

JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves feladat

Csomagkövető rendszer

Készítette: Gyáni Kevin Zsolt

Neptunkód: CBOYZF

Dátum: 2022.11.19

Tartalomjegyzék:

1 a Az adatbázis ER-modellje.....	4
1 b Az ER-modell relációs modellre történő konvertálása	5
1 c Az adatbázis relációs sémája.....	6
1 d Táblák létrehozása.....	7
1 e Az adattáblák feltöltése	13
1 f Lekérdezések (SQL és relációs algebra).....	16

A beadandó feladatom során, egy számítás technikai eszközök eladásával foglalkozó internetes áruház csomagkövető rendszerét igyekeztem lemodellezni. Az áruház egyebek mellet teszteli is a számítás technikai eszközöket, szamon tartja a gyárihibás darabokat, illetve több raktárral és futárszolgálatl is kapcsolatban áll. A fent említett adatok nyilvántartása érdekében 5 egyedet hoztam létre melyek a következők:

- Beszállító
- Raktárak
- Gyárihibás
- Rendelés
- Ügyfél

Először is érdemes pár szót ejteni a **beszállító** egyedről, ez a kiinduló pontja a teljes adatbázisnak. Ez az egyed tárolja a különböző beszállítókat ID (ezek lesznek az egyedi kulcsok) szerint, illetve információkat biztosít még a csomagolás típusáról, valamint a csomag pontos áráról is. Továbbá tárolja még a várható érkezéssel kapcsolat információkat, így a vevő pontos képet kaphat arról, hogy mikorra várható a csomagja. A beszállító és a raktár között több-több kapcsolat van ugyanis 1 beszállító cég több raktárba is szállíthat, és 1 raktárba több beszállító cég áruja is érkezhet.

A beszállítótól a **raktárak** nevezetű egyedbe érkeznek a csomagok, mivel az áruház nemzetközi szinten is forgalmaz termékeket így több raktára is van, amelyek más-más helyszínen helyezkednek el, éppen ezért ez a tábla tárolja a raktárba érkezett termék címét, valamint ID-ját (ezek lesznek az egyedi kulcsok). Ezek mellet információkat biztosít még a termék áráról, valamint raktárba érkezés pontos dátumáról, így számon lehet tartani mennyire volt pontos a beszállítótól kapott várható érkezés. A raktár és a gyári hibás termék között 1-1 kapcsolatot létesítettem ugyanis csak 1 egyedi ID-vall rendelkező termék lehetséges 1 raktárban.

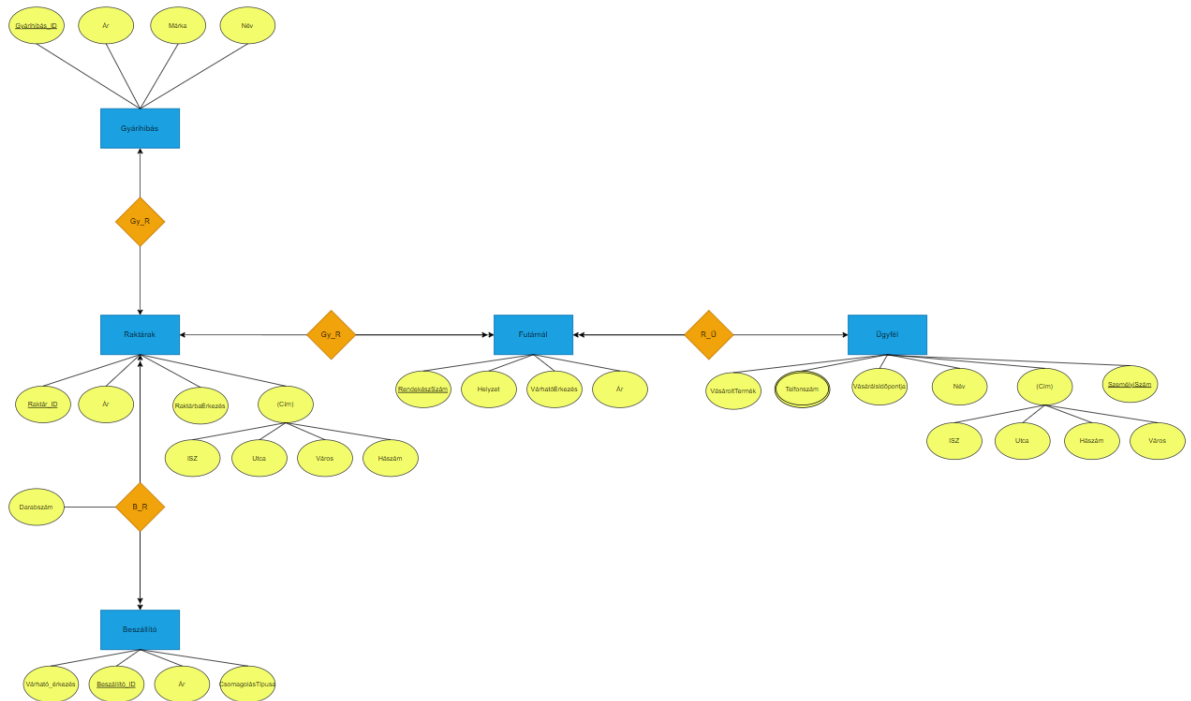
A webshop által forgalmazott termékek lehetnek **gyári hibásak** is. Éppen ezért ezeket a termékeket a raktárban ellenőrzik és amennyiben valamilyen hiba lép fel a termék tesztelése után azokat a beszállítón keresztül visszaküldik a forgalmazóhoz. Ebben a táblában a rendszerezés érdekében szükséges letárolni a termék ID-ját (ezek lesznek az egyedi kulcsok), illetve árát, márkáját és nevét a könnyeb azonosítás érdekében.

