



## - Adatbázis felépítése –

### Probléma meghatározása:

Az adatbázisban a *rendelések*, *kosarbolleadottrendeles*, és *megrendelesek* táblák között idegen kulcsokkal definiált kapcsolatok vannak, amelyek biztosítják az adatkonzisztenciát. A probléma abból adódik, hogy a **szülőtábla (Rendelések) rekordját nem lehet törölni, amíg az ahhoz kapcsolódó gyermek rekordok (Megrendelesek, kosarbolleadottrendeles) léteznek**. Ez a korlátozás az idegen kulcsok miatt van, amik megakadályozzák, hogy a törlés megsértse az adatkapcsolatokat. A **megoldás** lehet a gyermek táblákban lévő kapcsolódó rekordok törlése, mielőtt a szülőt törölnénk, vagy a kapcsolatok CASCADE törlési szabályának alkalmazása. Ezzel a megközelítéssel a szülőtábla rekordjainak törlése automatikusan törli a hozzá kapcsolódó gyermek rekordokat is.

### Milyen adatokra van szükség?

#### 1. Felhasználók adatai:

- ❖ **Felhasznalo tábla:** Felhasználó neve, e-mail címe, telefonszáma, jelszava, azonosítója (*Szemely\_id*).

#### 2. Rendelésekkel kapcsolatos adatok:

- ❖ **Rendelések tábla:** A rendelés azonosítója (*Rendeles\_id*), személyhez kapcsolódó azonosító (*Szemely\_id*), visszajelzés (pl. "Teljesítve", "Folyamatban").

#### 3. Kosárból leadott rendelésekkel kapcsolatos adatok:

- ❖ **kosarbolleadottrendeles tábla:** Kosár azonosító (*Kosar\_id*), termékek adatai (id, név, ár, cikkszám), rendelésekhez tartozó adatok.

#### 4. Megrendelések adatai:

- ❖ **Megrendelések tábla:** Egyedi megrendelés azonosítók (*Megrendeles\_id*), termékek, árak, mennyiségek, dátumok és megjegyzések, cikkszámok stb.

#### 5. Termékek és kosarak kapcsolódó adatai:

- ❖ **Termékek:** azonosításra szolgáló id/cikkszámok, nevek, elérhetőségek, leírások, árak.
- ❖ **Korábbi rendelések:** a kosár adataival megegyező adatok

## Milyen kapcsolatok léteznek?

### 1. Felhasználók és rendeléseik között:

- ❖ A **Felhasznalo** és a **Rendelések** tábla között az *Szemely\_id* az idegen kulcs, amely meghatározza, melyik rendelés melyik diákhoz tartozik.

### 2. Rendelések és kosarak között:

- ❖ A **Rendelések** és **Megrendelesek** táblák között a *Rendeles\_id* az idegen kulcs.

### 3. Kosarak és leadott rendelések között:

- ❖ A **kosarbolleadottrendeles** és a **Megrendelesek** között a *Kosar\_id* köti össze a kosarak tartalmát a leadott rendeléssel.

### 4. Megrendelések és termékek között:

- ❖ A termékeket az *Termek\_id* azonosító kapcsolja a kosarakhoz és a rendeléseken keresztül a diákokhoz.

## Táblák a **Diák** felhasználó oldalnál:

- 1) Felhasználó
- 2) Kosárból leadott rendelések
- 3) Termékek
- 4) Korábbi rendelések

### *Táblák elemeinek kifejtése/magyarázata:*

Felhasználó
Szemely_id (int, primary key)
Nev (varchar)
Telefonszam (varchar)
Email (varchar)
Jelszo (varchar)
Szerep (int)

### Felhasználó tábla célja:

Arra szolgál, hogy az alkalmazásban regisztrált **felhasználók adatait tároljuk**. Ez az alapja a felhasználókezelésnek, és biztosítja az egyedi azonosítást, valamint a kapcsolattartáshoz szükséges adatok tárolását. Az **e-mail** és **telefonszám** mezők lehetőséget biztosítanak a felhasználókkal történő kommunikációra.

- **Szemely\_id (int, primary key):** Egyedi azonosító, ami minden felhasználót megkülönböztet az adatbázisban.
- **Nev (varchar):** A felhasználó által használt egyedi felhasználónév.
- **Telefonszam (varchar):** A felhasználó telefonszáma, ami kapcsolattartási célt szolgál.
- **Email (varchar):** A felhasználó e-mail címe a bejelentkezéshez, kapcsolattartáshoz és értesítésekhez.
- **Jelszo (varchar):** A felhasználó által megadott jelszó, amely a bejelentkezéshez szükséges.
- **Szerep (int):** Meghatározza, hogy a felhasználó diákként (0) vagy eladóként (1) van regisztrálva.

