

JEGYZŐKÖNYV

Adatbázisrendszerek I.

Féléves feladat: Áruház-beszállító hálózat

Készítette: **Stremler László**
Neptunkód: **AQYO8L**
Gyakorlatvezető: **Dr. Bednarik László**

Miskolc, 2022

Tartalomjegyzék

1. A feladat leírása	3
2. Az adatbázis ER modellje	4
3. Az adatbázis konvertálása relációs modellre	5
4. Az adatbázis relációs modellje	6
5. Az adatbázis relációs sémája	6
6. Az adattáblák létrehozása	7
7. Az adattáblák feltöltése	8
8. Lekérdezések	10

1. A feladat leírása

Az adatbázisomban több áruház kapott szerepet, az árult termékek szempontjából. Bemutatásra kerülnek benne az áruház raktárának elemei, a különböző termékek ára, akciós ára, illetve külföldi pénznemi ára is. Továbbá megmutatkozik az áruház-beszállító kapcsolat, az áruház heti átlagos termékberendelésének száma, illetve a beszállító kapcsolata a raktárával. A beszállító raktárában -ugyanúgy ahogy az áruház raktáránál is- feltüntetésre kerülnek a különböző termékek, raktáron lévő darabszámuk, illetve az áruházi raktárban megtalálható termék új névvel kiadott változatai is.

Az ER-modell összesen 5 egyedet tartalmaz, melyek a következők:

- Áruház,
- Beszállító,
- Beszállító raktár,
- Áruház raktár tartalom,
- Akciós termékek

Először is az **Áruház** egyedet szeretném bemutatni. Ez az egyed tárolja az áruházak legfőbb tulajdonságait (név, cím, áruház azonosító). Ez az egyed az összeköttetés a beszállító és az áruházi raktár között. Elsődleges kulcs az AruhazID, amely minden áruház esetében egyedi és ez alapján lehet beazonosítani a raktárban, hogy az adott termék melyik áruház polcain található meg, illetve hogy az adott áruház mely beszállítóktól pótolja az áruit.

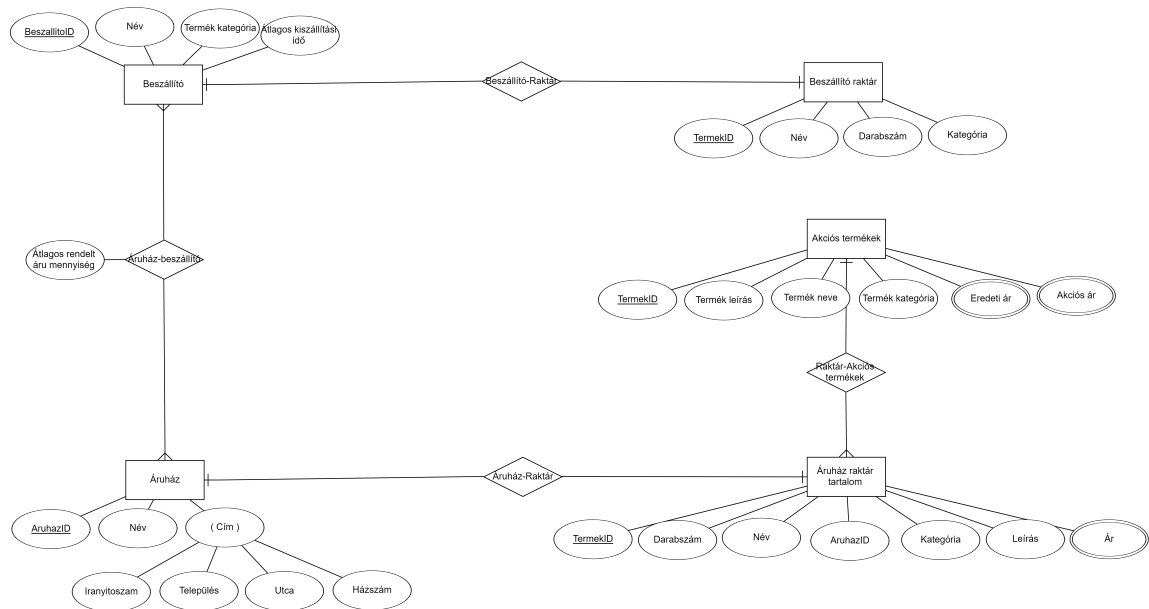
A következő fontos egyed a **Beszállító**. Ez az egyed tárolja a beszállítók adatait (Név, cím, azonosító, termék kategória, átlagos kiszállítási idő). Itt van meghatározva hogy mely beszállítók mely áruházaknak szállítanak. Mivel több áruház több beszállítótól is rendelhet árut és több beszállító is beszállíthat ugyanolyan kategóriájú árut az áruházba, ezért a beszállító és az áruház között **N:M kapcsolat** van. A kapcsolatot pedig az Áruház-beszállító kapcsolat jellemzi, amely összeköti az Áruház és Beszállító táblákat, valamint meghatározza az átlagos rendelt árumennyiséget.