Amennyiben a termék működése helyénvalónak bizonyul a raktárból a terméket feladják a kért **rendelésre**. Ebben az egyedben vannak tárolva az ezzel kapcsolatos információk, például a rendelés száma (ezek lesznek az egyedi kulcsok), a csomagkövetés érdekében a rendelés helyzete, illetve várható érkezése, valamint a fizetendő összeg, mivel ezek az ügyfél számára mind lényeges információk. A rendelés és ügyfél között több-egy kapcsolatot létesítettem mivel 1 rendelés csak 1 ügyfélhez tartozhat, de egy ügyfélnek lehet több különböző rendelése is.

Végül pedig érdemes még beszélni az **ügyfél** tábláról melyben a csomagok rendelőjének beazonosításának érdekében szükséges adatok tároljuk le. A vásárolt termék nevét, valamint a vásárlás időpontját. A pontosság és dupilációk elkerülésének érdekében bekérjük még az ügyfél személyi igazolványának számát (ezek lesznek az egyedi kulcsok), valamint a kapcsolattartás érdekében a telefonszámát és a kiszállításhoz szükséges lakcímet is.

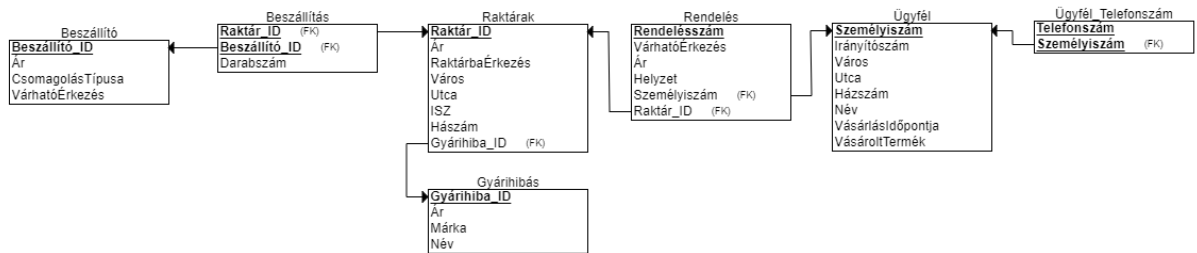
1 a Az adatbázis ER-modellje

Balról jobbra, valamint fentről lefelé haladva a feladat leírás sorrendjében építettem fel a táblákat és a közöttük lévő kapcsolati rendszert. A beszállító és raktárak Egyedek közötti kapcsolati tulajdonságként létrehoztam egy darabszámot. A címeket összetett tulajdonságként tartottam nyilván mivel a kiszállításhoz pontos cím szükséges, a telefonszámot több értékű tulajdonságként tartottam nyilván mivel egy ügyfélnek lehet céges, illetve magán telefonszáma is.