Kosárból leadott rendelés
Termek_id (int, foreign key)
Szemely_id (int, foreign key)
Rendeles_id (int)
Cikkszam (int)
Termeknev (varchar)
Ar (int)
Kert_szunet (int)
Összeg_Ft (int)
Allapot (varchar)
kosar_id (int, primary key)

### Kosár tábla célja:

Arra szolgál, hogy az alkalmazásban a felhasználók által kiválasztott **termékek tárolásra kerüljenek**, amit majd a felhasználó **leadhat rendelésként**. Tartalmazza a termékek sorszámát és egy kosár\_id-t, a táblák könnyebb összekapcsolása érdekében, valamint a diák nevét, a termékek adatait és azt, hogy melyik szünetre kéri a rendelést a diák.

- **kosar\_id (int, primary key):** Egyedi azonosító, amely az adott kosár bejegyzést azonosítja.
- **Termek\_id (int, foreign key):** A kosárhoz tartozó termék egyedi azonosítója, amely összekapcsolódik a Termékek táblával.
- **Szemely\_id (int, foreign key):** Az adott rendelést leadó személy azonosítója, amely a felhasználókhoz kapcsolódik.
- **Rendeles\_id (int):** Az adott kosárban szereplő rendelés azonosítója, amely a rendelések nyomon követésére szolgál.
- **Cikkszam (int):** Az adott termék cikkszáma a rendelésben.
- **Termeknev (varchar):** A kosárban szereplő termék neve.
- **Ar (int):** A kosárban szereplő termék ára.
- **Kert\_szunet (int):** Az adott rendeléshez tartozó választott szünet száma.
- **Összeg\_Ft (int):** Az adott rendelésben szereplő összes termék összértéke.
- **Allapot (varchar):** Az adott kosár rendelésének aktuális állapota (pl.: „függőben”, „teljesített”, „elutasított”).

Termékek
termek_id (int, primary key)
Cikkszam (int)
Termeknev (varchar)
Elerhetőség (varchar)
Leiras (varchar)
Ar (int)

### Termékek tábla célja:

Arra szolgál, hogy az alkalmazásban elérhető **termékek adatait tárolja**, és biztosítja azok kezelését, megjelenítését, valamint a felhasználók általi rendelését. Ez az alapja az online áruház működésének, mivel minden egyes termékéről tartalmazza azokat az információkat, amelyek szükségesek a termékek bemutatásához és a vásárlók számára történő értékesítéshez.

- **Termek\_id (int, primary key):** Ez az oszlop a termékek egyedi azonosítóját tárolja.
- **Cikkszam (int):** A termék cikkszáma.
- **Termeknev (varchar):** A termék neve.
- **Elerhetőség (varchar):** A termék elérhetősége (pl. készleten van-e).
- **Leiras (varchar):** A termék leírása, pl.: üdítőből milyen ízű, hány literes stb.
- **Ar (int):** A termék ára.

Korábbi rendelések	
Termek_id (int, foreign key)	
*Rendeles_id (int, primary key)	>
Termeknev (varchar)	
Cikkszam (int)	
*Ar (int)	>
Kert_szunet(int)	
*Mennyiseg (int)	>
Összeg_Ft (int)	
Szemely_id (int, foreign key)	

### Korábbi rendelések tábla célja:

Arra szolgál, hogy **nyomon kövesse** a felhasználók által már **leadott rendeléseket**, azok státuszát, tartalmát és **részleteit**. Ez biztosítja a vásárlási előzmények tárolását, amelyek segítenek a felhasználóknak az egyes rendeléseik nyomon követésében. Valamint a különböző id-k segítenek más táblákkal való összeköttetésben.

- **Termek\_id (int, foreign key):** Külső kulcs, amely egy termék táblához kapcsolódik.
- **Rendeles\_id (int, primary key):** Elsődleges kulcs, amely az egyedi rendeléseket azonosítja.
- **Termeknev (varchar):** A termék neve.
- **Cikkszam (int):** Termékazonosító - cikkszám.
- **Ar (int):** A termék ára.
- **Kert\_szunet (int):** Az adott rendeléshez tartozó választott szünet száma.
- **Mennyiseg (int):** A felhasználó által leadott termékek mennyisége.
- **Összeg\_Ft (int):** A rendelés összege (Ft).
- **Szemely\_id (int, foreign key):** Külső kulcs, amely egy adott személyt azonosít a rendeléshez kapcsolódóan.