Mivel a beszállító saját árukészlet nélkül nem ér sokat, ezért egy beszállító raktár egyedet is létrehoztam. Ebben a példában egy beszállító (mivel csak egy kategóriájú terméket szállít) egy raktárral rendelkezik és egy raktárhoz csak egy beszállító tartozik, ezért ez a kettő között **1:1 kapcsolat** található.

Az áruháznak is rendelkeznie kell saját raktárral, ahonnan feltölti az áruit és ahol tárolja a beérkezett árukat, így egy áruház raktár egyed is megtalálható a modellemben. Hasonlóképp, mint a beszállítónál, itt is egy áruház egy raktárral rendelkezik és egy raktár csak egy áruházat "szolgál ki", ezért a két egyed között **1:1 kapcsolat** jellemző.

Végül pedig létrehoztam egy akciós termékek egyedet is, mivel az áruházak csak úgy tudnak gördülékenyen működni, ha néha engednek az árból. Az akciós termékek egyed közvetlen kapcsolatban áll az áruház raktárával, ezáltal könnyen nyomkövethető mely termékek akciósak. A két egyed között **1:N kapcsolat** található, mivel egy raktárbeli elemnek csak egy akciós variánsa lehet, viszont egy akciós termék több áruház raktárában is megjelenhet.

2. Az adatbázis ER modellje



3. Az adatbázis konvertálása relációs modellre

Először az Áruház és a Beszállító táblákat hozzuk létre, mivel ez a kettő a legfontosabb a hálózat szempontjából. A kettő közötti **N:M kapcsolat** miatt egy külön táblát kell rendelni a kettő összekapcsolására, amely tartalmazni fogja az ÁruházID-t és a BeszállítóID-t, mindkettőt mint idegenkulcsot, valamint az átlagos berendelt áru mennyiséget, ami áruházanként egyedi tulajdonság. Mindkét idegenkulcs megkapta a NOT NULL integritási feltételt, hisz létező áruházat kell összekötnie, létező beszállítóval. Minden oszlopa ennek a táblának INT típusú. Amennyiben ez a tábla nem tartalmazza valamelyik kulcsot, úgy az adott intézmény nincs kapcsolatban a másikkal.

A beszállító tábla elsődleges kulcsa a BeszállítóID, amely egyedi és ez alapján kerül összeköttetésre a beszállító egy (vagy több) áruházzal. Ezenkívül tartalmazza még a beszállító nevét (VARCHAR), a Termék kategóriáját (VARCHAR) és az átlagos kiszállítási időt (INT). Idegenkulcsot ez a tábla nem tartalmaz.

A beszállító raktár tábla elsődleges kulcsa a TermékID (INT), amely a terméket azonosítja. Egy adott termékhez továbbá tartozik egy név (VARCHAR), egy kategória (VARCHAR), egy leírás (VARCHAR) és egy darabszám (INT). Utóbbi a raktáron elérhető darabszámmra utal. A tábla idegenkulcsa a BeszállítóID, ez alapján lehet tudni mely termékek mely beszállító tartománya alá tartozik.

Áruházi oldalról szemlélve a dolgokat, az Áruház tábla egy AruhazID elsődleges kulcsot tartalmaz, ez alapján kerül összeköttetésre egy áruház a beszállítóval. Továbbá megtalálhatóak a táblában az áruház jellemzésére szolgáló mezők. (Név, cím (mindkettő VARCHAR)). Ez a tábla idegenkulcsot nem tartalmaz.

