyanazt csinálja, mint az előbbi függvény, csak itt alapértelmezett értékek helyettesítik a hiányzó paramétereket. Már nem ezt használjuk a webes felületen.
<pre>import_table(tablename character varying, filename character varying, table_delimiter character varying, quote_character character varying, has_header boolean)</pre>	INTEGER	*Beolvassa az első paraméterben megadott táblázatba a második paraméterben megadott fájlt a harmadik paraméterben megadott elválasztót használva a negyedik paraméterben megadott idézőjel figyelembe vételével, az ötödik paraméter szerint ellenőrizve, hogy van-e fejléc az importálandó táblázatban. Minden egész táblázatot importáló függvény közvetlenül vagy közvetve ezt a függvényt használja.
<pre>import_table(tablename character varying, filename character varying, table_delimiter character varying)</pre>	INTEGER	*Ugyanazt csinálja, mint az előbbi függvény, de itt a hiányzó paramétereket alapértelmezett értékek helyettesítik.
<pre>insert_into_datasource(new_id text, new_keywords text, new_title text, new_description text, new_ange_code text, new_unit text, new_package_unit text, new_package_unit text, new_product_group text, new_availability_0 text, new_availability_0 text, new_availability_0 text, new_availability_a text, new_availability_al text, new_last_3_months_sale text, new_coverage_group text, new_minimum_availability_text, new_minimum_availability_text, new_lkz_text, new_lkz_text, new_monthly_availability_text, outgoing_sign_text)</pre>	INTEGER	*Beszúr egy új rekordot a rendelési segédlistába az összes, szövegként megadott paramétert átkonvertálva a megfelelő formátumra. Kisméretű rendelési segédlisták importálásához készült, amikor a webes felület gyorsan végig tud menni az importálandó állomány minden során és csak az abban megadott termékek adatait módosítja az adatbázisban, a többi termék adatát nem piszkálja.

Függvény	Visszatérési érték típusa	Leírás
<pre>insert_into_orderlist(new_order_id integer, new_family_name character varying, new_given_name character varying, new_mail character varying, new_shipping_family_name character varying, new_shipping_zip integer, new_shipping_city character varying, new_shipping_address character varying, new_billing_family_name character varying, new_billing_top integer, new_billing_city character varying, new_billing_dtress character varying, new_billing_dtress character varying, new_billing_dtress character varying, new_billing_dtress character varying, new_billing_dtress character varying, new_billing_dtress character varying, new_flag integer, new_flag integer, new_flag integer, new_flag integer, new_flag integer, new_flag integer, new_flag integer, new_product_id integer, new_product_title character varying, new_product_warranty integer, new_product_weight double precision, new_product_quantity double precision, new_product_quantity double precision, new_product_shop_code character varying, new_pickpack_shop_code character varying, new_pickpack_city character varying, new_pickpack_city character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_ity character varying, new_pickpack_ity character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_ity character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_ity character varying, new_pickpack_address character varying, new_pickpack_address character varying, new_pickpack_ity character varying, new_pickpack_ity character varying, new_pickpack_ity character varying, new_pickpack_ity character varying, new_pickpack_ity ch</pre>	INTEGER	*Az előbbi függvényhez hasonlóan beszúr egy új rekordot a megrendelések közé az adatbázisban, ha lehetséges, az adattábla többi adatát nem módosítja. Ha a beszúrás sikertelen volt, hibajelzéssel tér vissza. Megrendelések gyors és biztonságos importálásához készült, ahol fontos, hogy a régi adatok sértetlenek maradjanak. Néhány ezer rekord importálásához ideális, amikor a webes felület gyorsan végig tud menni a bemenetként megadott állományon, a webes felületen tehát ezt a függvényt használjuk.
isnumber(inputstring text)	BOOLEAN	*Ellenőrzi, hogy a megadott szöveg szám-értéket tartalmaz-e. Nyilvános.
lowernumber(value1 double precision, value2 double precision)	DOUBLE PRECISION	*Két valós szám közül visszaadja a kisebbet. Ha egyik paraméter NULL, akkor a visszatérési érték a másik szám lesz. Ha mindkettő NULL, akkor a visszatérési érték is NULL. Nyilvános.

Függvény	Visszatérési érték típusa	Leírás
<pre>mpl_shipping_available(indput_id integer)</pre>	BOOLEAN	*Ellenőrzi, hogy a megadott termék az adatai (tömege) alapján alkalmas- e arra, hogy MPL-es szállítást alkalmazzunk rá.
<pre>mpl_shipping_enabled(indput_id integer)</pre>	TEXT	*Ellenőrzi az előző függvény alapján, hogy a megadott termékre alkalmazható-e MPL-es szállítás, ha igen, akkor NULL-t ad vissza, egyébként "–" karaktert. Jelzés a Grando-kompatibilis terméklistában, hogy egy adott terméknél lehet-e MPL-lel szállíttatni.
<pre>new_admin(new_familyname character varying, new_givenname character varying, new_email character varying, new_loginname character varying, new_password character varying, new_phone character varying, new_zip integer, new_city character varying, new_address character varying, new_birthplace character varying, new_birthplace character varying, new_birthplace character varying, new_birthdate date, new_elementary_school character varying, new_niversity character varying, new_university character varying, new_previous_employments text, new_comments text, new_superuser boolean)</pre>	INTEGER	*Regisztrál egy új felhasználót az eladói központban, az összes személyes adatát megadva.
<pre>new_admin(new_familyname character varying, new_givenname character varying, new_email character varying, new_loginname character varying, new_password character varying, new_superuser boolean)</pre>	INTEGER	*Regisztrál egy új felhasználót az eladói központban, csak az alapadatokat megadva.
<pre>new_discount(inputid integer, discount_price_in double precision, availability_in double precision, start_date_in date, end_date_in date, comments_in text, promotion_in boolean)</pre>	INTEGER	*Új különleges akciót határoz meg egy termékhez, így a terméklista központi frissítésénél ez hatályon kívül helyezi az adatforrásban található árat és ezt az akciót alkalmazza a termékre. Lehetőséget ad promóció létrehozására is, ekkor a megjegyzés nem lehet üres, mert hozzá fogja fűzni azt a Grando-s terméklista megadott termékének megnevezéséhez gyűjtő linkek létrehozása érdekében.
<pre>new_manufacturer(new_id integer, new_title character varying, new_comments text)</pre>	INTEGER	*Új szállítót hoz létre a paraméterekben megadott adatok alapján, vagy módosít egy meglévőt.

Függvény	Visszatérési érték típusa	Leírás
<pre>new_order(new_familyname character varying, new_givenname character varying, new_email character varying, new_phone character varying, new_shipping_familyname character varying, new_shipping_ip integer, new_shipping_ip integer, new_shipping_city character varying, new_shipping_address character varying, new_billing_familyname character varying, new_billing_givenname character varying, new_billing_city character varying, new_billing_city character varying, new_billing_city character varying, new_billing_ddress character varying, new_billing_address character varying, new_billing_type integer, new_shippingtype integer, new_shippingtype integer, new_shippingtype integer, new_productid integer[], new_productuantity double precision[], new_takeoverparkid integer)</pre>	INTEGER	*Új megrendelést rögzít az adatbázisban a paraméterekben megadott felhasználói adatokkal. Nyilvános.
<pre>new_post(input_id integer, familyname_in character varying, givenname_in character varying, email_in character varying, post_in text, answer_to_post integer)</pre>	INTEGER	*Új hozzászólást rögzít a rendszerben a paraméterekben megadott termékhez a paraméterekben megadott adatok alapján. Nyilvános.
<pre>new_product(new_id integer, new_title character varying, new_keywords character varying, new_status integer, new_price double precision, new_discount_price double precision, new_marnaty integer, new_marnaty integer, new_manufacturer integer, new_disable_shipping integer, new_disable_shipping integer, new_bhoto_url_1 character varying, new_photo_url_2 character varying, new_photo_url_4 character varying, new_otdeo_link character varying, new_seller_category character varying, new_sibn13 bigint, new_gtin bigint, new_gtin bigint, new_availability_283 double precision, new_shipping_fee_4_1 double precision, new_atticle_group integer, new_areserved double precision)</pre>	INTEGER	*Új terméket hoz létre a rendszerben, vagy módosít egy meglévőt a paraméterekben megadott adatok alapján.

Függvény	Visszatérési érték típusa	Leírás
<pre>new_product_from_datasource(inputid integer, pricein double precision, discount_price_in double precision, photo_url_in character varying)</pre>	INTEGER	*Új terméket hoz létre a hivatalos adatforrásból, a paraméterekben adott cikkszám, normál ár, akciós ár és fénykép-link felhasználásával.
<pre>new_shop_reply(house_id integer, answer_to_post integer, post_in text)</pre>	INTEGER	*Új hozzászólást rögzít a rendszerben a paraméterekben megadott adatokkal. Eladói oldalon keresztül hívjuk meg, válaszként rögzül a rendszerben.
<pre>new_transaction(new_title character varying, new_description text, new_ammount double precision, new_comments text)</pre>	INTEGER	*Új tranzakciót rögzít a könyvelésben a paraméterekben megadott adatokkal.
<pre>new_virtual_product(new_id integer, new_title character varying, new_keywords character varying, new_status integer, new_discount double precision, new_discount double precision, new_discount double precision, new_discount double precision, new_discount double precision, new_discount double precision, new_arranty integer, new_disable_shipping integer, new_photo_url_1 character varying, new_photo_url_3 character varying, new_photo_url_4 character varying, new_seller_category character varying, new_seller_category character varying, new_isbn13 bigint, new_product_code bigint, new_product_code bigint, new_marehouse_code bigint, new_arehouse_code bigint, new_description text, new_product_group integer, new_article_group integer, new_article_group integer, new_comments text, new_product_quantities double precision[])</pre>	INTEGER	*Új virtuális terméket (szettet) rögzít a rendszerben vagy módosít egy meglévőt a paraméterekben megadott adatok alapján.
<pre>null_to_empty(inputstring text)</pre>	TEXT	*NULL értéket üres szöveggé konvertál a helyes összefűzéshez. Nyilvános.
order_closed(inputid integer)	BOOLEAN	*Lekérdezi, hogy a paraméterként adott megrendelés lezárt állapotban van-e vagy még módosítható.
<pre>product_group_income(input_product_group integer, input_status integer[], start_date date, end_date date)</pre>	DOUBLE PRECISION	*Lekérdezi a megadott árucsoportra vonatkozó forgalmat a megadott értékesítési státuszokra szűrve a megadott időintervallumon belül.
<pre>product_group_income(input_product_group integer, order_success boolean, order_canceled boolean, start_date date, end_date date)</pre>	DOUBLE PRECISION	*Ugyanazt csinálja, mint az előbbi függvény, de itt státuszkódok helyett logikai értékekkel megadjuk, hogy sikeres, sikertelen (vagy egyik sem) értékesítésekre nézzük.