## Táblák az **Eladó** felhasználó oldalnál:

- 1) Felhasználó (fentebb már kifejtve)
- 2) Megrendelések

### *Táblák elemeinek kifejtése/magyarázata:*

Megrendelések
Megrendelesek_id (int, primary key)
kosar_id (int, foreign key)
Szemely_id (int, foreign key)
Diak_neve (varchar)
Termeknev (varchar)
Cikkszam (int)
Ar (int)
Összeg_Ft (int)
Valasztott_szunet (int)
Datum (date)
Megjegyzések (varchar)

### **Megrendelések tábla célja:**

Célja, hogy **nyomon kövesse a felhasználók által leadott rendeléseket**, tartalmazza az összes releváns adatot, például a rendelés azonosítóját, a vásárló személyes adatait, a rendeléshez kapcsolódó termékeket. A **kosar\_id**: a kosár, amelyből a rendelés létrejött. Ez a mező a *Kosarak* táblára mutat (külső kulcs), és biztosítja, hogy a rendelés kapcsolódjon a felhasználó kosarához. A **szemely\_id**: ez a mező a *Felhasználók* táblára mutat, így lehetővé teszi a rendelés összekapcsolását a felhasználóval. **Megjegyzések**: a mezőn keresztül lehetőség van tárolni a rendelés állapotát.

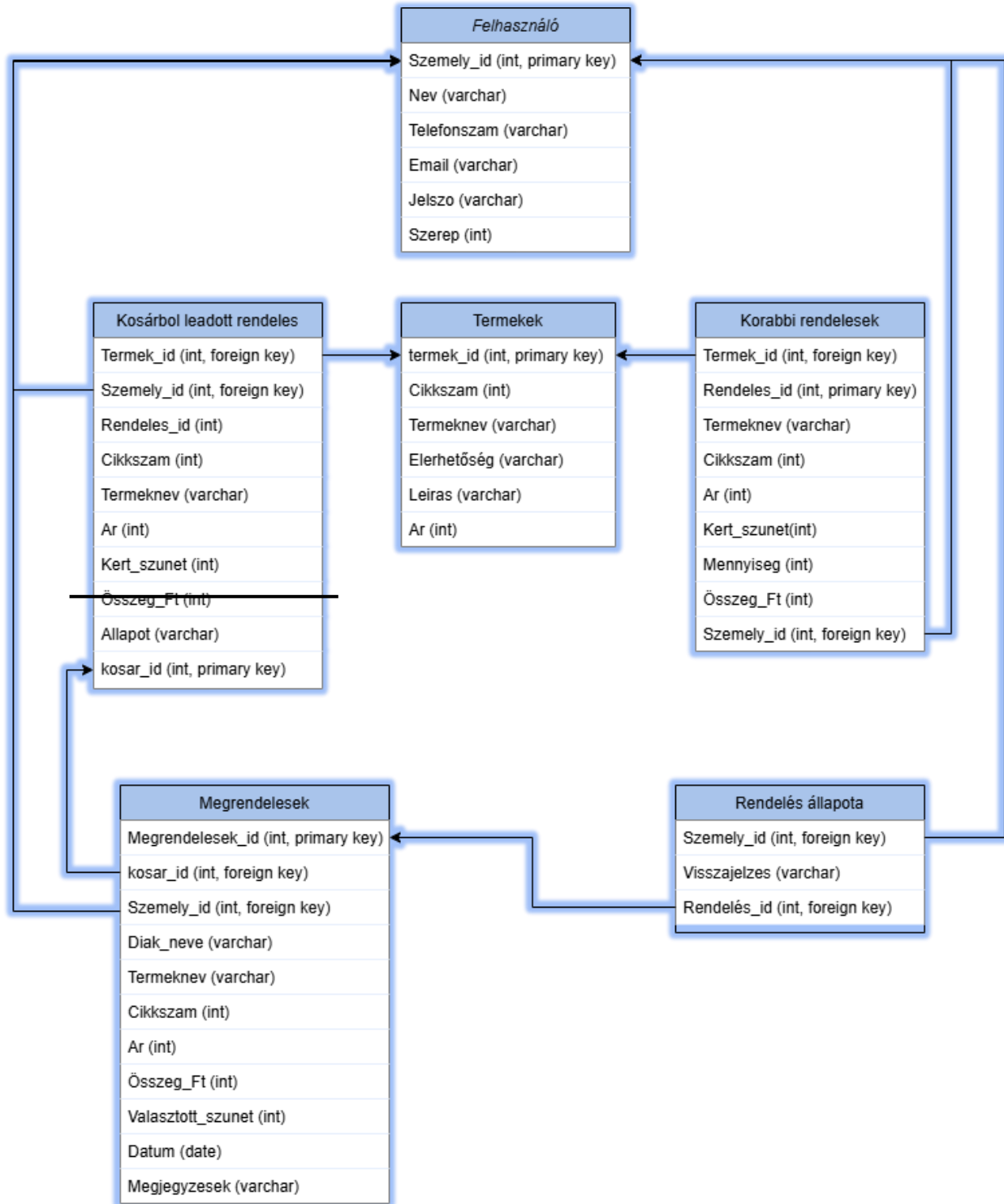
- **Megrendelesek\_id (int, primary key)** - Az elsődleges kulcs.
- **kosar\_id (int, foreign key)** - Külső kulcs, amely egy másik táblához kapcsolódik
- **Szemely\_id (int, foreign key)** - Egy másik külső kulcs, amely a rendelésben részt vevő személyhez kapcsolódik.
- **Diak\_neve (varchar)**: Egy személy vagy diák neve.
- **Termeknev (varchar)**: A termék neve.
- **Cikkszam (int)**: Cikkszám vagy termékazonosító.
- **Ar (int)**: Termék ára.
- **Összeg\_Ft (int)**: Rendelés összege.
- **Valasztott\_szunet (int)**: Az adott rendeléshez tartozó választott szünet száma.
- **Datum (date)**: Az rendelés dátumát tartalmazza.
- **Megjegyzések (varchar)**: Megjegyzéseket tehet a diák az eladó számára a rendelésével kapcsolatban.