1 b Az ER-modell relációs modellre történő konvertálása

Már megszokott módon, balról jobbra haladva rendeztem el a táblákat, az elsődleges kulcsokat beállítottam az ID-kgnak megfelelően. A beszállító és raktárak Egyedek között létrehoztam még egy beszállítás táblát, amely a 2 egyed közötti kapcsolat tulajdonságát tárolja, az az a darabszámot. A raktárak egyedből továbbá származik még a rendelés, illetve a gyárihibás tábla. A rendelés táblában több idegen kulcs is van az ügyfél, illetve a raktárak táblák összekötésének érdekében. Az ügyfélhez tartozik még egy Ügyfél Telefonszám tábla is mely a többértékkal rendelkező telefonszám tulajdonságot tartja számon.



1 c Az adatbázis relációs sémája

Beszállító [Beszállító_ID, Ár, CsomagolásTípusa, VárhatóÉrkezés]

Beszállítás [Raktár_ID, Beszállító_ID, Darabszám]

Raktárak [Raktár_ID, Ár, RaktárbaÉrkezés, Város, Utca, Házszám, Gyárihiba_ID]

Gyárihibás [Gyárihiba_ID, Ár, Márka, Név]

Rendelés [Rendelészszám, VárhatóÉrkezés, Ár, Helyzet, Név, Raktár_ID]

Ügyfél [Személyszám, név, irányítószám, Város, Utca, Házszám, VásárlásIdőpontja, VásároltTermék]

Ügyfél_Telefonszám [Telefonszám, Név]

1 d Táblák létrehozása

Az adattáblák létrehozásához create használtam. Az adatok megadásakor ügyelni kell a helyes sorrendre, illetve a megadott formátumok betartására. Az SQL kódot MYSQL workbench, illetve XAMPP segítségével hoztam létre.

```
CREATE TABLE Beszállító
(
    Beszállító_ID INT NOT NULL,
    Ár INT NOT NULL,
    CsomagolásTípusa INT NOT NULL,
    VárhatóÉrkezés date NOT NULL,
    PRIMARY KEY (Beszállító_ID)
);

CREATE TABLE Gyárihiba
(
    Gyárihiba_ID INT NOT NULL,
    Ár INT NOT NULL,
    Márka varchar(50) NOT NULL,
    Név varchar(50) NOT NULL,
    PRIMARY KEY (Gyárihiba_ID)
);

CREATE TABLE Ügyfél
(
    Irányítószám INT NOT NULL,
    Város varchar(50) NOT NULL,
    Utca varchar(50) NOT NULL,
    Házszám INT NOT NULL,
    Személyszám varchar(50) NOT NULL,
    Név varchar(50) NOT NULL,
    VásárlásIdőpontja date NOT NULL,
    VásároltTermék varchar(50) NOT NULL,

    PRIMARY KEY (Személyszám)
);
```

```
CREATE TABLE Ügyfél_Telefonszám
(
    Telefonszám varchar(50) NOT NULL,
    Személyiszám varchar(50) NOT NULL,
    PRIMARY KEY (Telefonszám, Személyiszám),
    FOREIGN KEY (Személyiszám) REFERENCES Ügyfél(Személyiszám)
);

CREATE TABLE Raktárak
(
    Raktár_ID INT NOT NULL,
    Ár INT NOT NULL,
    RaktárbaÉrkezés date NOT NULL,
    Város varchar(50) NOT NULL,
    Utca varchar(50) NOT NULL,
    ISZ varchar(50) NOT NULL,
    Hászám INT NOT NULL,
    Gyárihiba_ID INT NOT NULL,
    PRIMARY KEY (Raktár_ID),
    FOREIGN KEY (Gyárihiba_ID) REFERENCES Gyárihibás(Gyárihiba_ID)
);
```


CREATE TABLE Rendelés

```
(  
    Rendelésszám INT NOT NULL,  
    VárhatóÉrkezés date NOT NULL,  
    Ár INT NOT NULL,  
    Helyzet varchar(50) NOT NULL,  
    Személyszám varchar(50) NOT NULL,  
    Raktár_ID INT NOT NULL,  
    PRIMARY KEY (Rendelésszám),  
    FOREIGN KEY (Személyszám) REFERENCES Ügyfél(Személyszám),  
    FOREIGN KEY (Raktár_ID) REFERENCES Raktárak(Raktár_ID)  
);
```