Függvény	Visszatérési érték típusa	Leírás
<pre>product_group_percentage(input_product_group integer, input_status integer[], start_date date, end_date date)</pre>	DOUBLE PRECISION	Az előbbiekhez hasonlóan a forgalmi arányt kérdezi a megadott árucsoportra vonatkozóan. A szűrés státuszkód alapján történik.
<pre>product_group_percentage(input_product_group integer, order_success boolean, order_canceled boolean, start_date date, end_date date)</pre>	DOUBLE PRECISION	Ugyanaz, mint az előbbi, csak itt logikai értékekkel szűrjük az értékesítési státuszokat.
<pre>product_groups_income(input_product_groups integer[], input_status integer[], start_date date, end_date date)</pre>	DOUBLE PRECISION	*Adott árucsoportokra összegzi a forgalmat a megadott státuszkódok szerinti szűréssel a megadott időintervallumon belül.
<pre>product_groups_income(input_product_groups integer[], order_success boolean, order_canceled boolean, start_date date, end_date date)</pre>	DOUBLE PRECISION	*Ugyanaz, mint az előbbi, csak itt logikai értékekkel adjuk meg az értékesítések állapotát.
<pre>product_groups_orders(input_product_groups integer[], input_status integer[], start_date date, end_date date)</pre>	INTEGER	*Az előbbiekhez hasonló, de itt a rendelési darabszámot számolja ki a rendszer a megadott státuszkódok alapján.
<pre>product_groups_orders(input_product_groups integer[], order_success boolean, order_canceled boolean, start_date date, end_date date)</pre>	INTEGER	*Ugyanaz, mint az előbb, de itt a logikai értékek számítanak az értékesítési állapotok szűrésénél.
<pre>remove_from_cart(input_id integer, input_address character varying)</pre>	INTEGER	*Törli a második paraméterben adott IP-címhez rendelt kosárból az első paraméterben megadott terméket. Nyilvános.
<pre>set_order_status(inputid integer, status_in integer)</pre>	INTEGER	*Beállítja az első paraméterben adott megrendelés állapotát a második paraméterben megadott állapotra, sikeres állapot esetén pedig lekönyveli a megrendelésből származó bevételt.
<pre>spellcheck(inputstring text, treshold integer)</pre>	TEXT	*Megpróbálja kijavítani az első paraméterben megadott szöveget a keresőszavakat tartalmazó adattábla alapján, a második paraméterben megadott hibahatáron belül. Ha nem sikerül a javítás, akkor nincs változtatás. Nyilvános.

Függvény	Visszatérési érték típusa	Leírás
<pre>string_distance(inputstring1 text, inputstring2 text)</pre>	INTEGER	*Megszámolja a paraméterekben megadott szövegek közötti eltérő karakterek számát a gyors algoritmus segítségével. Ha elcsúszás van a karakterek között, akkor nem biztos, hogy jó eredményt ad, de egy-egy rosszul kódolt ékezetes karakter kijavítására általában ez is megfelel. Nyilvános.
<pre>string_distance(inputstring1 text, inputstring2 text, algorythm integer)</pre>	INTEGER	 *Megszámolja a paraméterekben megadott szövegek közötti eltérő karakterek számát. Az algoritmus választható: 0: Levenshtein-algoritmus (nem használjuk, mert sok adat feldolgozása esetén iszonyatosan lassú); Egyébként: gyors algoritmus (egy fokkal jobb). A Levenshtein-algoritmust kizárólag 1-2 rekord összehasonlítására használhatjuk, a gyors algoritmus akár több százra is működik. Nyilvános.
<pre>strtodouble(inputstring text)</pre>	DOUBLE PRECISION	*A "000.000.000,00" formátumú szövegeket konvertálja át valós számokká, akkor is, ha "*" karakterrel kezdődik. Nyilvános.
update_all_products()	INTEGER[]	*Frissíti az összes terméket a hivatalos adatforrásól és tömbként visszaadja azokat a problémás termékeket, amelyeknek árait nem lehetett frissíteni. Az ilyen termékek automatikusan -1 (problémás, letiltott) státuszt kapnak.

17. táblázat. Felhasználói függvények

Azokat a függvényeket, melyek leírását a "*" karakterrel megjelöltük, ténylegesen használja az adatbázis, illetve a webes felhasználói felület, míg a többi függvény a rendszer továbbfejlesztett verzióiban fog szerepet játszani, ezeknek a függvényeknek a segítségével kiköveztük az utat a további, új szolgáltatásokat is bevezető fejlesztésekhez.

Még egy sor további, egyik kategóriába sem igazán tartozó adattábla és nézet is része az adatbázisnak, a fontosabbakat a következőkben részletezzük,

a kevésbé fontosakat (melyeket jelenleg a rendszer ténylegesen még nem használ) csak megemlítés szintjén tárgyaljuk.

3.2.1.5. Egyéb elemek az adatbázisban

Az említett további és eddig nem tárgyalt elemek nagy része egymástól teljesen független adattáblákból és nézetekből áll, ám néhányuk között valódi relációs kapcsolat is van. Az ide tartozó relációs adatmodellt a 47-48. ábrákon láthatjuk.



47. ábra. Egyéb adatbáziselemek relációs adatmodellje (1)

> popl chipping fa	oc RK (id)	categor 🔅	yid (category)	
mp_snipping_re	use foo (min weight may weight	categor	ryid_unique (category)	
cneck_mpi_snipp	oing_ree (min_weight, max_weight,	recipient_point)		
	vaducts default weights		products.upda	te_products
	oroduces.deradic_weights		id INTEG	SER
PF* product_group	INTEGER		title VARC	HAR2 (255 CHAR)
P * article_group	INTEGER		keywords VARC	HAR2 (255 CHAR)
P * min_price	NUMBER		status INTEG	SER
max_price	NUMBER		price NUME	REP
* weight	NUMBER		discount price NUM	REP
comments	CLOB		availability 283 NUME	BER
🛥 default weights	PK (product group article group	min price)	shipping fee 4 1 NUM	BER
acroant_rreignes	_, it (produce_group; citicite_group;	(find)	product group INTEG	SER
4		11	article group INTEG	SED
			reserved NUM	DED
products	alabal text variables	products.weights	reserved home	5EIX
produces	giobal_cext_valiables	P * id INTEGER	productlist	
P 10	INTEGER	* weight NUMBER		
U title	VARCHAR2 (255 CHAR)	* colli INTEGER		
* value	CLOB	comments CLOR		
comments	CLOB	commence CLOB	orders.gra	ndo_orderlist
蹄 globaltextid (id)		🖙 weights_PK (id)	order id	INTEGER
今 globaltextdatan	ame_unique (title)		family name	VARCHAR2 (255 CHAR)
			diven name	VARCHAR2 (255 CHAR)
			email	VARCHAR2 (255 CHAR)
_	and a state of the state of the state of the		phone	VARCHAR2 (20 CHAR)
(products.grando_prod	ucclisc	shipping family name	VARCHAR2 (255 CHAR)
title	btrim(((((((products.global_text_	variables.value ' - '::Text) pro	odu shipping riven name	VARCHAR2 (255 CHAR)
seller_product_id	INTEGER		shipping_given_name	INTEGER
status	INTEGER		shipping_zip	VARCHAR2 (255 CHAR)
price	ROUND(products.productlist.price	e)::Integer	subbuildCork	VARCHARZ (200 CHAR)
		· · · · · · · · · · · · · · · · · · ·	chipping addrose	VADOUADO (OSS OUAD)
discount_price	ROUND(products.productlist.disco	punt_price)::Integer	shipping_address	VARCHAR2 (255 CHAR)
discount_price warranty	ROUND(products.productlist.disco INTEGER	ount_price)::Integer	shipping_address billing_family_name billing_given_name	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR)	ount_price)::Integer	shipping_address billing_family_name billing_given_name	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig	punt_price)::Integer Jht)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_zip	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER
discount_price warranty manufacturer weight category	ROUND (products. productlist. disco INTEGER VARCHAR2 (255 CHAR) ROUND (products. productlist. weig INTEGER	ount_price)::Integer ht)::Integer	shipping_address billing_family_name billing_given_name billing_gip billing_city billing_city	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER	ount_price)::Integer ht)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_city billing_address infe	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_city billing_address info	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER VARCHAR2 VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_city billing_address info shipping_type	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER VARCHAR2 VARCHAR2 VARCHAR2	ount_price)::Integer Int)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_city billing_address info shipping_type pay_method	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_zip_ billing_city billing_address info shipping_type pay_method status	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_city billing_city billing_address info shipping_type pay_method status flag	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_cty billing_address info shipping_type pay_method status flag date_ordered	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER	ount_price)::Integer	shipping_address billing_family_name billing_gipen_name billing_cip billing_city billing_address info shipping_type pay_method status flag date_ordered product_id	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER TIMESTAMP INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_city billing_city billing_address info shipping_type pay_method status flag date_ordered product_id product_title	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code	ROUND (products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND (products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_cty billing_address info shipping_type pay_method status flag date_ordered product_id product_title product_title	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn.13 gtin product_code warehouse_code	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_cty billing_address info shipping_type pay_method status flag date_ordered product_id product_title product_title	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2	ount_price)::Integer	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city billing_address info shipping_type pay_method status flag date_ordered product_id product_title product_waranty product_weight product_price	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER NUMBER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER	ount_price)::Integer ht)::Integer ability_283)::Integer	shipping_address billing_family_name billing_given_name billing_cip billing_city billing_address info shipping_type pay_method status flag date_ordered product_id product_itle product_warranty product_wight product_price product_quantity	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER NUMBER
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_4 1	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 ROUND(products.productlist.stwia ROUND(products.productlist.stwia	ount_price)::Integer ht)::Integer ability_283)::Integer ping fee 4 1)::Integer	shipping_address billing_family_name billing_given_name billing_zip billing_cty billing_address info shipping_type pay_method status flag date_ordered product_id product_title product_title product_title product_price product_quantity takeover_park	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER NUMBER VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_64_1	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 ROUND(products.productlist.avai ROUND(products.productlist.avai ROUND(products.productlist.avai	ount_price)::Integer ht)::Integer ability_283)::Integer ring_fee_4_1)::Integer ring_fee_4_1)::Integer	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city billing_address info shipping_type pay_method status flag date_ordered product_id product_tid product_weight product_weight product_quantity takeover_park pickpack_shop_code	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_4_1 shipping_fee_64_1 description	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER IN	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city billing_address info shipping_type pay_method status flag date_ordered product_itle product_itle product_title product_wright product_wright product_price product_price product_price product_park	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_4_1 shipping_fee_64_1 description	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 ROUND(products.productlist.avai ROUND(products.productlist.avai nmL_shipping_enabled(products.pr CLOB	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1)::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_given_name billing_cip billing_cty billing_address info shipping_type pay_method status flag date_ordered product_id product_id product_id product_title product_title product_warranty product_price product_price product_price product_price product_price product_price product_price product_price product_price product_price product_price	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER NUMBER NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_64_1 shipping_fee_64_1 description	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER VARCHAR2 ROUND(products.productlist.avai ROUND(products.productlist.avai ROUND(products.productlist.ship) .mpLshipping_enabled(products.pr CLOB	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1)::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city billing_cy and the set of	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) NUMBER NUMBER NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_64_1 description global_text_variat productlist	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER IN	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1)::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city shipping_type pay_method status flag date_ordered product_id product_id product_title product_title product_wright produc	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) NUMBER NUMBER NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_3 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_4_1 shipping_fee_64_1 description globa_text_variat productlist manufacturers	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER ROUND(products.productlist.avai ROUND(products.productlist.avai npl_shipping_enabled(products.pr CLOB	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1)::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_cip billing_cip billing_city billing_city shipping_type pay_method status flag date_ordered product_id product_id product_id product_warranty product_warranty product_wight product_price price product_price	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)
discount_price warranty manufacturer weight category disable_shipping photo_url_1 photo_url_2 photo_url_4 seller_category video_link isbn13 gtin product_code warehouse_code page_link availability_283 shipping_fee_4_1 shipping_fee_4_1 description global_text_variat productist	ROUND(products.productlist.disco INTEGER VARCHAR2 (255 CHAR) ROUND(products.productlist.weig INTEGER VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 ROUND(products.productlist.avai ROUND(products.productlist.avai ROUND(products.productlist.ship) .mpl_shipping_enabled(products.pr CLOB	ount_price)::Integer ht)::Integer ability_283)::Integer ping_fee_4_1)::Integer roductlist.id)::Character Varying	shipping_address billing_family_name billing_given_name billing_city billing_city billing_city billing_cy pay_method status flag date_ordered product_id product_id product_id product_weight product_weight product_weight product_weight product_guantity takeover_park pickpack_city pickpack_city pickpack_city pickpack_city pickpack_city pickpack_city	VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) INTEGER VARCHAR2 (255 CHAR) CLOB INTEGER INTEGER INTEGER INTEGER INTEGER VARCHAR2 (255 CHAR) INTEGER NUMBER NUMBER NUMBER VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR) VARCHAR2 (255 CHAR)