Rendelés állapota
Személy_id (int, foreign key)
Visszajelzes (varchar)
Rendelés_id (int)

### Rendelések állapota tábla célja:

Az eladó visszajelzést tud küldeni ezáltal a diáknak a rendelése állapotáról egy szöveges üzenetként.

- **Személy\_id (int, foreign key)** - A rendelést végrehajtó személy azonosítóját jelöli, amely egy külső kulcs másik táblához.
- **Visszajelzés (varchar)** - Egy szöveges mező, az adott rendelés állapotáról tájékoztathatja az eladó a diákot (pl.: „függőben”, „teljesített”, „elutasított”).
- **Rendelés\_id (int)** - Egy egész szám típusú mező, amely a rendelés azonosítására szolgál.



## - Sql -

```
SQL lekérdezés(ek) futtatása a(z) quickserve adatbázison: ⓘ

1 CREATE TABLE termek (
2   Termek_id INT AUTO_INCREMENT PRIMARY KEY,
3   Cikkszam INT,
4   Termeknev VARCHAR(255),
5   Elerhetoseg VARCHAR(50),
6   Leiras VARCHAR(255),
7   Ar INT
8 );
```

//kód

```
CREATE TABLE termek ( Termek_id
INT AUTO_INCREMENT PRIMARY K
EY, Cikkszam INT, Termeknev VARCH
AR(255), Elerhetoseg VARCHAR(50), L
eiras VARCHAR(255), Ar INT );
```

//termekek tábla létrehozása

```
SQL lekérdezés(ek) futtatása a(z) quickserve.termekek táblán: ⓘ

1 INSERT INTO termek (Termek_id, Cikkszam, Termeknev, Elerhetoseg, Leiras
2 (1, 101, 'Szendvics ', 'Elérhető', 'Friss sonkás szendvics', 350),
3 (2, 102, 'Szendvics ', 'Nem elérhető', 'Friss sajtos szendvics', 300),
4 (3, 103, 'Topjoy', 'Elérhető', 'kaktusz', 250),
5 (4, 104, 'Cappy 250ml', 'Elérhető', 'alma', 200),
6 (5, 105, 'Xixo ', 'Nem elérhető', 'eper', 250),
7 (6, 106, 'Milka ', 'Elérhető', 'tejcokoládé', 350),
8 (7, 107, 'Haribo ', 'Elérhető', '200g gumicukor', 300),
9 (8, 108, 'Lays chips', 'Nem elérhető', 'sós ', 250),
10 (9, 109, 'Lays chips', 'Elérhető', 'paprikás', 250),
11 (10, 110, 'Milka', 'Elérhető', 'kekszes tejcokoládé', 400);
12
```

```
INSERT INTO termek (Termek_id, Cikkszam,
Termeknev, Elerhetoseg, Leiras, Ar) VALUES
```

```
(1, 101, 'Szendvics ', 'Elérhető', 'Friss sonkás
szendvics', 350),
```

```
(2, 102, 'Szendvics ', 'Nem elérhető', 'Friss sajtos
szendvics', 300),
```

```
(3, 103, 'Topjoy', 'Elérhető', 'kaktusz', 250),
```