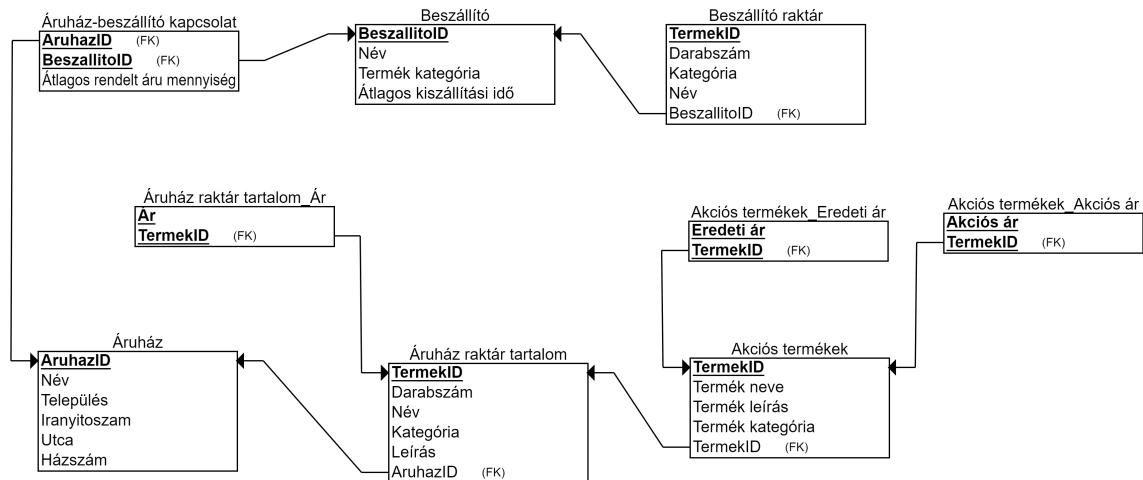
Ezt követően jön az Áruház_raktár_tartalom tábla, amely egy összesített raktárként üzemel, minden bolt, minden terméke ebben található meg. Elsődleges kulcsa a TermékID (INT), amellyel egy termék azonosítható. Ezenkívül a termék jellemzésére szolgáló Név (VARCHAR), Darabszám (INT), Leírás (VARCHAR), Kategória (VARCHAR) is helyet kapott a táblában. A tábla idegenkulcsa nem más, mint az AruhazID, amely arra hivatkozik, hogy az adott termék melyik áruház polcain található meg.

Az Áruház_raktár_tartalom tábla ár mezője a többértékűsége miatt külön táblában foglalja el a helyét. Ez a tábla 2 oszlopot tartalmaz, egy elsődlegeskulcsot, az Árat (INT) és egy elsődleges idegenkulcsot, a TermékID-t. Ilyen módon lett megoldva az, hogy egy terméknek több pénznembeli ára is lehessen.

Az Akciós_termékek kapcsolódik bele az áruházi raktár táblába, így ez a soron következő a jellemzés szempontjából. Ez a tábla tartalmazza azon termékeket, amelyek fellelhetőek az áruházban és akciós áruk van, ezáltal felülírva eredeti árukat. A tábla elsődleges kulcsa a TermékID, valamint ez az idegenkulcsa is, hisz ily módon tud kapcsolódni a raktár táblához. Ezenkívül a termék jellemzői is megtalálhatóak benne, úgymint a Termék neve (VARCHAR), Termék leírása (VARCHAR), Termék kategóriája (VARCHAR).

Mivel egy termék több árral is rendelkezhet, ezáltal az akciós változata is, így az akciós és az eredeti árát egy terméknek külön-külön táblában tároljuk. Ezeknek a tábláknak az oszlopai a termék eredeti, vagy akciós ára, valamint a TermékID, ami elsődleges és idegenkulcsként is funkcionál és mindkét tábla esetében ezen keresztül tudjuk visszavezetni az akciós termékek táblára a terméket.