48. ábra. Egyéb adatbáziselemek relációs adatmodellje (2)

Bár tematikailag az "egyéb" kategóriában kapott helyet, tényleges használata és rendkívül fontos szerepe miatt részletesen kell foglalkoznunk a products sémához tartozó global_text_variables nevű adattáblával. Ez az adattábla globális szöveges változókat tartalmaz a legkülönfélébb célra az adattábla s szerepe, az adattábla konkrét tartalmát is meg kell vizsgálnunk. Ebből az adattáblából szigorúan tilos rekordokat törölni, csak hozzáértő felhasználók módosíthatják, és az újonnan beszúrt rekordok azonosítószámainak szigorúan nagyobbnak kell lennie, mint az adattáblában meglévő azonosítószámok! Erről

a products sémában található global_text_variables_id_seq szekvencia gondoskodik, melyet a következő SQL parancsokkal hozhatunk létre:

```
CREATE SEQUENCE products.global_text_variables_id_seq
INCREMENT 1
MINVALUE 0
MAXVALUE 9223372036854775807
START 1
CACHE 1;
ALTER TABLE products.global text variables id seq OWNER TO "admin";
```

A szekvencia teljesen hasonló az előzőleg bemutatott szekvenciához, a called adattag értéke itt is TRUE. Ennek a szekvenciának az értékeit azonban csak erősen indokolt esetben szabad módosítani!

Név	PostgreSQL adattípus	Leírás
id	SERIAL	Változó azonosítószáma. Elsődleges kulcs, megadása kötelező, de értékét a global_text_variables_id_seq szekvencia generálja. Nem egyenértékű a változó nevével!
title	CHARACTER VARYING(255)	Változónév. Megadása kötelező.
value	TEXT	Változó értéke. Megadása kötelező, és legalább 1 értelmes karaktert tartalmaznia kell!
comments	TEXT	A változóhoz fűzött megjegyzések. A további fejlesztések segítése érdekében általában ki szoktuk tölteni ezt is, de nem kötelező.

Az adattábla felépítését a következő táblázatban találjuk:

18. táblázat. A global_text_variables adattábla felépítése

Az adattáblában található változók értékeinek egy része automatikusan beszúrásra kerül minden új termékhez, amennyiben az adatforrásból hoztuk azokat létre. Más részük a weboldalon történő érthető megjelenítés érdekében szerepelnek az adattáblában.

Az adattábla tartalmát a következő táblázatban foglaltuk össze:

ID	TITLE	VALUE	COMMENTS
1	Rendelési tájékoztató	Kedves online	Minden termék
-		megrendelő!	leírásához hozzáfűzni!
		Jogosult a megrendeléstől	
		a termék átvételét	
		belül indoklás nélkül	
		elállni, ebben az esetben	
		a termék visszaszállitási	
		költségei Önt	
		terhelik.	
		A2 elado elernelosege: 06-1-123-4567 < br >	
		<pre></pre>	
		A megrendeléstől	
		számított egy héten belül	
		kérjük szíveskedjen	
		szemelyesen atvenni a	
		áruházunk	
		vevőszolgálatán!	
		Amennyiben előre	
		utalással szeretné	
		megrendelni a termeket,	
		legven kedves elutalni az	
		összeget, különben	
		rendelése törlésre	
		kerül.	
		 br>	
		<pre>//beger tesuket koszonjuk!</pre>	
		A fenti kép csak	
		illusztráció! 	
		Az ár csak a	
		termékeket tartalmazza.	
2	Cég monogram	Fake	Minden termékhez
			hozzáfűzésre kerül
3	Sikeres értékesítés	Sikeres értékesítés	Minden sikeres eladás
			után bejegyzésre kerül
			a kõnyvelesben
4	Megrendelés azonosítója	Megrendelés-azonosító	Megrendelés
			azonosítószáma
5	Generált bejegyzés	Automatikusan generált	Függvények által
L		bejegyzés	létrehozott rekordok
6	Virtuális szett részei	A szett elemei	Minden virtuális
			szetthez hozzáfűzésre
			kerül
7	Elsődleges azonosítószám	Cikkszám	Minden termékre
			érvényes
8	Valuta rövidítése	HUF	Használt valuta
L			(Forint)
9	Cég neve	Fake Lakberendezési Kft.	Elsősorban a weboldalon
			történő megjelenítéshez
			szükséges
10	Súlymérték	d	Általános súlymérték
			(mindenhol így adjuk
			meg)

19. táblázat. A global_text_variables adattábla tartalma

Az adattábla tartalma nem végleges, újabb és újabb szolgáltatások bevezetésével folyamatosan bővül az itt tárolt speciális szerepű adatok halmaza.

Mint korábban többször utaltunk rá, az adatbázis támogat egy speciális adatforrást, ami hatalmas segítséget jelent akkor, amikor nagyon nagy tételben szeretnénk új termékeket feltölteni rendszerünkbe а NEW PRODUCT FROM DATASOURCE () függvény segítségével. Ez az adattábla a products sémában található és datasource a neve. Ez az adattábla rengeteg termék forrás-adatát tartalmazza. Az adattábla tartalmát pontosvesszővel vagy tabulátorral tagolt CSV-fájlból olvassuk be. Ezt a fájlt külső cégtől kapjuk meg, nem garantálható, hogy mindig megfelel a formai és tartalmi követelményeknek, így létrehozásakor rendkívül megengedőek voltunk: minden mező szövegformátumban van, kivéve a legelsőt (id), mely a cikkszámokat tartalmazza. Az adattáblából függvények segítésével olvassuk ki az adatokat, melyek alaposan leellenőrzik azokat, elvégzik a szükséges konverziókat, a hiányzó adatokat paraméterben kell nekik megadni, ezáltal garantálják, hogy hibás adat ne kerülhessen ebből az adattáblából a terméklistába. Erre az adattáblára a dokumentumban többször rendelési segédlistaként is hivatkoztunk.

Amikor a frissen megkapott rendelési segédlistából frissítjük termékeink adatait az UPDATE_ALL_PRODUCTS() függvény segítségével, szükségünk lehet arra, hogy bizonyos termékek esetében, meghatározott időre hatályon kívül helyezzük a rájuk vonatkozó akciókat és saját magunk mondjuk meg, hogy mekkora kedvezményt akarunk a termékekre, illetve akarjuk-e, hogy tényleges készlet hiányában rendelhető státuszt kapjanak-e. Az ilyen egyéni akciókat a products séma discounts táblázatában kell megadni a NEW_DISCOUNT() függvény segítségével. Ez a táblázat már tényleges relációs kapcsolatban áll a terméklistával. A benne található termékek azonosítói (product_id) idegen kulcsok, a products.productlist adattáblára hivatkoznak.