//termekek tábla feltöltése

```
SQL lekérdezés(ek) futtatása a(z) quickserve adatbázison: ⓘ

1 CREATE TABLE Felhasznalo (
2   Szemely_id INT PRIMARY KEY,
3   Nev VARCHAR(255),
4   Telefonszam VARCHAR(50),
5   Email VARCHAR(255),
6   Jelszo VARCHAR(255),
7   Szerep INT
8 );
9
```

CREATE TABLE Felhasznalo (

Szemely\_id INT PRIMARY KEY,

Nev VARCHAR(255),

Telefonszam VARCHAR(50),

Email VARCHAR(255),

Jelszo VARCHAR(255),

Szerep INT

//felhasznalo tábla létrehozása



```
INSERT INTO Felhasznalo (Szemely_id, Nev, Telefonszam, Email)
VALUES
(1, 'Kovács János', '06123456789', 'kovacs.janos@example.com'),
(2, 'Nagy Anna', '06201234567', 'nagy.anna@example.com'),
(3, 'Szabó Péter', '06304567890', 'szabo.peter@example.com'),
(4, 'Varga Mária', '06431234567', 'varga.maria@example.com'),
(5, 'Tóth Zoltán', '06501234567', 'toth.zoltan@example.com');
```

```
INSERT INTO Felhasznalo (Szemely_id, Nev,
Telefonszam, Email, Jelszo, Szerep) VALUES
```

```
(1, 'Kovács János', '06123456789',
'kovacs.janos@example.com', 'janos123', 1),
```

```
(2, 'Nagy Anna', '06201234567',
'nagy.anna@example.com', 'anna456', 2);
```

//felhasznalo tábla feltöltése

SQL lekérdezés(ek) futtatása a(z) quickserve adatbázison:

```
1 CREATE TABLE kosarbolleadottrendeles (
2   Kosar_id INT AUTO_INCREMENT PRIMARY KEY,
3   Termek_id INT,
4   Szemely_id INT,
5   Rendeles_id INT,
6   Cikkszam INT,
7   Termeknev VARCHAR(255),
8   Ar INT,
9   Kert_szunet INT,
10  Osszeg_ft INT,
11  Allapot VARCHAR(255),
12  FOREIGN KEY (Termek_id) REFERENCES termekek(Termek_id),
13  FOREIGN KEY (Szemely_id) REFERENCES felhasznalo(Szemely_id)
14 );
15
```

CREATE TABLE kosarbolleadottrendeles (

Kosar\_id INT AUTO\_INCREMENT PRIMARY KEY,

Termek\_id INT,

Szemely\_id INT,

Rendeles\_id INT,

Cikkszam INT,

Termeknev VARCHAR(255),

//kosarba leadott rendelések tábla létrehozása

SQL lekérdezés(ek) futtatása a(z) quickserve.kosarbolleadottrendeles táblán:

```
1 INSERT INTO kosarbolleadottrendeles
2   (Termek_id, Szemely_id, Rendeles_id, Cikkszam, Termeknev,
3   (1, 1, 1001, 101, 'Szendvics', 350, 2, 3500, 'Feldolgozás
4   (2, 1, 1002, 102, 'Szendvics', 300, 5, 1500, 'Feldolgozás
5   (3, 2, 1003, 103, 'Topjoy', 250, 3, 2000, 'Feldolgozás ala
6   (4, 2, 1004, 104, 'Cappy 250ml', 200, 10, 2000, 'Feldolgoz
7   (5, 3, 1005, 105, 'Xixo', 250, 6, 1500, 'Feldolgozás alatt
8   (6, 3, 1006, 106, 'Milka', 350, 4, 1400, 'Feldolgozás alát
9   (7, 4, 1007, 107, 'Haribo', 300, 3, 1500, 'Feldolgozás ala
10  (8, 4, 1008, 108, 'Lays chips', 250, 2, 750, 'Feldolgozás
11  (9, 5, 1009, 109, 'Lays chips', 250, 1, 1750, 'Feldolgozás
12  (10, 5, 1010, 110, 'Milka', 400, 2, 800, 'Feldolgozás alát
13
```

```
INSERT INTO kosarbolleadottrendeles (Termek_id,
Szemely_id, Rendeles_id, Cikkszam, Termeknev, Ar,
Kert_szunet, Osszeg_ft, Allapot) VALUES (1, 1, 100
1, 101, 'Szendvics', 350, 2, 3500, 'Feldolgozás
alatt'), (2, 1, 1002, 102, 'Szendvics', 300, 5, 1500, 'Fel
dolgozás
alatt'), (3, 2, 1003, 103, 'Topjoy', 250, 3, 2000, 'Feldol
gozás
alatt'), (4, 2, 1004, 104, 'Cappy
250ml', 200, 10, 2000, 'Feldolgozás
alatt'), (5, 3, 1005, 105, 'Xixo', 250, 6, 1500, 'Feldolgo
zás
```