4. Az adatbázis relációs modellje



5. Az adatbázis relációs sémája

Áruház[AruhazID, Név, Település, Iranyitoszam, Utca, Házsám, TermekID]

Áruház-beszállító kapcsolat[AruhazID, BeszallitoID, Átlagos rendelt áru mennyiség]

Beszállító[BeszallitoID, Név, Termék kategória, Átlagos kiszállítási idő]

Beszállító raktár[TermekID, Darabszám, Kategória, Név, BeszallitoID]

Áruház raktár tartalom[TermekID, Darabszám, Név, Kategória, Leírás, AruhazID]

Áruház raktár tartalom ár[Ár, TermekID]

Akciós termékek[TermekID, Termék neve, Termék leírás, Termék kategória, TermekID]

Akciós termékek_akciós_ár[Akciós ár, TermekID]

Akciós termékek_eredeti_ár[Eredeti ár, TermekID]

6. Az adattáblák létrehozása

Az adattáblák létrehozásánál számít a sorrend, mivel először azokat a táblákat kell létrehozni, ami tartalmazza azt a kulcsot, amire mutat egy másik tábla. Máskülönben hibát fogunk kapni. Az SQL kódokat a XAMPP program segítségével a phpMyAdmin felületen írtam.

A baloldali kép a táblák létrehozásának SQL kódját mutatja, a jobb oldali pedig a táblák lekérdezését a DESCRIBE paranccsal:

```
CREATE TABLE Aruhaz
(
    AruhazID INT NOT NULL,
    Név VARCHAR(45) NOT NULL,
    Település VARCHAR(45) NOT NULL,
    Iranyítószám INT NOT NULL,
    Utca VARCHAR(45) NOT NULL,
    Házszám INT NOT NULL,
    PRIMARY KEY (AruhazID)
);

CREATE TABLE Aruhaz_raktár_tartalom
(
    Darabszám INT NOT NULL,
    Név VARCHAR(45) NOT NULL,
    Kategória VARCHAR(45) NOT NULL,
    Leírás VARCHAR(145) NOT NULL,
    TermekID INT NOT NULL,
    AruhazID INT NOT NULL,
    PRIMARY KEY (TermekID),
    FOREIGN KEY (AruhazID) REFERENCES Aruhaz(AruhazID)
);

CREATE TABLE Akciók_termékek
(
    TermekID INT NOT NULL,
    Termek_neve VARCHAR(45) NOT NULL,
    Termek_leírás VARCHAR(145) NOT NULL,
    Termek_kategória VARCHAR(45) NOT NULL,
    PRIMARY KEY (TermekID),
    FOREIGN KEY (TermekID) REFERENCES Aruhaz_raktár_tartalom(TermekID)
);

CREATE TABLE Aruhaz_raktár_tartalom_ár
(
    Ár INT NOT NULL,
    TermekID INT NOT NULL,
    PRIMARY KEY (Ar, TermekID),
    FOREIGN KEY (TermekID) REFERENCES Aruhaz_raktár_tartalom(TermekID)
);

CREATE TABLE Akciók_termékek_eredeti_ár
(
    Eredeti_ár INT NOT NULL,
    TermekID INT NOT NULL,
    PRIMARY KEY (Eredeti_ár, TermekID),
    FOREIGN KEY (TermekID) REFERENCES Akciók_termékek(TermekID)
);

CREATE TABLE Akciók_termékek_Akció_ár
(
    Akció_ár INT NOT NULL,
    TermekID INT NOT NULL,
    PRIMARY KEY (Akció_ár, TermekID),
    FOREIGN KEY (TermekID) REFERENCES Akciók_termékek(TermekID)
);

CREATE TABLE Beszállító
(
    BeszállítóID INT NOT NULL,
    Név VARCHAR(45) NOT NULL,
    Termek_kategória VARCHAR(45) NOT NULL,
    Átlagos_kiszállítási_idő INT NOT NULL,
    PRIMARY KEY (BeszállítóID)
);

CREATE TABLE Beszállító_raktár
(
    TermekID INT NOT NULL,
    Darabszám INT NOT NULL,
    Kategória VARCHAR(45) NOT NULL,
    Név VARCHAR(45) NOT NULL,
    BeszállítóID INT NOT NULL,
    PRIMARY KEY (TermekID),
    FOREIGN KEY (BeszállítóID) REFERENCES Beszállító(BeszállítóID)
);

CREATE TABLE Aruhaz_beszállító_kapcsolat
(
    Átlagos_rendelt_ár_mennyiség INT NOT NULL,
    AruhazID INT NOT NULL,
    BeszállítóID INT NOT NULL,
    PRIMARY KEY (AruhazID, BeszállítóID),
    FOREIGN KEY (AruhazID) REFERENCES Aruhaz(AruhazID),
    FOREIGN KEY (BeszállítóID) REFERENCES Beszállító(BeszállítóID)
);
```