Normális esetben csak olyan terméket szeretnénk rendelhető státuszra állítani – amennyiben nincs belőle tényleges készlet –, ami nem kifutó termék, azaz várható még belőle valaha szállítmány. A rendelési segédlistában – amit fájlban megkapunk – nincs külön oszlop arra vonatkozóan, hogy melyik termék kifutó és melyik aktív. Az egyetlen jelölés erre a "+" karakter a cikkszám végén. Amelyik termék cikkszámának végén ez a karakter szerepel, az kifutó, a többi aktív. A rendelési segédlistát önhatalmúlag nem változtathatjuk meg, nem adhatunk hozzá újabb oszlopot és nem törölhetünk belőle, így külön adattáblában kell tárolni azt, hogy melyik termék kifutó, mert az adatbázis nem fogad el "+" karaktereket a cikkszámok végén, az ugyanis csak szám lehet. Ezt az információt a products séma outgoing adattáblájában tároljuk. Ez egy kétoszlopos táblázat, egyik oszlopa (id, ami elsődleges kulcs, és csak számokat tartalmazhat) a kifutó termékek cikkszámait tartalmazza, másik oszlopa (comments) a hozzájuk fűzött megjegyzéseket. Irdatlan mennyiségű rekordról van szó mind a rendelési segédlistában, mint a kifutó termékek listájában, így ezeket importáláskor automatikusan a rendszer kezeli.

A products.not_outgoing nézet lényegében a products.outgoing adattábla negáltja, azokat a rendelési segédlistában található elemeket adja vissza, amelyek nincsenek benne a kifutó termékek listájában (products.outgoing).

A korrekt hibakezelés érdekében szándékosan nem definiáltunk tényleges relációs kapcsolatot a datasource és az outgoing adattáblák között, mert néha a kívülről megkapott rendelési segédlista erősen hiányos lehet, viszont a régi bejegyzéseket a kifutó termékekre vonatkozóan meg kell tartanunk, tehát néha meg kell tűrnünk bizonyos fokú inkonzisztenciát a két táblázat között, ami miatt nem szabad idegen kulcsokat definiálni közöttük!

A következő négy táblázat a szállítási díjakat határozza meg az egyes termékeknél a GETSHIPPINGFEE () függvény használata esetén. A szállítási díj beállításának szempontjait az alábbi pontokban foglaltuk össze, prioritás

szerint növekvő sorrendben (az adatbázis a szállítási díj beállítását ennek megfelelően végzi):

- 1. Először megvizsgáljuk, hogy az adott termék szállítható-e az MPL futárszolgálattal (tömege benne van-e még az általuk megadott keretben). A megadott termék tömegének ismeretében ezt а products.mpl shipping fees táblázatból tudjuk kideríteni az MPL SHIPPING AVAILABLE () logikai függvény segítségével. Ha a terméket viheti az MPL, akkor a szállítási díjat ők határozzák meg, üresen kell hagyni a terméklistában, értéke tehát NULL lesz. A termék tömege а terméklistából kapható meg, vagy а products.default weights és products.weights adattáblákból a GET PRODUCT WEIGHT () függvény segítségével;
- Ha a termék tömege nem ismert vagy nem szállítható az MPL futárszolgálattal, akkor minden termékre beállítjuk az alapértelmezett (legolcsóbb) szállítási díjakat. Ha nincs elég adatunk egy termékhez, akkor ez a szállítási díj vonatkozik rá;
- 3. Utána általánosságban meghatározzuk az egyes árucsoportokhoz és cikkcsoportokhoz tartozó, kollektív szállítási díjat az egyes árkategóriákban. Ezek а kollektív szállítási díjak а products.default shipping fees adattáblában találhatók meg;
- 4. Ezt követően pontosítjuk a kapott szállítási díjakat a products.hdt_products és products.hdt_shipping_fees adattáblák segítségével. Előbbi a bútorok lemért adatait tartalmazza és azokhoz egy kategória-azonosítót rendel, utóbbi pedig az egyes kategória-azonosítókhoz szállítási díjat rendel;
- A virtuális termékek megközelítőleg egyforma méretű és súlyú szetteket tartalmaznak, ezek egységes szállítási díjat kapnak a products.hdt_shipping_fees adattáblából.

A rengeteg feldolgozandó adat miatt nem ellenőrizzük azt, hogy egy termékre a megelőző lépésekben állítottunk-e már be szállítási díjat, mert így túl sokáig futna az algoritmus. Helyette prioritás szerint növekvő sorrendben állítgatjuk a szállítási díjakat (ami a termékek egyre szűkülő részhalmazait érinti), hagyva, hogy a fontosabb szempont felülírja az előző lépésben beállított, kevésbé fontos szempont alapján meghatározott szállítási díjat. Így ugyanazt az eredményt kapjuk, mintha mindent korrektül ellenőriztünk volna, csak sokkal gyorsabban. Ugyanez az alapelv vonatkozik az egyes termékek árainak és készleteinek beállítására is frissítéskor.

Most tekintsük a products séma grando_productlist nézetét. Ez a nézet nem csinál mást, mint lekérdezi az információs rendszerünkben tárolt összes termék adatát (beleértve a virtuális termékeket is) és olyan formátumban jeleníti meg azokat, amit a népszerű Grando online pláza eladói központjában fel lehet tölteni. A főbb különbségek saját weboldalunk megjelenítéséhez képest:

- Az első oszlop a termékek megnevezése, nem a cikkszáma. A cikkszám a második oszlopban található (seller_product_id néven);
- A termék megnevezése tartalmazza áruházunk monogramját, a termék nevét, a hozzá tartozó keresőszavakat és a termék cikkszámát a következő formátumban: MONOGRAM – TERMÉKNÉV – KERESŐSZAVAK – CIKKSZÁM;
- A szállítók kódja helyett a szállítók nevét tartalmazza a lista;
- Árucsoport, cikkcsoport, lefoglalt készlet, megjegyzések nem kellenek a listába, a Grando webáruház ezekkel az adatokkal jelenleg nem foglalkozik, így nem is tettük bele;
- Tartalmaz viszont egy shipping_fee_61_1 nevű oszlopot, aminek értéke "-" karakter vagy NULL lehet, előbbi esetben a termékre
alkalmazható az MPL-es díjszabás, utóbbi esetben nem, hanem fix szállítási díjat állítunk be rá. Ennek a mezőnek az értékét az MPL SHIPPING ENABLED() függvény segítségével határozzuk meg.

A Grando-ra szánt lista teljes mértékben kimeneti nézet, saját rendszerünk semmilyen módon nem épít erre a nézetre, ez csak egy exportálási lehetőség, hogy ismert webáruházakra is könnyedén fel tudjuk tölteni a termékeinket tömegesen. A nézetben megjelenő adatok meghatározásánál a Grando eladói központjának előírásait kell kötelező érvényűnek tekinteni.

Végül, tekintsük az orders.grando_orderlist nézetet! Ez a nézet lehetővé teszi, hogy a rendszerben tárolt megrendelési adatok ugyanolyan formátumban legyenek letölthetőek, mint ahogyan a Grando webáruház eladói központjában is letölthetőek a megrendelések. Visszafele is működik a művelet (lásd: Felhasználói dokumentáció), a webes felület végigmegy a Grando-ról letöltött megrendeléseket tartalmazó táblázat minden rekordján és az INSERT_INTO_ORDERLIST() függvény segítségével beszúrja a megrendelési tételeket, amennyiben azok még nem szerepelnek információs rendszerünk megfelelő adattáblájában.

3.2.2. A WEBES FELÜLET FELÉPÍTÉSE

Információs rendszerünk webes felhasználói felülete a következő főbb elemekből tevődik össze:

- index.php: ez a vásárlói oldal főoldala. Minden vásárlói szolgáltatás ezen az oldalon keresztül érhető el;
- admin.php: ez az eladói központ főoldala. Minden eladói szolgáltatás ezen az oldalon keresztül érhető el;

- login.php: az eladói központba történő bejelentkezéshez szükséges oldal. Itt kerül azonosításra a felhasználó, sikeres bejelentkezés után pedig lehetőséget kap adatainak gyors módosításaira;
- logout.php: az eladói központból történő kijelentkezéskor erre az oldalra navigál minket a rendszer. Kitisztítja a felhasználói adatokat és lezárja a session-t;
- download.php: ezen az oldalon keresztül tudjuk letölteni a különféle statisztikákat és egyéb táblázatokat az adatbázisból;
- upload.php: ezen az oldalon keresztül lehet adatokat feltölteni. Elméletileg minden implementálva van, ami technikailag egy feltöltéshez szükséges, de a gyakorlatban ezt az állományt soha nem használjuk információs rendszerünkben, semmilyen hivatkozás nincs rá, így valószínűleg változtatás nélkül nem is működik. A jövőbeli fejlesztések segítése érdekében hagytuk benne a rendszerben;
- definitions.php: ez tartalmazza az összes függvény, illetve globálisan használt változó definícióját. Az itt definiált függvények segítségével tudunk csatlakozni az adatbázishoz és adatokat lekérni vagy módosítani benne;
- maintheme.css: az oldal megjelenítésének stílusát leíró stíluslap. Ha bármilyen más színárnyalattal vagy háttérképekkel szeretnénk megjeleníteni a weboldalt, azt itt kell megadni.

A felülethez tartozó alkönyvtárak és a benne található elemek szerepe:

- config: itt található a csatlakozáshoz és import/export műveletek elvégzéséhez szükséges konfigurációs fájl. Szerepét már tárgyaltuk (lásd: 3.1.2. A webes felület telepítése, beállítása);
- lang: itt találhatók a webes felület nyelvi fájljai. Minden szöveg, ami a

weboldalon megjelenik, innen vagy az adatbázisból kerül kiolvasásra. Gyárilag csak az angol (en.php) és magyar (hu.php) nyelvi fájlok találhatók meg benne. Az ide vonatkozó megkötéseket később tárgyaljuk;

- images: a weboldalon megjelenített összes kép itt található, alkönyvtárakra bontva, csoportosítva;
- apps: a weboldalba beépített Java applet-ek találhatók meg benne.
 Jelenleg csak 1 applet-et tartalmaz, mely a weboldalon diagramok kirajzolásához szükséges.

Meg kell jegyeznünk, hogy a weboldalak által a webböngészőnek elküldött kód a legtöbb helyen megfelel a W3C szabványnak, tartalma érvényes "HTML 4.01 Transitional" weblap. Ez fontos szempont, amikor garantálni akarjuk, hogy weboldalunk a legtöbb ismert webböngészőben egyformán jelenjen meg. Az oldal fejlesztése során ezt szigorúan betartottuk. Sajnos az adatbázisban tárolt felhasználói adatok miatt a validitás nem mindig garantálható, de ilyen jellegű problémák csak ritkán fordulnak elő. A másik megjegyzés, hogy kis kerülőutakkal sikerült teljes mértékben mellőzni a JavaScript nyelv használatát a webes felületben. A JavaScript nyelv használatát azért igyekeztünk teljes mértékben mellőzni, mert – konyhanyelven és kissé pontatlanul fogalmazva – minden webböngészőben másképpen működik, ami nem megengedhető fontos adatok kezelésénél. A látványos design helyett tehát a biztonságot részesítettük előnyben.