//kosarba leadott rendelések tábla feltöltése

SQL lekérdezés(ek) futtatása a(z) quickserve adatbázison:

```

1 CREATE TABLE korabbi_rendelesek (
2     Termek_id INT,
3     Rendeles_id INT PRIMARY KEY,
4     Termeknev VARCHAR(255),
5     Cikkszam INT,
6     Ar INT,
7     Kert_szunet INT,
8     Mennyiseg INT,
9     Osszeg_Ft INT,
10    Szemely_id INT,
11    FOREIGN KEY (Termek_id) REFERENCES termekek(Termek_id),
12    FOREIGN KEY (Szemely_id) REFERENCES felhasznalok(Szemely_id),
13 );

```

```

INSERT INTO korabbi_rendelesek (Termek_id,
Rendeles_id, Termeknev, Cikkszam, Ar,
Kert_szunet, Mennyiseg, Osszeg_ft, Szemely_id)
VALUES

```

```
(1, 1001, 'Szendvics', 101, 350, 2, 1, 3500, 1),
```

```
(2, 1002, 'Szendvics', 102, 300, 5, 1, 1500, 1),
```

```
(3, 1003, 'Topjoy', 103, 250, 3, 2, 2000, 2),
```

```
(4, 1004, 'Cappy 250ml', 104, 200, 10, 2, 2000, 2),
```

//korabbi\_rendelesek tábla létrehozása + feltöltése

lekérdezés(ek) futtatása a(z) quickserve adatbázison:

```

1 CREATE TABLE Megrendelesek (
2     Megrendeles_id INT PRIMARY KEY,
3     kosar_id INT,
4     Szemely_id INT,
5     Diak_neve VARCHAR(100),
6     Termeknev VARCHAR(100),
7     Cikkszam INT,
8     Ar INT,
9     Osszeg_Ft INT,
10    Valasztott_szunet INT,
11    Datum DATE,
12    Megjegyzesek varchar(255),
13    FOREIGN KEY (kosar_id) REFERENCES kosarbolleadottrendeles(kosar_id),
14    FOREIGN KEY (Szemely_id) REFERENCES Felhasznalo(Szemely_id),
15 );

```

```

CREATE TABLE Megrendelesek (
    Megrendeles_id INT PRIMARY KEY,
    Kosar_id INT,
    Szemely_id INT,
    Diak_neve VARCHAR(100),
    Termeknev VARCHAR(100),
    Cikkszam INT,
    Ar INT,

```

//megrendelesek tábla létrehozása

```

INSERT INTO Megrendelesek (Megrendeles_id, Kosar_id, Szemely_id, Diak_neve,
Megjegyzesek) VALUES
(1, 1, 1, 'Kovács János', 'Szendvics', 101, 350, 3500, 2, '2024-11-18',
(2, 2, 1, 'Kovács János', 'Szendvics', 102, 300, 1500, 5, '2024-11-18',
(3, 3, 2, 'Nagy Anna', 'Topjoy', 103, 250, 2000, 3, '2024-11-18', 'Elut
(4, 4, 2, 'Nagy Anna', 'Cappy 250ml', 104, 200, 2000, 10, '2024-11-18',
(5, 5, 3, 'Szabó Péter', 'Xixo', 105, 250, 1500, 6, '2024-11-18', 'Függ
(6, 6, 3, 'Szabó Péter', 'Milka', 106, 350, 1400, 4, '2024-11-18', 'Függ
(7, 7, 4, 'Varga Mária', 'Haribo', 107, 300, 1500, 3, '2024-11-18', 'Függ
(8, 8, 4, 'Varga Mária', 'Lays chips', 108, 250, 750, 2, '2024-11-18',
(9, 9, 5, 'Tóth Zoltán', 'Lays chips', 109, 250, 1750, 1, '2024-11-18',
(10,10, 5, 'Tóth Zoltán', 'Milka', 110, 400, 800, 2, '2024-11-18', 'Tel