CREATE TABLE Beszállítás

```
(  
    Darabszám INT NOT NULL,  
    Raktár_ID INT NOT NULL,  
    Beszállító_ID INT NOT NULL,  
    PRIMARY KEY (Raktár_ID, Beszállító_ID),  
    FOREIGN KEY (Raktár_ID) REFERENCES Raktárak(Raktár_ID),  
    FOREIGN KEY (Beszállító_ID) REFERENCES Beszállító(Beszállító_ID)  
);
```

4

3 • desc Beszállító;

Result Grid						
		Filter Rows:			Export:	Wr
	Field	Type	Null	Key	Default	Extra
►	Beszállító_ID	int(11)	NO	PRI	NULL	
	Ár	int(11)	NO		NULL	
	CsomagolásTípusa	int(11)	NO		NULL	
	VárhatóÉrkezés	date	NO		NULL	

10 • desc Gyárihiba;

Result Grid						
		Filter Rows:				
				Export:		
	Field	Type	Null	Key	Default	Extra
►	Gyárihiba_ID	int(11)	NO	PRI	NULL	
	Ár	int(11)	NO		NULL	
	Márk-	varchar(50)	NO		NULL	
	Név	varchar(50)	NO		NULL	

15 • desc Ügyfél

Result Grid						
		Filter Rows:				
				Export:		
	Field	Type	Null	Key	Default	Extra
►	Irányítószám	int(11)	NO		NULL	
	Város	varchar(50)	NO		NULL	
	Utca	varchar(50)	NO		NULL	
	Házzszám	int(11)	NO		NULL	
	Személyszám	int(11)	NO	PRI	NULL	
	Név	varchar(50)	NO		NULL	
	VásárlásIdőpontja	date	NO		NULL	
	VásároltTermék	varchar(50)	NO		NULL	

o

9 • desc Ügyfél

Result Grid						
		Filter Rows:				
				Export:		
	Field	Type	Null	Key	Default	Extra
►	Irányítószám	int(11)	NO		NULL	
	Város	varchar(50)	NO		NULL	
	Utca	varchar(50)	NO		NULL	
	Házzszám	int(11)	NO		NULL	
	Személyszám	varchar(50)	NO	PRI	NULL	
	Név	varchar(50)	NO		NULL	
	VásárlásIdőpontja	date	NO		NULL	
	VásároltTermék	varchar(50)	NO		NULL	

3 • desc Ügyfél_Telefonszám;

Result Grid						
		Filter Rows:				
				Export:	Wrap	
	Field	Type	Null	Key	Default	Extra
►	Telefonszám	varchar(50)	NO	PRI	NULL	
	Személyszám	varchar(50)	NO	PRI	NULL	

14 • desc Raktárak;

Result Grid						
		Filter Rows:				
				Export:	Wrap Cell Content	
	Field	Type	Null	Key	Default	Extra
►	Raktár_ID	int(11)	NO	PRI	NULL	
	Ár	int(11)	NO		NULL	
	RaktárbaÉrkezés	date	NO		NULL	
	Város	varchar(50)	NO		NULL	
	Utca	varchar(50)	NO		NULL	
	ISZ	varchar(50)	NO		NULL	
	Hászám	int(11)	NO		NULL	
	Gyárihiba_ID	int(11)	NO	MUL	NULL	

13 • desc Rendelés;

Result Grid						
		Filter Rows:				
				Export:	Wrap C	
	Field	Type	Null	Key	Default	Extra
►	Rendelészám	int(11)	NO	PRI	NULL	
	VárhatóÉrkezés	date	NO		NULL	
	Ár	int(11)	NO		NULL	
	Helyzet	varchar(50)	NO		NULL	
	Személyszám	varchar(50)	NO	MUL	NULL	
	Raktár_ID	int(11)	NO	MUL	NULL	

10 • desc Beszállítás;

Result Grid						
		Filter Rows:			Export:	
	Field	Type	Null	Key	Default	Extra
►	Darabszám	int(11)	NO		NULL	
	Raktár_ID	int(11)	NO	PRI	NULL	
	Beszállító ID	int(11)	NO	PRI	NULL	