Minden önállóan futtatandó (tehát nem az include-utasítással beillesztett) PHP kódnak a legelső utasítása a következő:

error_reporting(0);

Ezt azt jelenti, hogy ideiglenesen minden hibaüzenetet kikapcsolunk, a webszerver naplóállományába semmilyen hibaüzenet nem kerülhet. Erre azért van szükség, mert a távoli tartalmak, vagy egyéb, beolvasott adatok esetén

nem tudjuk garantálni, hogy semmilyen probléma nem merül fel az adatokkal kapcsolatban, ami a webszervert érinti. Nagyon ritka, hogy egy ilyen oldal tökéletesen hibátlanul tölt be, viszont a naplóállomány folyamatosan csak növekszik, és például egy éveken keresztül folyamatosan aktív webszerveren betelítheti az összes szabad tárhelyet. Éppen ezért akkor, amikor már biztosak vagyunk abban, hogy mi nem hibáztunk a kód megírásakor, és az összes PHP programunk minden bemenetre úgy fut le, ahogyan elvárjuk tőle, érdemes a fenti utasítással kezdeni minden PHP kódunkat.

3.2.2.1. A vásárlói oldal felépítése

A vásárlói oldalt tartalmazó fájl az index.php. Ez ugyanúgy kezdődik, mint egy hagyományos HTML oldal, ugyanúgy megvannak benne a dokumentum típusának definíciói és az egyes metaadatok. A különbség a "<?php ... ?>" részekbe írt kódban rejlik. Ezekben találhatók a tényleges tartalom megjelenítéséért felelős PHP kódok, a többi HTML kód lényegében csak az oldal – korábban tárgyalt – elrendezését határozza meg.

Az oldal fejlécében a következő PHP kód található:

```
<?php
    error_reporting(0);
    include("definitions.php");
?>
```

A korábbiak tükrében világos, hogy az első utasítás mit csinál. A második utasítás lefuttatja a definitions.php fájlban található kódot, aminek következtében értelmet nyer minden olyan elem, amire a későbbiekben a weblap hivatkozik.

A másodikként beillesztett PHP kód a következő:

```
<?php
    //Handle shopping cart informations
    echo "<a href=\"".$phpfilename."?show_cart=yes\">\n";
    echo "<img src=\"images/icons/shopping_cart.png\" alt=\"".</pre>
```

```
$lang["cart icon"]."\">\n";
       echo "</a><br>\n";
       if(isset($ SERVER["REMOTE ADDR"]) == true) {
              $client_address = trim($_SERVER["REMOTE_ADDR"]);
              $add_to_cart = get_array_value($_GET,"add_to_cart");
              if($add to cart>=1) {
                      $quantity value = 1;
                      $quantity = get_array_value($_GET,"quantity");
                      if($quantity>=1){ $quantity_value = $quantity; }
                      add_to_cart($add_to_cart, $quantity_value, $client_address);
               }else{
                      $remove from cart = get array value($ GET, "remove from cart");
                      if ($remove from cart>=1) {
                             remove_from_cart($remove_from_cart, $client_address);
                      }
              get shopping cart($client address);
```

Ez lekérdezi az adatbázisból a bevásárlókosár tartalmát, illetve lehetőséget biztosít a bevásárlókosár megnyitására. A kód természetesen elvégez minden ellenőrzést, ami a helyes lekérdezéshez szükséges.

A harmadikként beillesztett "<?php echo \$PHP_SELF; ?>"kód nem tesz mást, mint lekérdezi az oldal azonosítóját és beágyazza azt a HTML kódba a keresőmező űrlapjához cél-hivatkozásként.

A negyedikként beágyazott "<?php get_search_text(); ?>" kód az URL-paraméterekben rákeres a keresőmező értékére, ha nem találja, akkor üres szöveget ad vissza.

Az ötödikként beágyazott "<?php open_category_list(); ?>" kód a weboldal megfelelő helyén megnyitja a termékekhez tartozó kategórialistát. Ez jelenleg még csak a főkategóriát (árucsoportot) érinti.

A tényleges tartalom megjelenítéséért a hatodikként beágyazott kód felel:

```
<?php
//Handle orders
if(strtolower(trim(get_array_value($_GET, "show_cart")))=="yes"){
    if(isset($_SERVER["REMOTE_ADDR"])==true){
        $client_address = trim($_SERVER["REMOTE_ADDR"]);
        list_shopping_cart($client_address);
    }
}else{
    $id = get_array_value($_GET,"id");
    if($id>=1){
        //Show product page
        $image_number_value = 1;
        $image_number = get_array_value($_GET,"image_number");
        if($image_number>=1){ $image_number_value = $image_number; }
}
```

hogy milyen tartalmat akarunk megjeleníteni. Ez megnézi, Ha а bevásárlókosarat, akkor megjeleníti annak tartalmát. Ha nem а bevásárlókosarat, akkor megnézi, hogy melyik termékhez tartozó terméklapot szeretnénk megnézni (cikkszám alapján), és megjeleníti azt. Ha erre sem talál megfelelő adatot, akkor megjeleníti a vásárlói oldalhoz tartozó terméklistát.

A hetedikként beágyazott PHP kód megjeleníti az elérhetőségi információkat tartalmazó lista fejlécét:

```
<?php echo $lang["contact"].": ".$corporation name; ?>
```

A nyolcadik PHP kód megjeleníti az áruházunk elérhetőségi információit tartalmazó listát:

```
<?php get_house_list(); ?>
```

Végül pedig, a kilencedik PHP kód megjeleníti az eladói központra mutató hivatkozást:

```
<?php echo $lang["retail center"]; ?>
```

Erre rákattintva megnyílik az eladói központ bejelentkezési oldala.

3.2.2.2. Az eladói központ felépítése

Az eladói központ weboldalát tartalmazó fájl az admin.php. Ennek felépítése teljesen hasonló a vásárlói oldalhoz, így a továbbiakban csak a kettőjük közötti különbségeket részletezzük.

Az egyik ilyen különbség, hogy ez a fájl már nem a szokásos HTML kóddal kezdődik, hanem azonnal PHP kódot írunk, ugyanis, még mielőtt bármit megjeleníthetnénk, ellenőrizni kell, hogy az oldalra látogató felhasználó

egyáltalán jogosult-e az oldalon található tartalom megtekintésére. Amennyiben nem, akkor nem jelenítünk meg semmit, hanem azonnal átirányítjuk a bejelentkezési oldalra. Amennyiben jogosult, akkor az oldal szokásos előkészítése után megjelenítjük a tartalmat. Mindezt az alábbi PHP kód hajtja végre:

```
<?php
       error_reporting(0);
       //Check session
       session start();
       $canlogin = false;
       if(isset($ SESSION["current login name"])==true){
              if(strlen(trim($_SESSION["current_login_name"]))>=1){
                      if(isset($_SESSION["current_password"])==true){
                      if(strlen(trim($ SESSION["current password"]))>=1) {
                      if(isset($ SESSION["current_address"]) == true) {
                      if(strlen(trim($_SESSION["current_address"]))>=1){
                      if($ SESSION["current address"]==$ SERVER["REMOTE ADDR"]){
                             $canlogin = true;
                      }
              }
       //Back to login page if cannot login
       if($canlogin==false){
             header("location:login.php");
              exit;
       }
```

A következő különbség a megjelenített oldal tetején található "Login menü": ebben megtaláljuk, hogy milyen néven vagyunk éppen bejelentkezve, lehetőséget kapunk adataink módosítására és a kijelentkezésre:

```
<?php
$login_name = trim(get_array_value($_SESSION, "current_login_name"));
if(strlen(trim($login_name))>=1){
        echo $lang["logged_in_as"]." <b><a href=\"".
            $phpfilename."?menu=account_settings\">".
            $phpfilename."?menu=account_settings\">".
            $login_name."</a></b> (<a href=\"logout.php\">".
            $lang["logout"]."</a>)\n";
}
```

A harmadik különbség hogy a weblap bal oldalán a termékkategóriák helyett az eladói központ főmenüje kapott helyet:

```
<?php
//Go to Customer site
echo "<table width=\"100%\">\n";
echo "<a target=\"_blank\" href=\"index.php\">".
$lang["customer_site"]."</a>\n";
echo "\n";
echo "\n";
//Account settings
```

```
echo "\n";
echo "<a href=\"".$phpfilename."?menu=account_settings\">".
     $lang["account_settings"]."</a>\n";
echo "\n";
echo "\n";
//Manage products
echo "\n";
echo "<a href=\"".$phpfilename."?menu=manage products\">".
     $lang["manage products"]."</a>\n";
echo "\n";
echo "\n";
//Manage manufacturers
echo "\n";
echo "<a href=\"".$phpfilename."?menu=manage_manufacturers\">".
     $lang["manage_manufacturers"]."</a>\n";
echo "\n";
echo "\n";
//Manage orders
echo "\n";
echo "<a href=\"".$phpfilename."?menu=manage_orders\">".
     $lang["manage orders"]."</a>\n";
echo "\n";
echo "\n";
//Manage orders
echo "\n";
echo "<a href=\"".$phpfilename."?menu=accountancy\">".
     $lang["accountancy"]."</a>\n";
echo "\n";
echo "\n";
//View statistics
echo "\n";
echo "<a href=\"".$phpfilename."?menu=view statistics\">".
     $lang["statistics"]."</a>\n";
echo "\n";
echo "\n";
//Forum
echo "\n";
echo "<a href=\"".$phpfilename."?menu=forum\">".$lang["forum"]."</a>\n";
echo "\n";
echo "\n";
//Special discounts
echo "\n";
echo "<a href=\"".$phpfilename."?menu=special_discounts\">".
     $lang["special discounts"]."</a>\n";
echo "\n";
echo "\n";
//Import/Export
echo "\n";
echo "<a href=\"".$phpfilename."?menu=import export\">".
     $lang["import_export"]."</a>\n";
echo "\n";
```

A másik legfontosabb különbség az oldal által megjelenített tartalomban rejlik: itt jóval több dolgot tekinthetünk meg, mint a vásárlói oldalon:

```
<?php
```