```

```

INSERT INTO Megrendelesek (Megrendeles_id,
Kosar_id, Szemely_id, Diak_neve, Termeknev,
Cikkszam, Ar, Osszeg_Ft, Valasztott_szunet, Datum,
Megjegyzesek) VALUES

```

```
(1, 1, 1, 'Kovács János', 'Szendvics', 101, 350, 3500,
2, '2024-11-18', 'Teljesitve'),
```

```
(2, 2, 1, 'Kovács János', 'Szendvics', 102, 300, 1500,
5 '2024-11-18' 'Függőben')
```

//megrendelesek tábla feltöltése

SQL lekérdezés(ek) futtatása a(z) quickserve adatbázison:

```
1 CREATE TABLE Rendelések (  
2     Rendelés_id INT PRIMARY KEY,  
3     Szemely_id INT,  
4     Visszajelzes VARCHAR(255),  
5     FOREIGN KEY (Szemely_id) REFERENCES Felhasznalo(Szemely_id),  
6     FOREIGN KEY (Rendelés_id) REFERENCES Megrendelesek(Megrendeles_id)  
7 );  
8
```

CREATE TABLE Rendelések (

Rendelés\_id INT PRIMARY KEY,

Szemely\_id INT,

Visszajelzes VARCHAR(255),

FOREIGN KEY (Szemely\_id) REFERENCES

//rendelesek tábla létrehozása

L lekérdezés(ek) futtatása a(z) quickserve.rendelesek táblán:

```
1 INSERT INTO Rendelések (Rendelés_id, Szemely_id, Visszajelzes)  
2 VALUES  
3 (1, 1, 'Folyamatban'),  
4 (2, 2, 'Teljesitve'),  
5 (3, 3, 'Elutasitva'),  
6 (4, 4, 'Folyamatban'),  
7 (5, 5, 'Teljesitve');  
8
```

INSERT INTO Rendelések (Rendelés\_id,  
Szemely\_id, Visszajelzes)

VALUES

(1, 1, 'Folyamatban'),

(2, 2, 'Teljesitve'),

(3, 3, 'Elutasitva'),

//rendelesek tábla feltöltése

## LEKÉRDEZÉSEK

- 1) Jelenítsd meg azoknak a felhasználóknak a nevét és telefonszámát, akik rendeltek valamilyen terméket, és a rendelésük állapota "Folyamatban".

```
SELECT F.Nev, F.Telefonszam FROM felhasznalo F INNER join rendelések R on  
R.Szemely_id=F.Szemely_id WHERE R.Visszajelzes='Folyamatban';
```

Nev	Telefonszam
Kovács János	06123456789
Varga Mária	06431234567

2) Írj egy lekérdezést, ami csoportosítja a rendeléseket és egy sorban megjeleníti az adott rendeléshez tartozó összes termék cikkszámát, nevét, valamint a termékek árának összegét.

```
SELECT K.Rendeles_id, GROUP_CONCAT(K.Cikkszam) AS Cikkszam, GROUP_CONCAT(K
.Termeknev) AS Termek, SUM(K.Ar) AS Osszeg FROM kosarbolleadottrendeles
K GROUP BY K.Rendeles_id;
```

	Rendeles_id	Cikkszam	Termek	Osszeg
s	1001	101,201,101	Szendvics,kinkat,Szendvics	900
s	1002	102,202,102	Szendvics,Ragó,Szendvics	700
s	1003	103	Topjoy	250

3) Írj egy lekérdezést, amely a rendeléseket és azok állapotát jeleníti meg, csak azokat a rendeléseket listázva, amelyekhez tartozó termékek ára meghaladja a 300 Ft-ot.

```
SELECT K.Rendeles_id, K.Allapot FROM kosarbolleadottrendeles K WHERE K.Ar > 3
00 GROUP BY K.Rendeles_id, K.Allapot;
```

	Rendeles_id	Allapot
és	1001	Feldolgozás alatt
és	1006	Feldolgozás alatt
és	1010	Feldolgozás alatt

