

# JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves feladat

Csomagkövető rendszer

Készítette: Gyáni Kevin Zsolt

Neptunkód: CBOYZF

Dátum: 2022.11.19

|  |           |
|--|-----------|
| <b>A feladat leírása: .....</b>                                  | <b>3</b>  |
| <b>1 a, Az adatbázis ER-modellje:.....</b>                       | <b>5</b>  |
| <b>1b, Az ER-modell relációs modellre történő konvertálása:6</b> |           |
| <b>1c, Az adatbázis relációs sémája:.....</b>                    | <b>7</b>  |
| <b>1d, Táblák létrehozása.....</b>                               | <b>8</b>  |
| <b>1,e, Az adattáblák feltöltése: .....</b>                      | <b>11</b> |
| <b>1f, Lekérdezések (SQL és relációs algebra) .....</b>          | <b>12</b> |

## A feladat leírása:

A beadandó feladatom során, egy számítás technikai eszközök eladásával foglalkozó internetes áruház csomagkövető rendszerét igyekeztem lemodellezni. Az áruház egyebek mellett teszteli is a számítás technikai eszközöket, szamon tartja a gyárihibás darabokat, illetve több raktárral és futárszolgálattal 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 mellett 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.

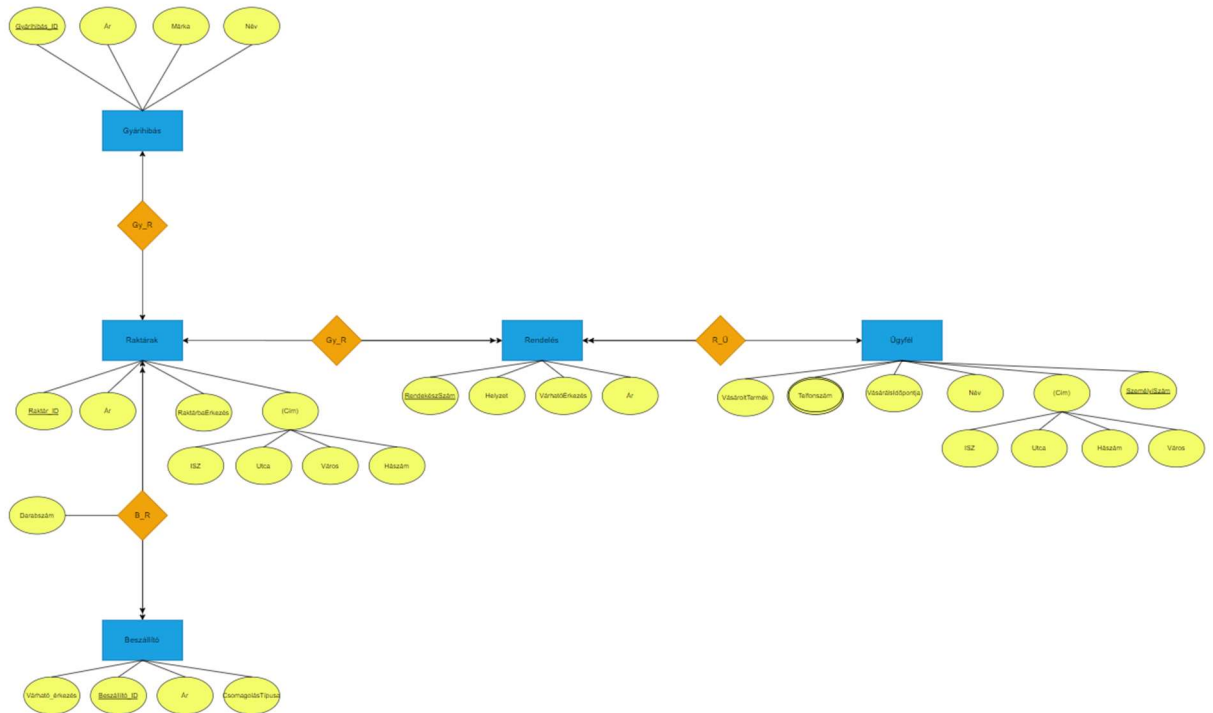
Amennyiben a termék működése helyénvalónak bizonyul a raktárból a terméket feladják a kért **rendelésre**. Ebben az egyedben vannak tárolva az ezzel kapcsolatos információk, például a rendelés száma (ezek lesznek az egyedi kulcsok), a csomagkövetés érdekében a rendelés helyzete, illetve várható érkezése, 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 duplikációk 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.