?>

```
//Import data from files
$import_name = strtolower(trim(get_array_value($_POST,"import_name")));
if(strlen(trim($import_name))>=1){
    import_data($import_name);
}else{
    //Apply changes before (re)loading data
    change_database();
    //Open selected product
    $id = get_array_value($_GET,"id");
    if($id>=1){
        $image number_value = 1;
    }
}
```

```
$image number = get array value($ GET, "image number");
                      if($image_number>=1) { $image_number_value = $image_number; }
                      $display_tab = get_array_value($_GET,"display_tab");
                     open_product_page($id, $image_number_value, $display_tab, true);
              }else{
                      //Edit selected manufacturer
                      $manufacturer id = get array value($ GET,"edit manufacturer");
                      if($manufacturer id>=1) {
                             edit manufacturer($manufacturer id);
                      }else{
                             //Edit selected product
                             $edit id = get array value($ GET, "edit product");
                             if($edit id>=1) {
                                     edit_product($edit_id);
                             }else{
                                     //Edit selected order
                                     $order id = get array value($ GET, "edit order");
                                     if($order id>=1){
                                            edit_order($order_id);
                                     }else{
                                            $current menu =
                                                    strtolower(trim(
                                                    get_array_value($_GET,"menu")));
                                            switch($current_menu) {
                                                    //Manage manufacturers
                                                    case "manage manufacturers":
                                                          manufacturer menu(); break;
                                                    case "new_manufacturer":
                                                           edit manufacturer(0); break;
                                                    //Manage products
                                                    case "manage products":
                                                          products_menu(true); break;
                                                    case "upload_product":
                                                           edit product(0); break;
                                                    case "upload collection":
                                                          edit_product(-1); break;
                                                    //Manage orders
                                                    case "manage_orders":
                                                          orders menu(); break;
                                                    //Accountancy
                                                    case "accountancy":
                                                           accountancy_menu(); break;
                                                    //Forum
                                                    case "forum": forum menu(); break;
                                                    //Manage accounts
                                                    case "account_settings":
                                                           accounts menu(); break;
                                                    //Special discounts
                                                    case "special_discounts":
                                                           special discounts(); break;
                                                    //Update products
                                                    case "update products":
                                                          update products(); break;
                                                    //Import/Export
                                                    case "import_export":
                                                           import_export(); break;
                                                    //Generate downloads
                                                    case "download":
                                                           generate_download(); break;
                                                    //View statistics
                                                    default: view statistics(); break;
                                            }
                                   }
                        }
                    }
              }
       }
?>
```

Mielőtt az oldal bármit megjelenítene, előbb kérvényezi az adatbázisban az

adatok frissítését, hogy a felhasználók által kért módosítások azonnal megjelenjenek az oldalon. Ezt követően, prioritás szerint csökkenő sorrendben megvizsgálja a weboldal a paraméterekben megadott adatokat, melyek meghatározzák, hogy milyen tartalmat szeretnénk megjeleníteni. Az adatok frissítése szintén prioritás szerint csökkenő sorrendben történik.

Ha mást nem mondunk, akkor a megrendelések adataiból készített statisztika kerül megjelenítésre, amit egy, az oldalba beépített Java-applet jelenít meg, diagram formájában. A diagram kirajzolásához szükséges adatokat az applet a weboldaltól kapja meg paraméterekben.

A bejelentkezési oldal tartalmilag az eladói központhoz tartozik. Miután vagy közvetlenül ide navigáltunk, vagy szabálytalan hozzáférési kísérlet miatt át lettünk ide irányítva, ez az oldal bekéri a felhasználónevünket és jelszavunkat. Az oldal a következő PHP kóddal kezdődik:

```
<?php
       error reporting(0);
       //Check session
       session start();
       $account_changed = false;
       if(isset($ POST["account changed"])==true) {
              if(strtolower(trim($_POST["account_changed"])) == "yes") {
                      $account_changed = true;
              }
       if ($account changed==false) {
              if(isset($ SESSION["current login name"])==true) {
                      if(strlen(trim($ SESSION["current login name"]))>=1){
                              if(isset($ SESSION["current_password"])==true) {
                              if(strlen(trim($_SESSION["current_password"]))>=1){
                              if(isset($_SESSION["current_address"])==true){
                              if(strlen(trim($ SESSION["current address"]))>=1){
                              if($ SESSION["current address"]==$ SERVER["REMOTE ADDR"]){
                                     header("location:admin.php");
                                     exit;
                      }
              }
       }
```

A fenti kód ellenőrzi, hogy változtak-e a profilbeállítások, és amennyiben igen, akkor tárolja ezt az információt az oldalban. Ezt követően ellenőrzi, hogy be vagyunk-e már jelentkezve az eladói központba. Ha igen, és nem változtak a

profilbeállítások, akkor automatikusan átirányít minket az eladói központ főoldalára.

Ha még nem vagyunk bejelentkezve az eladói központba, akkor a következő PHP kód fut le a továbbiakban:

```
<?php
      //Check login if given
      $login_name = get_array_value($_POST,"login_name");
      $password = get_array_value($_POST,"password");
      $checklogin = false;
      if(strlen(trim($login name))>=1){
            if(strlen(trim($password))>=1){
                  if(check_login($login_name, $password) == true) {
                         $checklogin = true;
                   }else{
                         echo "<b>".$lang["login incorrect"]."</b>\n";
                   }
      //Login window
      if($checklogin==false){
            echo "\n";
            echo "<h1>".$lang["login"]."</h1>\n";
            echo "<form name=\"login\" method=\"post\" action=\"".$PHP SELF."\">\n";
            echo "<i>".$lang["login_conditions"]."</i>\n";
            echo "\n";
            echo "\n";
            echo "<b>".$lang["login_name"].": </b>\n";
            echo "<input type=\"text\" name=\"login name\" size=\"20\"".
                   " maxlength=\"255\">\n";
            echo "\n";
            echo "<b>".$lang["password"].": </b>\n";
            echo "<input type=\"password\" name=\"password\" size=\"20\"".
                   " maxlength=\"255\">\n";
            echo "\n";
            echo "\n";
            echo "<input type=\"submit\" value=\"".$lang["login"]."\">\n";
            echo "</form>\n";
            echo "<a href=\"index.php\">".
                   $lang["back_to_main_page"]."</a>\n";
            echo "\n";
      }else{
            //Personal details
            account_settings($login_name, $password, true);
      }
```

A kód lényege, hogy ellenőrizzük a megadott felhasználónevet és jelszót, ha rendben van, akkor megjelenítjük a fiókbeállításokat, ha nincs rendben, vagy nincs megadva (mert például most először navigálunk az oldalra), akkor megjeleníti a bejelentkező ablakot. Ha az oldal elfogadta a felhasználónevünket és jelszavunkat, akkor a "Tovább az eladói központra"-gombra való kattintással történik meg a tényleges bejelentkezés, a session-adatok ekkor kerülnek ténylegesen beállításra.

A kijelentkezési oldal mindössze ennyiből áll:

```
<?php
    error_reporting(0);
    session_start();
    $_SESSION["current_login_name"] = null;
    $_SESSION["current_password"] = null;
    $_SESSION["current_address"] = null;
    session_destroy();
    header("location:login.php");
    exit;
?>
```

Ez kinullázza a session-adatokból a felhasználónevet, jelszót és IP-címet, majd lezárja a session-t és átirányít minket a bejelentkezési oldalra.

3.2.2.3. A definíciós fájl

A definíciós fájl (definitions.php) egy önállóan nem megnyitható állomány, nem küld a böngészőnek semmilyen kimenetet, kizárólag más oldalakba beágyazva használható. Méret szempontjából ez teszi ki a felület programjának legnagyobb részét. Az összes, valódi megjelenítéssel rendelkező oldal használja (de nem feltétlenül csak azok), valójában az egész felület erre épül. Működése a következőképpen alakul:

- Először definiáljuk a felület működéséhez szükséges összes globális változót, melyeket rendszeresen használunk;
- Utána definiáljuk a felületen használt összes függvényt (melyek sok esetben felhasználják a globális változókat).

A definíciók sorrendje kötött, nem cserélhetők fel, mert sok esetben egymásra építenek! A legfontosabb globális változókat a következő táblázatban foglaltuk össze (definiálásuk sorrendjében):

Név	A DATTÍPUS	Leírás
\$phpfilename	STRING	A definíciós fájlt meghívó weboldal neve

Név	Adattipus	Leírás
\$maximum_timeout	INTEGER	Maximálisan engedélyezhető futási idő. Bizonyos script-ek futtatása hosszú időt vehet igénybe (például: adatimportálás, terméklista frissítése), ilyenkor nem biztos, hogy elég lesz a script-nek az alapértelmezés szerinti 30 másodperces időkorlát, tehát ezt feljebb kell húznunk. Ennek a változónak az alapértelmezett értéke 3600 (másodperc), tehát a maximálisan engedélyezhető futási idő alapértelmezés szerint 1 óra lehet.
\$language_file	STRING	A használni kívánt nyelvi fájl neve. Az oldal megnyitásakor a rendszer automatikusan felismeri a böngésző nyelvét, és az őt reprezentáló két betűs kódot (kisbetűsre konvertálás után) kiegészíti a ".php" résszel (illetve az elérési útvonallal), és ez lesz a nyelvi fájl neve. Ha ez nem található, akkor alapértelmezés szerint "en.php" lesz az értéke.
\$exclude_fields	ARRAY	Olyan paraméterek listája, melyeket csak egyszer akarunk használni, a kiegészített URL-ekhez már nem akarjuk hozzáfűzni őket, mert az hibás működést eredményezne. A kulcsok egész számok, az értékek szövegek.
\$connection_string	STRING	Az adatbázis eléréséhez használt kapcsolatleíró
<pre>\$primary_identifier</pre>	STRING	A termékek elsődleges azonosítójának neve. Az adatbázisból olvassuk ki értékét, ez gyárilag "Cikkszám".
\$currency_used	STRING	A pénzügyi adatoknál használt valuta rövidítése. Az adatbázisból olvassuk ki értékét, gyárilag ez "HUF" (Forint).
\$corporation_name	STRING	Webáruházunk neve. Az adatbázisból olvassuk ki értékét, gyárilag ez "Fake Lakberendezési Kft.".
\$weight_measure	STRING	Az információs rendszerben használt tömeg mértékegysége. Az adatbázisból olvassuk ki, értéke alapértelmezés szerint gramm.
\$available_delimiters	ARRAY	Tömböket tartalmazó tömb: CSV táblázatok importálása és exportálása során használható mezőelválasztók karakterkódjait tartalmazza, illetve a script-ekben a rájuk vonatkozó, alias-ként használt karaktereket.
\$special_product_groups	ARRAY	Különleges árucsoportok kódjait tartalmazó tömb, ezekre az árucsoportokra a rendszer külön havi forgalmi kimutatást készít.