4) Listázd ki a diákok neveit és a rendeléseik dátumát azok számára, akik rendeltek valamilyen terméket.

```
SELECT F.Nev, M.Datum FROM Felhasznalo F INNER JOIN Megrendelesek M ON F.Szemely_id = M.Szemely_id;
```

Nev	Datum
Kovács János	2024-11-18
Kovács János	2024-11-18
Nagy Anna	2024-11-18
Nagy Anna	2024-11-18
Szabó Péter	2024-11-18

- 5) Listázd ki azoknak a diákoknak a nevét és az általuk rendelt termékeket, akik több mint egy terméket rendeltek.

```
SELECT F.Nev, GROUP_CONCAT(K.Termeknev) AS Rendelték_Termekek FROM Felhasznalo F INNER JOIN kosarbolleadtrendeles K ON F.Szemely_id = K.Szemely_id GROUP BY F.Szemely_id HAVING COUNT(K.Termek_id) > 1;
```

Nev	Rendelték_Termekek
Kovács János	Szendvics,Szendvics,Szendvics,S
Nagy Anna	Topjoy,Cappy 250ml
Szabó Péter	Xixo,Milka
Varga Mária	Haribo,Lays chips
Tóth Zoltán	Lays chips,Milka

- 6) Listázd ki azoknak a diákoknak a nevét és a rendelés id-ját, akiknek a rendelési státusza "Folyamatban" van.

```
SELECT F.Nev, R.Rendelés_id FROM Felhasznalo F INNER JOIN Rendelések R ON F.Szemely_id = R.Szemely_id WHERE R.Visszajelzes = 'Folyamatban';
```

Nev	Rendelés_id
Kovács János	1
Varga Mária	4

- 7) Listázd ki azon diákok nevét, akik rendeltek olyan terméket, ahol a választott szünet ideje kisebb mint 3.

```
SELECT F.Nev, M.Termeknev, M.Valasztott_szunet FROM Felhasznalo F INNER JOIN Megrendelesek M ON F.Szemely_id = M.Szemely_id WHERE M.Valasztott_szunet < 3;
```

Nev	Termeknev	Valasztott_szunet
Kovács János	Szendvics	2
Varga Mária	Lays chips	2
Tóth Zoltán	Lays chips	1
Tóth Zoltán	Milka	2

- 8) Módosítsd azokat a megrendeléseket 'Teljesítve' értékre ahol a választott szünet kisebb vagy egyenlő, mint 3.

```
UPDATE Megrendelesek SET Megjegyzesek = 'Teljesítve' WHERE Valasztott_szunet <= 3;
```

3	2024-11-18	Teljesítve
2	2024-11-18	Teljesítve
1	2024-11-18	Teljesítve
2	2024-11-18	Teljesítve

- 9) Módosítsd a kosarbolleadottrendeles táblában szereplő termékek árát úgy, hogy azoknál a termékeknél, amelyek ára magasabb, mint az átlagos termékár, az ár 10%-kal csökkenjen.

```
UPDATE kosarbolleadottrendeles K INNER JOIN Megrendelesek M ON K.Rendeles_id = M.Megrendeles_id SET K.Ar = K.Ar * 0.9 WHERE K.Ar > ( SELECT AVG(Ar) FROM kosarbolleadottrendeles );
```

- 10) Töröld minden táblából azokat az adatokat, ahol a rendelés\_id 3.

✓ A(z) 1 sor törlése megtörtént. (A lekérdezés 0,0004 másodpercig tartott.)

```
DELETE FROM Megrendelesek WHERE Kosar_id IN (SELECT Kosar_id FROM kosarbolleadottrendeles WHERE Rendeles_id = 1003);
```

[ Szerkesztés helyben ] [ Módosítás ] [ PHP-kód létrehozása ]

✓ A(z) 1 sor törlése megtörtént. (A lekérdezés 0,0004 másodpercig tartott.)

```
DELETE FROM kosarbolleadottrendeles WHERE Rendeles_id = 1003;
```

[ Szerkesztés helyben ] [ Módosítás ] [ PHP-kód létrehozása ]

✓ A(z) 0 sor törlése megtörtént. (A lekérdezés 0,0004 másodpercig tartott.)

```
DELETE FROM Rendelesek WHERE Rendelés_id = 1003;
```

	Megrendeles_id	Kosar_id	Szemely_id
\$	1	1	1
\$	2	2	1
\$	4	4	2
\$	5	5	3