Az SQL-lekérdezés végrehajtása sikertelen					
describe Aruhaz;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
AruhazID	int(11)	NO	PRI	NULL	
Név	varchar(45)	NO		NULL	
Település	varchar(45)	NO		NULL	
Iranyítószám	int(11)	NO		NULL	
Utca	varchar(45)	NO		NULL	
Házszám	int(11)	NO		NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe beszállító;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
BeszállítóID	int(11)	NO	PRI	NULL	
Név	varchar(45)	NO		NULL	
Termek_kategória	varchar(45)	NO		NULL	
Átlagos_kiszállítási_idő	int(11)	NO		NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe beszállító_raktár;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
TermekID	int(11)	NO	PRI	NULL	
Darabszám	int(11)	NO		NULL	
Kategória	varchar(45)	NO		NULL	
Név	varchar(45)	NO		NULL	
BeszállítóID	int(11)	NO	MUL	NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe Aruhaz_raktár_tartalom_ár;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
Átlagos_rendelt_ár_mennyiség	int(11)	NO		NULL	
AruhazID	int(11)	NO	PRI	NULL	
BeszállítóID	int(11)	NO	PRI	NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe Aruhaz_raktár_tartalom;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
Darabszám	int(11)	NO		NULL	
Név	varchar(45)	NO		NULL	
Kategória	varchar(45)	NO		NULL	
Leírás	varchar(145)	NO		NULL	
TermekID	int(11)	NO	PRI	NULL	
AruhazID	int(11)	NO	MUL	NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe Aruhaz_raktár_tartalom_ár;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
Ar	int(11)	NO	PRI	NULL	
TermekID	int(11)	NO	PRI	NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe akciók_termékek;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
TermekID	int(11)	NO	PRI	NULL	
Termek_neve	varchar(45)	NO		NULL	
Termek_leírás	varchar(145)	NO		NULL	
Termek_kategória	varchar(45)	NO		NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe akciók_termékek_Akció_ár;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					
Field	Type	Null	Key	Default	Extra
Eredeti_ár	int(11)	NO	PRI	NULL	
TermekID	int(11)	NO	PRI	NULL	
Az SQL-lekérdezés végrehajtása sikertelen					
describe akciók_termékek_Akció_ár;					
[Szorossági helyben] [Művelet] [PHP-aid Webhözza]					
Extra options					