Név	Adattípus	Leírás
\$lang	ARRAY	Nem a definitions.php fájlban hozzuk létre, de rendkívül fontos szerepe van. Ez tárolja a nyelvi fájlból kiolvasott szövegeket. Mind a kulcsok, mind a hozzájuk tartozó értékek adattípusa szöveg.

20. táblázat. A felület globális változói

Mind a vásárlói oldal, mind az eladói központ kizárólag a definíciós fájlban található adatok és függvények segítségével férhet hozzá az adatbázishoz, közvetlen elérés nincs (ha csak a definíciós fájlban található közvetlen, nem módosítható lekérdezéseket nem tekintjük annak).

A felület működéséhez szükséges függvényeket az alábbi táblázatban találhatjuk:

Név	Visszatérési érték	Leírás
<pre>get_logical_text(\$input_value)</pre>	STRING	Logikai értékekhez tartozó szövegeket ad vissza: IGAZ (t) érték esetén a nyelvi fájlban definiált IGEN-nek megfelelő szöveget, HAMIS (f) érték esetén a nyelvi fájlban definiált NEM-nek megfelelő szöveget, különben NULL-t. A bemenő paraméter adattípusa szöveg (1 karakter).
<pre>get_db_number(\$input_number)</pre>	Bemenettől függ	Ha a bemenő adat nem üres, akkor azt adja vissza, egyébként NULL-t.
<pre>get_db_string(\$input_string)</pre>	STRING	Ugyanazt csinálja, mint az előző, de ha az eredmény nem NULL, akkor idézőjelbe teszi.
<pre>get_optional_value(\$input_value)</pre>	Bemenettől függ	Ha a bemenő adat nem üres, akkor azt adja vissza, egyébként "-" karaktert.
<pre>get_row_value(\$query_result, \$index, \$column)</pre>	Lekérdezéstől és bemenettől függ	Az adatbázisból lekérdezett táblázat megadott sorában a megadott elemet adja vissza.
<pre>get_image_from_url(\$image_url, \$width, \$height, \$alt)</pre>	STRING	A paraméterekben megadott URL-ből HTML-kódot generál képek megjelenítéséhez a többi megadott paraméterrel.

Név	Visszatérési érték	Leírás
get_array_value(\$array, \$index)	Bemenettől függ	Ha a megadott tömb eleme létezik, akkor azt adja vissza, egyébként NULL-t.
<pre>get_normal_url()</pre>	STRING	Visszaadja az aktuális URL-t, de az \$exclude_fields tömbben található paraméterek nélkül.
append_url(\$new_parameter, \$new_value)	STRING	Kiegészíti az aktuális URL-t a megadott paraméterrel, de az \$exclude_fields tömbben található paraméterek nélkül.
<pre>import_params_to_form(\$handle_excludes, \$source)</pre>	Eljárás	Rejtett mezőként beilleszti az URL-ben lévő paramétereket a környezetben található HTML- űrlapba.
<pre>prepare_file(\$access_directory, \$filename)</pre>	STRING	A paraméterként megadott fájl és az őt tartalmazó, paraméterként megadott könyvtár, valamint az operációs rendszer által használt elválasztó alapján visszaadja a fájl elérési útvonalát, hogy ezt követően a fájlt már gond nélkül el lehessen érni.
return_search_text()	STRING	Visszaadja az URL-ben található keresési kifejezést, ha van.
get_search_text()	Eljárás	Kiírja az URL-ben található keresési kifejezést, ha van.
open_category_list()	Eljárás	Megnyitja a termék-kategóriák listáját (ez jelenleg csak főkategória, vagy árucsoport).
<pre>get_product_image(\$db_records, \$index)</pre>	Eljárás	A terméklistában megnyitja az adott sorszámú rekordhoz tartozó elsődleges képet.
<pre>get_product_title(\$db_records, \$index)</pre>	Eljárás	A terméklistában megnyitja az adott sorszámú rekordhoz tartozó megnevezést.
<pre>get_product_description(\$db_records, \$index, \$length)</pre>	Eljárás	A terméklistában megnyitja az adott sorszámú rekordhoz tartozó leírást.
<pre>get_product_price(\$db_records, \$index)</pre>	Eljárás	A terméklistában megnyitja az adott sorszámú rekordhoz tartozó ára(ka)t.
rows_per_page()	Eljárás	Listákban a lista elemszámának beállítására ad lehetőséget az érintett weboldalon.

Név	Visszatérési érték	Leírás
<pre>display_paging(\$numrows, \$numrows_total, \$display_rows, \$start_row)</pre>	Eljárás	Sok elemes lista esetén az érintett weboldalon lehetőséget ad lapozásra.
<pre>open_product_list(\$filter_rule, \$order_by, \$startrow, \$displayrows, \$retail_center)</pre>	Eljárás	A megadott paramétereknek eleget tevő termékek listájának megnyitása
<pre>get_correct_name(\$family_name, \$given_name)</pre>	STRING	A megadott név megjelenítése a területi beállításnak megfelelően
<pre>submit_post(\$id, \$database_link, \$reply)</pre>	BOOLEAN	Hozzászólás elküldése, és az eredményesség visszaadása
<pre>make_combobox(\$combobox_name, \$table_name, \$column_name, \$index_name, \$database_link, \$default_value)</pre>	Eljárás	Az adatbázis adott táblájának adott oszlopában szereplő értékekből az adott indexekhez rendelve adott alapértelmezett értékkel legördülő menüt készít egy tetszőleges HTML űrlaphoz.
<pre>open_forum(\$id, \$displayrows, \$database_link, \$reply)</pre>	Eljárás	Fórum megnyitása
<pre>open_product_page(\$id, \$imagenumber, \$displaytab, \$retail_center)</pre>	Eljárás	Terméklap megnyitása
add_to_cart(\$id, \$quantity, \$address)	BOOLEAN	Termék kosárba helyezése és az eredményesség visszaküldése
<pre>remove_from_cart(\$id, \$address)</pre>	BOOLEAN	Termék eltávolítása a kosárból és az eredményesség jelzése
<pre>get_shopping_cart(\$address)</pre>	Eljárás	Hányféle termék található a bevásárlókosárban?
<pre>get_pay_methods(\$database_link)</pre>	Eljárás	Fizetési módok megjelenítése
<pre>prepare_order(\$database_link, \$cart_products)</pre>	Eljárás	Rendelés előkészítése és elküldése
list_shopping_cart(\$address)	Eljárás	Bevásárlókosár megnyitása
<pre>get_house_list()</pre>	Eljárás	Telephelyek és elérhetőségeik megjelenítése
check_login(\$login_name, \$password)	BOOLEAN	Felhasználó azonosítása. Ha sikeres a művelet, akkor az eredmény TRUE, egyébként FALSE.
<pre>new_user(\$database_link, \$family_name, \$given_name, \$email, \$login_name, \$password, \$superuser)</pre>	BOOLEAN	Felhasználó létrehozása. Ha a művelet sikeres, akkor az eredmény TRUE, egyébként FALSE.
remove_user(\$id)	Eljárás	Felhasználó törlése
<pre>magane_admins(\$database_link, \$login_name)</pre>	Eljárás	Felhasználók kezelése

Név	Visszatérési érték	Leírás
<pre>account_settings(\$login_name, \$password, \$go_retail_center)</pre>	Eljárás	Fiókbeállítások megjelenítése
edit_product(\$id)	Eljárás	Megadott termék szerkesztése
new_product()	BOOLEAN	Új termék feltöltése / régi termék módosítása. Ha sikeres, akkor az eredmény TRUE, egyébként FALSE.
open_manufacturer_list(\$filter_rule, \$order_by, \$startrow, \$displayrows)	Eljárás	Szállítók kilistázása
edit_manufacturer(\$id)	Eljárás	Megadott szállító szerkesztése
new_manufacturer()	BOOLEAN	Új szállító feltöltése / régi szállító módosítása. Ha sikeres, akkor az eredmény TRUE, egyébként FALSE.
open_order_list(\$filter_rule, \$order_by, \$startrow, \$displayrows)	Eljárás	Megrendelések kilistázása
edit_order(\$id)	Eljárás	Adott megrendelés kezelése
<pre>set_order_status()</pre>	BOOLEAN	Adott megrendelés státuszának átállítása. Ha sikeres, akkor az eredmény TRUE, egyébként FALSE.
<pre>new_transaction()</pre>	BOOLEAN	Új tranzakció könyvelése. Ha sikeres a művelet, akkor az eredmény TRUE, egyébként FALSE.
<pre>open_accountancy(\$filter_rule, \$order_by, \$startrow, \$displayrows)</pre>	Eljárás	Könyvelés megnyitása
proceed_to_download(\$download_name, \$download_title)	Eljárás	Űrlapot készít a letöltéshez, mely segítségével elnavigálhat minket a rendszer a tényleges letöltési oldalhoz.
generate_download_link(\$download_title, \$filename)	Eljárás	Űrlapot készít a rendszer a letöltéshez, melyhez letöltési linket is generál a letöltési oldalon, miután a rendszer elkészítette a letöltendő anyagot. Segítségével a rendszer elnavigál minket a letöltő script- hez, ahol megkezdődik az állomány tényleges letöltése.
generate_download()	Eljárás	Legeneráltatja a rendszerrel a konkrét letöltési tartalmaz. Tulajdonképpen a letöltési oldal tartalmát ez generálja.