1 e Az adattáblák feltöltése

A feltöltésnél, csak úgy, mint a létrehozásnál, ügyelni kell a helyes sorrendre, illetve az adattípusra. A feltöltéshez insert to utasítást használtam.

```
INSERT INTO Beszállító values (01,500,2,'2022-11-21');
INSERT INTO Beszállító values (02,1500,1,'2022-11-22');
INSERT INTO Beszállító values (03,2000,3,'2022-11-20');
INSERT INTO Beszállító values (04,3250,2,'2022-11-10');
INSERT INTO Beszállító values (05,4500,1,'2022-11-25');

INSERT INTO Gyárihibás values (01,5000,'Asus','PRIME H610M-D D4');
INSERT INTO Gyárihibás values (02,16000,'Asus','PRIME H410M');
INSERT INTO Gyárihibás values (03,19000,'Acer','Nitro');
INSERT INTO Gyárihibás values (04,45000,'Razer','Kraken 2');
INSERT INTO Gyárihibás values (05,380000,'Apple','Iphone 14');

INSERT INTO Raktárak values(01,5000,'2022-10-11','Abaújszántó','Béke út',1,3881,01);
INSERT INTO Raktárak values(02,15000,'2022-10-19','Abaújszántó','Béke út',1,3881,02);
INSERT INTO Raktárak values(03,18000,'2022-11-12','Miskolc','Kazinczy út',3500,50,03);
INSERT INTO Raktárak values(04,44000,'2022-09-12','Abaújszántó','Béke út',1,3881,04);
INSERT INTO Raktárak values(05,390000,'2022-08-13','Miskolc','Kazinczy út',3500,50,05);

INSERT INTO Ügyfél values (3881,'Abaújszántó','Béke út',38,'1928374IE','Kovács Zsolt','2022-10-18','Iphone 12');
INSERT INTO Ügyfél values (3500,'Miskolc','Kazinczy út',51,'1972351KE','Tóth János','2022-10-22','Samsung Galaxy A53');
INSERT INTO Ügyfél values (3907,'Tállya','Szabadság út',21,'1826951HJ','Nagy Levente','2022-11-01','Samsung TV');
INSERT INTO Ügyfél values (3907,'Rátka','Kossuth út',71,'1831928KI','Csanád Gergő','2022-09-01','Playstation 4');
INSERT INTO Ügyfél values (3906,'Golop','József Attila út',11,'1944882BE','Kiss Lajos','2022-11-24','Xbox Series X');

INSERT INTO Ügyfél_Telefonszám values('06-30-395-1849','1831928KI');
INSERT INTO Ügyfél_Telefonszám values('06-70-213-1994','1944882BE');
INSERT INTO Ügyfél_Telefonszám values('06-20-564-2022','1928374IE');
INSERT INTO Ügyfél_Telefonszám values('06-30-723-1643','1826951HJ');
INSERT INTO Ügyfél_Telefonszám values('06-70-987-9986','1972351KE');

INSERT INTO Rendelés values(01,'2022-10-18',5000,'Depoban','1928374IE',01);
INSERT INTO Rendelés values(02,'2022-10-22',16000,'Depoban','1972351KE',02);
INSERT INTO Rendelés values(03,'2022-11-01',19000,'Futárnál','1826951HJ',03);
INSERT INTO Rendelés values(04,'2022-09-01',45000,'Futárnál','1831928KI',04);
INSERT INTO Rendelés values(05,'2022-11-24',380000,'Kiszállításra Átadva','1944882BE',05);

INSERT INTO Beszállítás values(01,01,01);
INSERT INTO Beszállítás values(10,02,02);
INSERT INTO Beszállítás values(5,03,03);
INSERT INTO Beszállítás values(6,04,04);
INSERT INTO Beszállítás values(4,05,05);
```

7 • `select * from Beszállító;`

Result Grid	Filter Rows:	Edit:	Export/Import
Beszállító_ID	Ár	CsomagolásTípusa	VárhatóÉrkezés
1	500	2	2022-11-21
2	1500	1	2022-11-22
3	2000	3	2022-11-20
4	3250	2	2022-11-10
5	4500	1	2022-11-25
NULL	NULL	NULL	NULL

7 • `select * from Gyárihiba;`

Result Grid

Filter Rows:

Edit:

	Gyárihiba_ID	Ár	Márka	Név
▶	1	5000	Asus	PRIME H610M-D D4
	2	16000	Asus	PRIME H410M
	3	19000	Acer	Nitro
	4	45000	Razer	Kraken 2
	5	380000	Apple	Iphone 14
★	NULL	NULL	NULL	NULL

```
7 • select * from Raktárák;
```

[illegible][illegible]

```
7 • select * from Ügyfél;
```

[illegible]