7. Az adattáblák feltöltése

A feltöltésnél ügyelni kell a helyes sorrendre és arra, hogy megfelelő változótípust használjunk, különben súlyos hibákat kaphatunk. Az első kép a tábla feltöltések SQL kódját mutatja, míg a második kép a a táblák tartalmát.

```
INSERT INTO `áruház` (`AruhazID`, `Név`, `Település`, `Iranyitoszam`, `Utca`, `Házszám`) VALUES ('1', 'Coop', 'Mucsomröcsöge', '1234', 'Bánat utca', '21'), ('2', 'Spar', 'Bekecs', '6789', 'Károly utca', '98'), ('3', 'Penny Market', 'Andalujza', '9183', 'Petőfi utca', '56'), ('4', 'Marika kisboltja', 'Sajószöged', '6756', 'Korda György utca', '69');

INSERT INTO `beszállító` (`BeszallitoID`, `Név`, `Termék_kategória`, `Átlagos_kiszállítási_idő`) VALUES ('1', 'Beszál-Litok', 'Gumiabroncs', '120'), ('2', 'Hűtő feltöltők', 'Fagyasztott áru', '45'), ('3', 'KertÉSZkedés', 'Kertészet', '60'), ('4', 'Rip-Rop', 'Rágcsa', '70');

INSERT INTO `áruház_beszállító_kapcsolat` (`AruhazID`, `BeszallitoID`) VALUES ('1', '2'), ('1', '4'), ('4', '4'), ('4', '2'), ('3', '2'), ('3', '1'), ('3', '3'), ('3', '4'), ('2', '4'), ('2', '3'), ('2', '2'), ('2', '1');

INSERT INTO `áruház_raktár_tartalom` (`Darabszám`, `Név`, `Kategória`, `Leírás`, `TermekID`, `AruhazID`) VALUES ('20', 'Mogyoróvaj', 'Rágcsa', 'Finomság :)', '1', '2'), ('10', 'Kecsketej', 'Fagyasztott áru', 'Finom, frissen fejt kecsketejcsike', '2', '4'), ('200', 'Kinder MaxiKing', 'Rágcsa', 'A létező legjobb édesség', '3', '3'), ('11', 'Kerekerdő Gumiabroncs', 'Jármű', 'Jó kis abroncs', '4', '2'), ('5', 'Tujá', 'Kertészet', 'Szimpla tujá', '5', '2'), ('6', 'Gyorsfagyasztott pizza', 'Fagyasztott áru', 'Egy gyorsan megfagyasztott pizza', '6', '1');

INSERT INTO `áruház_raktár_tartalom_ár` (`Ár`, `TermekID`) VALUES ('200', '3'), ('500', '1'), ('300', '2'), ('15000', '4'), ('600', '5'), ('1200', '6');

INSERT INTO `akciós_termékek` (`TermekID`, `Termék_neve`, `Termék_leírás`, `Termék_kategória`) VALUES ('1', 'Akciós mogyoróvaj', 'Finom, krémes és még akciós is!', 'Rágcsa'), ('4', 'Kerekerdő Gumiabroncs', 'Jól megy és még akciós is!', 'Jármű');

INSERT INTO `akciós_termékek_eredeti_ár` (`Eredeti_ár`, `TermekID`) VALUES ('600', '1'), ('15000', '4');

INSERT INTO `akciós_termékek_akciós_ár` (`Akciós_ár`, `TermekID`) VALUES ('450', '1'), ('10000', '4');

INSERT INTO `beszállító_raktár` (`TermekID`, `Darabszám`, `Kategória`, `Név`, `BeszallitoID`) VALUES ('1', '34', 'Rágcsa', 'Mogyoróvaj', '4'), ('2', '10', 'Fagyasztott áru', 'Birkatej', '2'), ('6', '200', 'Fagyasztott áru', 'Gyorsfagyasztott ananászos pizza', '2');
```


9

8. Lekérdezések

1. Kérdezzük le az összes olyan beszállított, aki fogyasztott árut szállított
 SELECT * FROM beszállító WHERE Termék_kategória="Fogyasztott áru"
 0 termék_megneve = "Fogyasztott áru" beszállító

2. Kérdezzük le az összes olyan áru nevét a beszállító raktárból, amit ha bármelyik áruház berendelne ugyanolyan névvel kapna meg!
 SELECT b.Név FROM beszállító_raktár b JOIN áruház_raktár_tartalom a ON b.TermekID=a.TermekID
 WHERE b.Név LIKE a.Név
 IT a_név
 0 a_név LIKE a_név
 (P a_beszallito_raktar_M b_termekid = a_termekid
 P a_beszallito_raktar_tartalom)

3. Kérdezzük le az összes olyan akciók termék nevét, amit be tud rendelni az áruház a beszállítótól készletlátny esetén! (nem számít hogy az áruház kapcsolatban ill a beszállítóval)
 SELECT a.Termék_neve FROM akciók_termékek a INNER JOIN beszállító_raktár b ON a.TermekID=b.TermekID
 IT a_termek_neve
 (P a_akciok_termek_M a_termekid = b_termekid
 P a_beszallito_raktar)

4. Kérdezzük le az összes olyan termék nevét, amit be tud rendelni az áruház a beszállítótól készletlátny esetén! (nem számít hogy az áruház kapcsolatban ill a beszállítóval)
 SELECT a.Termék_neve FROM akciók_termékek a INNER JOIN beszállító_raktár b ON a.TermekID=b.TermekID LEFT JOIN áruház_raktár_tartalom arak ON a.TermekID=arak.TermekID JOIN áruház ar ON ar.AruhaID=ar.AruhaID JOIN áruház_beszallító_kapcsolat abk ON ar.AruhaID=abk.BeszallitoID GROUP BY a.Termék_neve;
 IT a_termek_neve
 (P a_akciok_termek_M a_termekid = b_termekid
 P a_beszallito_raktar_M a_termekid = arak_termekid
 P arak_aruhas_raktar_tartalom_M arak_aruhasid = ar_aruhasid
 P ar_aruhasid_M ar_aruhasid = abk_beszallito
 P abk_aruhasid_beszallito_kapcsolat)