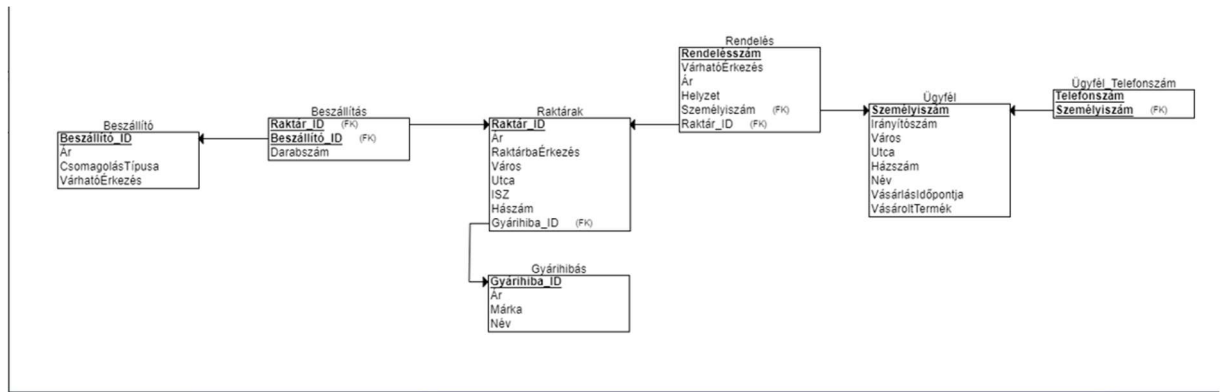
## 1 a, Az adatbázis ER-modellje:

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.



## 1b, Az ER-modell relációs modellre történő konvertálása:

Már megszokott módon, balról jobbra haladva rendeztem el a táblákat, az elsődleges kulcsokat beállítottam az ID-eknek 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 többértékkel rendelkező telefonszám tulajdonságot tartja számon.



### 1c, 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]

## 1d) 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

```
(  
    Rendelészám 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 (Rendelészám),  
    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)  
);
```

## 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);
```

## 1f) 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;`

$$\sigma_{gyárihibás.ár = raktarak.ár} (gyárihibás \bowtie_{gyárihibás.gyárihiba\_id = raktarak.gyárihiba\_id} raktarak)$$

|   | Gyárihiba_ID | Ár   | Márka | Név              | Raktár_ID | Ár   | RaktárbaÉrkezés | Város       | Utca    | ISZ | Hászám | 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%';`

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

|   | Irányítószám | Város | Utca         | Hászám | 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         | Tálya | 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;`

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

|   | Darabszám |
|---|-----------|
| ▶ | 3         |

## 5. Listázzuk ki a legdrágább rendeléssel rendelkező ügyfél adatait

select \* from ügyfél

join Rendelés on ügyfél.SzemélyisZám=Rendelés.SzemélyisZám

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

```

    ügyfél
    MAX(ár)
    MAX(ár) rendelés

```

| Irányítószám | Város | Ulica            | Házszám | SzemélyisZám | Név        | Vásárlásidőpontja | VásároltTermék | RendelésZám | VárhatóÉrkezés | Ár     | Helyzet            | SzemélyisZám | 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 kaikknek a rendelése depoban vannak és értékük nagyobb mint 10.000

select \* from ügyfél

join Rendelés on ügyfél.SzemélyisZám=Rendelés.SzemélyisZám

where Helyzet='Depoban' and ár>10000;

```

    Helyzet = "Depoban" AND ár > 10000
    ügyfél
    MAX(ár)
    MAX(ár) rendelés

```

| Irányítószám | Város   | Ulica       | Házszám | SzemélyisZám | Név        | Vásárlásidőpontja | VásároltTermék     | RendelésZám | VárhatóÉrkezés | Ár    | Helyzet | SzemélyisZám | Raktár_ID |
|--------------|---------|-------------|---------|--------------|------------|-------------------|--------------------|-------------|----------------|-------|---------|--------------|-----------|
| 3500         | Miskolc | Kazinczy út | 51      | 1972351KE    | Tóth János | 2022-10-22        | Samsung Galaxy A53 | 2           | 2022-10-22     | 16000 | Depoban | 1972351KE    | 2         |

## 7. Írjuk ki és rendezzük név szerint azokat az ügyfeleket akiknek házsZámában van 1-es

select \* from ügyfél

where Házszám like '%1%'

Order by Név;

```

    név
    házszám LIKE "%1%"
    ügyfél

```

| Irányítószám | Város   | Ulica            | Házszám | SzemélyisZám | 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';

```

    személyisZám, raktárbaérkezés
    raktárbaérkezés > "2022-09-12"
    rendelés
    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.SzemélyisZám=ügyfél.SzemélyisZám

where VásároltTermék like '%Samsung%';

$\pi_{raktarak . raktár\_id}$   
 $\sigma_{vásárolttermék LIKE \%Samsung\%} (raktarak \bowtie_{rendelés . raktár\_id = raktarak . raktár\_id} rendelés \bowtie_{rendelés . személyiszám = ügyfél . személyiszám} ügyfél)$

|   | 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);

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

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