7 • `select * from Ügyfél_Telefonszám;`

Result Grid		
	Telefonszám	Személyszám
▶	06-20-564-2022	1928374IE
	06-30-395-1849	1831928KI
	06-30-723-1643	1826951HJ
	06-70-213-1994	1944882BE
	06-70-987-9986	1972351KE
*	NULL	NULL

	Rendelészám	VárhatóÉrkezés	Ár	Helyzet	Személyszám	Raktár_ID
▶	1	2022-10-18	5000	Depoban	1928374IE	1
	2	2022-10-22	16000	Depoban	1972351KE	2
	3	2022-11-01	19000	Futárnál	1826951HJ	3
	4	2022-09-01	45000	Futárnál	1831928KI	4
	5	2022-11-24	380000	Kiszállításra Átadva	1944882BE	5
*	NULL	NULL	NULL	NULL	NULL	NULL

	Darabszám	Raktár_ID	Beszállító_ID
▶	1	1	1
	10	2	2
	5	3	3
	6	4	4
	4	5	5
*	NULL	NULL	NULL

1 f Lekérdezések (SQL és relációs algebra)

1. Listázzuk ki azokat a gyári hibás termékeket, amelyeket az asus gyártott!

`select * from gyárihibás where márka='Asus' GROUP BY Gyárihiba_ID;`

```

 $\pi_{gyárihiba\_id}$ 
 $\gamma_{gyárihiba\_id,}$ 
 $\sigma_{márka = "Asus"} gyárihibás$ 

```


	Gyárihiba_ID	Ár	Márka	Név
▶	1	5000	Asus	PRIME H610M-D D4
	2	16000	Asus	PRIME H410M
*	NULL	NULL	NULL	NULL

2.Kérdezzük le az összes olyan gyári hibás terméket amelynek az ára ugyan annyi mint egy másik raktárban lévő termék ára!

select * from gyárihibás

INNER JOIN Raktarak ON gyárihibás.Gyárihiba_ID = Raktarak.Gyárihiba_ID

where gyárihibás.ár=Raktarak.ár;

$\sigma_{\text{gyárihibás.ár} = \text{raktarak.ár}} (\text{gyárihibás} \bowtie_{\text{gyárihibás.gyárihiba_id} = \text{raktarak.gyárihiba_id}} \text{raktarak})$

	Gyárihiba_ID	Ár	Márka	Név	Raktár_ID	Ár	RaktárbaÉrkezés	Város	Utca	ISZ	Hátszám	Gyárihiba_ID
▶	1	5000	Asus	PRIME H610M-D D4	1	5000	2022-10-11	Abaújszántó	Béke út	1	3881	1

3.Listázzuk azoknak az ügyfeleknek az adatait akiknek 06-30-as telefonszáma van.

select * from ügyfél

JOIN ügyfél_telefonszám on ügyfél.Személyszám=ügyfél_telefonszám.Személyszám

where Telefonszám LIKE '06-30%';

$\sigma_{\text{telefonszám LIKE "06-30\%"}} (\text{ügyfél} \bowtie_{\text{ügyfél.személyszám} = \text{ügyfél_telefonszám.személyszám}} \text{ügyfél_telefonszám})$

	Irányítószám	Város	Utca	Hátszám	Személyszám	Név	VásárlásiDőpontja	VásároltTermék	Telefonszám	Személyszám
▶	3907	Rátka	Kossuth út	71	1831928KI	Csanád Gergő	2022-09-01	Playstation 4	06-30-395-1849	1831928KI
	3907	Tállya	Szabadság út	21	1826951HJ	Nagy Levente	2022-11-01	Samsung TV	06-30-723-1643	1826951HJ

4. Listázzuk ki mennyi olyan rendelés van amely esetén a darabszám nagyobb mint 4

SELECT COUNT(Darabszám) AS Darabszám FROM Beszállítás

WHERE Darabszám>4;

$\pi_{\text{COUNT}(\text{darabszám}) \rightarrow \text{darabszám}}$
 $\gamma_{\text{COUNT}(\text{darabszám})}$
 $\sigma_{\text{darabszám} > 4} \text{beszállítás}$

	Darabszám
▶	3

5.Listázzuk ki a legdrágább rendeléssel rendelkező ügyfél adatait

select * from ügyfél

join Rendelés on ügyfél.Személyszám=Rendelés.Személyszám

where ár= (SELECT MAX(ár) FROM rendelés);