5. Kérdezzük le a Coop áruházi raktárában lévő termékek átlagárát!
 SELECT AVG(arta.Ar) AS 'Átlagár' FROM áruház_raktár_tartalom a JOIN áruház_raktár_tartalom ar_arta ON a.TermekID=arta.TermekID JOIN áruház ar ON a.AruhaID=ar.AruhaID WHERE ar.Név="Coop"
 IT AVG(ar)
 Y AVG(ar)
 0 ar_név = "Coop"
 (P ar_aruhas_raktar_tartalom_M a_termekid = arta_termekid
 P arta_aruhas_raktar_tartalom_a_M a_aruhasid = ar_aruhasid
 P ar_aruhasid)

6. Kérdezzük le az akciók termékek nevét és árának különbségét a normál ártól!
 SELECT art.Név, (arta.Ar-ataa.AkcioAr) AS 'Árkülönbség' FROM akciók_termékek akciók_ar_arta JOIN áruház_raktár_tartalom ar_arta ON ataa.Ataa.TermekID=akciók_ar_arta.TermekID JOIN áruház_raktár_tartalom art ON ataa.TermekID=art.TermekID
 IT art_arta_arta_arta_akcio_ar_arta_akcio_ar
 (P ataa_akciok_termek_akcio_ar_M ataa_termekid = arta_termekid
 P arta_aruhas_raktar_tartalom_a_M ataa_termekid = art_termekid
 P art_aruhas_raktar_tartalom)

7. Kérdezzük le az akciók termékek nevét és árának különbségét az akcióban kiírt normál ártól!
 SELECT art.Név, (area.Eredeti_ar-ataa.AkcioAr) AS 'Árkülönbség' FROM akciók_termékek akciók_ar_arta JOIN áruház_raktár_tartalom ar_arta ON area.Ataa.TermekID=area.TermekID JOIN áruház_raktár_tartalom art ON ataa.TermekID=art.TermekID
 IT art_arta_arta_akcio_ar_arta_akcio_ar
 (P ataa_akciok_termek_akcio_ar_M ataa_termekid = area_termekid
 P area_akciok_termek_akcio_ar_M ataa_termekid = art_termekid
 P art_aruhas_raktar_tartalom)

8. Hány darab különböző 300 forint feletti termék kapható a Sparban?
 SELECT COUNT(art.TermekID) AS 'Darabszám' FROM áruház_raktár_tartalom art JOIN áruház_raktár_tartalom ar_arta ON art.TermekID=arta.TermekID JOIN áruház a ON art.AruhaID=a.AruhaID WHERE arta.Ar>300 AND a.Név LIKE 'Spar'
 IT COUNT(Darabszam)
 Y COUNT(Darabszam)
 0 arta_ar = 300 AND a_név LIKE "Spar"
 (P art_aruhas_raktar_tartalom_M art_termekid = arta_termekid
 P arta_aruhas_raktar_tartalom_a_M arta_aruhasid = a_aruhasid
 P a_aruhasid)

9. Mely termékeket szállítják le a legrovidebb idő alatt a beszállítók?
 SELECT b.Átlagos_kiszallitasi_ido AS 'Idő', br.Név AS 'Leggyorsabban leszállítható termékek' FROM beszállító_raktár br JOIN beszállító b ON br.BeszallitoID=b.BeszallitoID ORDER BY b.Átlagos_kiszallitasi_ido ASC LIMIT 1
 IT atlagos_kiszallitasi_ido Y atlagos_kiszallitasi_ido 0 b_beszallito_M b_beszallitoID b_beszallitoID b_atlagos_kiszallitasi_ido b_beszallito

10. Melyik áruház rendel a legtöbb terméket a beszállítóktól?
 SELECT a.Név FROM áruház a JOIN áruház_beszallító_kapcsolat abk ON abk.AruhaID=a.AruhaID HAVING MAX(abk.Átlagos_rendelt_aru_mennyiseg);
 IT max Y max 0 abk_aruhas_beszallito_kapcsolat_M abk_aruhasid_AruhaID b_atlagos_rendelt_aru_mennyiseg abk_aruhasid

1

2

3

4

5

6

7

8

9

10