Név	Visszatérési érték	Leírás
<pre>generate_upload_form(\$upload_name, \$filename, \$upload_text)</pre>	Eljárás	Űrlapot generál állományok feltöltéséhez. Nem használjuk, de a későbbi verziókban szükség lehet rá, így nem távolítottuk el.
view_statistics()	Eljárás	Statisztikák megjelenítése
forum_menu()	Eljárás	Fórum megnyitása
remove_discount(\$id)	Eljárás	Eltávolítja a megadott különleges akciót.
new_discount()	BOOLEAN	Új különleges akciót regisztrál az adatbázisban a felületen megadott adatokkal. Sikeres regisztráció esetén IGAZ értékkel, sikertelen regisztráció vagy hiba esetén pedig HAMIS értékkel tér vissza.
<pre>open_special_discounts(\$filter_rule, \$order_by, \$startrow, \$displayrows)</pre>	Eljárás	Megjeleníti a különleges akciók kezelését a weboldalon.
special_discounts()	Eljárás	Előkészíti a különleges akciókat megjelenítő tartalom lekérését.
update_all_products(\$database_link)	Eljárás	Frissíti a termékeket az adatbázisban és megjeleníti a frissítés eredményét.
update_products()	Eljárás	Kezdeményezi az összes termék adatának frissítését, majd kiírja a futási időt.
<pre>make_combobox_for_encodings(\$default_encoding)</pre>	Eljárás	Legördülő menüt készít egy tetszőleges űrlapon a használt karakterkódolásokhoz az adott alapértelmezett értékkel.
<pre>make_combobox_for_delimiters(\$default_delimiter)</pre>	Eljárás	Az előző eljáráshoz hasonlóan legördülő menüt készít a használt mezőelválasztókhoz a megadott alapértelmezéssel.
<pre>import_export()</pre>	Eljárás	Megjeleníti az Import/Export menüt.
<pre>import_data(\$import_name)</pre>	Eljárás	Elvégzi a tényleges importálást, végigolvassa az importadatokat, majd az adatbázis függvényei segítségével importáltatja a beolvasott adatokat, amennyiben lehetséges.
change_database()	Eljárás	Adatbázis módosítása

Név	Visszatérési érték	Leírás
manufacturer_menu()	Eljárás	Szállítók megnyitása
<pre>products_menu(\$retail_center)</pre>	Eljárás	Termékek megnyitása
orders_menu()	Eljárás	Megrendelések megnyitása
accountancy_menu()	Eljárás	Könyvelés megnyitása
accounts_menu()	Eljárás	Fiókbeállítások megnyitása

21. táblázat. A felület működéséhez szükséges függvények

A függvényeket a táblázatban megadott sorrendben definiáljuk, sokszor előfordul a függvényeknél is, hogy egymásra hivatkoznak.

3.2.2.4. Diagramok kirajzolása

Mint említettük, a diagramok kirajzolásáért az eladói központban egy Java applet felel. Ez az applet a DiagramPlotterApplet, melynek forráskódja a DiagramPlotterApplet.java és Alignment.java fájlban található. Előbbi egy rendes Java-osztályt jelent, a második pedig egy felsoroló típus. Az osztályhoz tartozó adattagokat a következő táblázatban láthatjuk:

Név	Adattípus	Leírás
labels	String[]	A diagram címkéit tartalmazó tömb
values	double[]	Az egyes címkékhez tartozó értékek tömbje
totalQuantity	double	Az értékek összege (a teljes mennyiség)
title	String	Diagram címe
type	int	Diagram típusa
displayChange	String	Rövid kezelési utasítás (hint)

22. táblázat. A DiagramPlotterApplet osztály adattagjai

Az osztályhoz tartozó függvényeket az alábbi táblázatban láthatjuk:

Név	Visszatérési érték típusa	Leírás
init()	void	Eseményvezérelt grafikus felület inicializálása
stop()	void	Eseményvezérelt grafikus felület leállítása

Név	Visszatérési érték típusa	Leírás
drawTextCenter(Graphics g, String text, int x, int y)	void	A középre igazított text szöveg kirajzolása a g grafikus felületen az (x, y) koordinátákhoz
DrawRotatedText(Graphics2D g, String text, int x, int y, double angle, Alignment align)	void	align-alapján igazított, angle-szöggel elforgatott text szöveg kirajzolása a g grafikus felületen az (x,y) koordinátákhoz
DrawRotatedText(Graphics2D g, String text, int x, int y, double angle)	void	Ugyanazt csinálja, mint az előző függvény, alapértelmezett igazítást használva.
drawDiagramValue(Graphics g, int x, int y, double value, double percentage)	void	A value érték és/vagy a percentage százalék kirajzolása a g grafikus felületen az (x,y) koordinátákhoz
DrawRotatedValue(Graphics2D g, int x, int y, double value, double percentage, double angle, Alignment align)	void	align-alapján igazított, angle-szöggel elforgatott value érték és percentage százalék kirajzolása az (x,y) koordinátákhoz a g grafikus felületen.
DrawRotatedValue(Graphics2D g, int x, int y, double value, double percentage, double angle)	void	Ugyanazt csinálja, mint az előző függvény, alapértelmezett igazítást használva.
mouseEntered(MouseEvent e)	void	Akkor fut le, amikor az egérkurzor belép a grafikus felületre
mouseExited(MouseEvent e)	void	Akkor fut le, amikor az egérkurzor kilép a grafikus felületről
mouseClicked(MouseEvent e)	void	Akkor fut le, amikor kattintunk a grafikus felületen
mousePressed(MouseEvent e)	void	Akkor fut le, amikor lenyomjuk az egér gombját a grafikus felületen
mouseReleased(MouseEvent e)	void	Akkor fut le, amikor fölengedjük az egér gombját a grafikus felületen
mouseMoved(MouseEvent e)	void	Akkor fut le, amikor mozgatjuk az egeret a grafikus felületen
mouseDragged(MouseEvent e)	void	Akkor fut le, amikor húzunk az egérrel a grafikus felületen (egérgomb lenyomva + mozgatás)
paint(Graphics g)	void	A grafikus felület és tartalmának kirajzolásához / újrarajzolásához szükséges függvény

23. táblázat. A DiagramPlotterApplet osztályhoz tartozó függvények

Az egér-kezelő függvények közül csak a kattintáshoz tartozó függvényt használjuk. Az init() függvényben beállítjuk az összes változó kezdőértékét és a paraméterekből kiolvassuk a megjelenítendő adatokat. A konkrét tartalmat a paint() függvényben rajzoltatjuk ki az applet-tel.

Az előbb bemutatott osztályban használt Alignment felsoroló típus forráskódját az Alignment.java fájlban találhatjuk:

public enum Alignment{LEFT, CENTER, RIGHT }

A fájl ezt az egyetlen sort tartalmazza, a Java fordító automatikusan megtalálja és beolvassa az őt használó osztály Bytecode-ra történő fordítása során.

3.3. A RENDSZER TESZTELÉSE

Információs rendszerünk két fő lépésben készült el:

- Először elkészítettük az adatbázist az eddig tárgyalt elemek nagy részével együtt. A fejlesztési modell lényegében egy alulról fölfele épülő piramis-modell volt;
- Utána elkészítettük a webes felhasználói felületet, és közben folyamatosan kiegészítettük az adatbázist a szükséges újabb elemekkel. A felület fejlesztési modellje egy felülről lefele épülő piramis modell volt.

A tesztelés folyamatosan történt az újabb és újabb szolgáltatások megvalósításakor. Egységteszteket hajtottunk végre, majd ezt követően mindig leteszteltük a már tesztelt egységekre épülő újabb egységeket is.

Függetlenül attól, hogy az éppen tesztelni kívánt elem az adatbázisban található-e vagy a felületen, a következő tesztelési módszerek érvényesültek:

1. Először mindenféle kivételkezelés nélkül teszteltük az adott egységeket (függvények, eljárások, táblázatok, nézetek, stb.) normális,

rendeltetésszerű adatokkal. Amikor arra jól működtek, akkor – továbbra is kivételkezelés nélkül – teszteltük az összes lehetséges szélsőséges esetre is az adott egységet. Hagytuk a programot elszállni, mert a keletkező hibaüzenetek nagy segítséget jelentettek a hibajavításban;

 Miután már minden általunk megadott adatra jól működött az egység, bekapcsoltuk a kivételkezelést és ráeresztettük az egységet az adatbázisban tárolt, nagy mennyiségű, véletlenszerű tesztadatra. Ezzel egyúttal az algoritmusok hatékonyságát is tesztelni tudtuk.

Miután a fent említett tesztelési terv alkalmazásával már minden tesztadatra elfogadható sebességgel és jól futott le az adott programrész, továbbmentünk a következő egységekre. A tesztelés egyre magasabb rétegeket érintett, így végül az egész rendszer összességében le lett tesztelve.

4. IRODALOMJEGYZÉK

A dokumentum és az általa tárgyalt rendszer elkészítéséhez kizárólag nyilvánosan elérhető, online segédanyagokat használtam. Az adatbázisban található, véletlenszerű tesztadatok alapjául szolgáló forrásadatokat a Kika Lakberendezési Kft. biztosította. A dokumentációban tárgyalt adatforrás (products.datasource), ún. rendelési segédlista adattáblájának felépítését szintén a Kika Lakberendezési Kft. határozta meg.

A dokumentumban tárgyalt adatbázisban, a termékek adataira vonatkozóan a következő adattáblák és nézetek elkészítése is az áruházzal egyeztetve történt:

- products.outgoing;
- products.not_outgoing;
- products.discounts;
- products.default_weights;
- products.weights;
- products.mpl shipping fees;
- products.default shipping fees;
- products.hdt products;
- products.hdt shipping fees;
- products.gls_products;

products.grando_productlist;

A megrendelési adatok a Grando webáruház eladói központjában letölthető megrendelések mintájára kerültek tárolásra az adatbázisban (illetve az exportálható lista is ugyanabban a formátumban került elkészítésre):

- orders.orderlist;
- orders.grando orderlist;

A letölthető forgalmi kimutatásokat generáló nézetek az áruház által kért paraméterek alapján lettek elkészítve:

- orders.incoming orders;
- orders.incoming_orders_detailed;
- orders.order monthly income;
- orders.order_monthly_income_detailed;
- orders.orders not canceled;
- orders.orders_not_canceled_detailed.

Ezen kívül, a webes felhasználói felület eladói központjában elérhető Javaapplet-ben szereplő diagramok is az áruház által kért formátumban jelennek meg.

4.1. ONLINE SEGÉDANYAGOK

PostgreSQL 8.4.14 Documentation
 <u>http://www.postgresql.org/docs/8.4/interactive/index.html</u>
 Copyright © 1996-2012 The PostgreSQL Global Development Group

130

- PHP Documentation
 <u>http://php.net/docs.php</u>
 Copyright © 2001-2012 The PHP Group. All rights reserved.
- JAVA Documentation <u>http://docs.oracle.com/javase/6/docs/</u> Copyright © 1993, 2011, Oracle and/or its affiliates. All rights reserved.
- CG Textures website (háttérképekhez) <u>http://www.cgtextures.com/</u> 2012.11.01.
- Icon Archive website (ikonokhoz) <u>http://www.iconarchive.com/</u> 2012.11.01.
- Find Icons (ikonokhoz) http://findicons.com/ 2012.11.01.