$\text{ügyfél} \bowtie_{\text{ügyfél.személyszám} = \text{rendelés.személyszám}} \text{rendelés}$

π MAX (ár)
 γ MAX (ár) rendelés

Irányítószám	Város	Utca	Házszám	Személyszám	Név	VásárlásIdőpontja	VásároltTermék	RendelésIdőpontja	VárhatóÉrkezés	Ár	Helyzet	Személyszám	Raktár_ID
3906	Golop	József Attila út	11	1944882BE	Kiss Lajos	2022-11-24	Xbox Series X	5	2022-11-24	380000	Kiszállásra Átadva	1944882BE	5

6.Listázzuk ki azokat az ügyfeleket akiknek a rendelése depoban van és értékük nagyobb mint 10.000.

select * from ügyfél

join Rendelés on ügyfél.Személyszám=Rendelés.Személyszám

where Helyzet='Depoban' and ár>10000;

σ helyzet = "Depoban" AND ár > 10000 (ügyfél \bowtie ügyfél . személyszám = rendelés . személyszám rendelés)

Irányítószám	Város	Utca	Házszám	Személyszám	Név	VásárlásIdőpontja	VásároltTermék	RendelésIdőpontja	VárhatóÉrkezés	Ár	Helyzet	Személyszám	Raktár_ID
3500	Miskolc	Kazinczy út	51	1972351KE	Tóth János	2022-10-22	Samsung Galaxy A53	2	2022-10-22	160000	Depoban	1972351KE	2

7.írjuk ki és rendezzük név szerint azokat az ügyfeleket akiknek házsámában van 1-es.

select * from ügyfél

where Házsám like '%1%'

Order by Név;

τ név
 σ házsám LIKE "%1%" ügyfél

Irányítószám	Város	Utca	Házszám	Személyszám	Név	VásárlásIdőpontja	VásároltTermék
3907	Rátka	Kossuth út	71	1831928KI	Csanád Gergő	2022-09-01	Playstation 4
3906	Golop	József Attila út	11	1944882BE	Kiss Lajos	2022-11-24	Xbox Series X
3907	Tálya	Szabadság út	21	1826951HJ	Nagy Levente	2022-11-01	Samsung TV
3500	Miskolc	Kazinczy út	51	1972351KE	Tóth János	2022-10-22	Samsung Galaxy A53
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8.Listázzuk ki azon ügyfelek SzemélyiSzámát amelyeknek a terméke később érkezett raktárba mint 2022-09-01.

select SzemélyiSzám,RaktárbaÉrkezés from rendelés

join raktárak on raktárak.Raktár_ID=rendelés.Raktár_ID

where RaktárbaÉrkezés > '2022-09-12';

π személyszám, raktárbaérkezés
 σ raktárbaérkezés > "2022-09-12" (rendelés \bowtie raktárak . raktár_id = rendelés . raktár_id raktárak)

SzemélyiSzám	RaktárbaÉrkezés
1928374IE	2022-10-11
1972351KE	2022-10-19
1826951HJ	2022-11-12

9.Listázzuk ki azon raktárak ID-ját ahol ügyfeleknek valamilyen Samsung típusú eszköze van.

```

select raktarak.Raktár_ID from raktarak
join rendelés on rendelés.Raktár_ID = raktarak.Raktár_ID
join ügyfél on rendelés.Személyszám=ügyfél.Személyszám
where VásároltTermék like '%Samsung%';

```

$\pi_{\text{raktarak.raktár_id}}$
 $\sigma_{\text{vásárolttermék LIKE \%Samsung\%}} (\text{raktarak} \bowtie \text{rendelés} \bowtie \text{raktár_id} = \text{raktarak.raktár_id} \text{ rendelés} \bowtie \text{rendelés.személyszám} = \text{ügyfél.személyszám} \text{ ügyfél})$

	Raktár_ID
▶	2
	3

10.Írassuk ki azokat a beszállítókat ahol az ár nagyobb mint a beszállítói átlagár.

```

select * from beszállító
where Ár>(select avg(Ár) from beszállító);

```

$\pi_{\text{AVG}(\text{ár})}$
 $\gamma_{\text{AVG}(\text{ár})} \text{ beszállító}$

	Beszállító_ID	Ár	CsomagolásTípusa	VárhatóÉrkezés
▶	4	3250	2	2022-11-10
	5	4500	1	2022-11-25
*	NULL	NULL	NULL	NULL