

# JEGYZŐKÖNYV

Adatbázisrendszerek I.

Féléves feladat: Online vásárlás

Készítette: **Tóth József**  
Neptunkód: **WI2GDP**  
Gyakorlat: **Szerda 12-14**  
Vezér: **Dr. Bednarik László**

## A feladat leírása:

A beadandóm során egy vagy több internetes áruházban található termék(ek) online megrendelését, és azon rendelésekben lévő termékek csomagszállítását, illetve a címzett személy fontosabb adatainak nyilvántartására törekszem.

Öt egyed lesz jelen az adatbázisban:

- Termékek
- Webshop
- Rendelések
- Csomagszállító
- Vásárló

A Termékek egyed talán a legegyszerűbb a többi egyeddel összehasonlítva, ugyanis nem tartalmaz speciális tulajdonságokat - kivéve egy elsődleges kulcsot -, ráadásul az egyed által birtokolt attribútumok is magától értetődőek. Ugyanakkor a Termékek és a Webshop egyedek N:M kapcsolatban állnak egymással, ami később a relációs modell létrehozásánál fontos tényező lesz. Több Webáruháznál is megjelenhet ugyanaz a termék, és több termék is megjelenhet ugyanabban az áruházban.

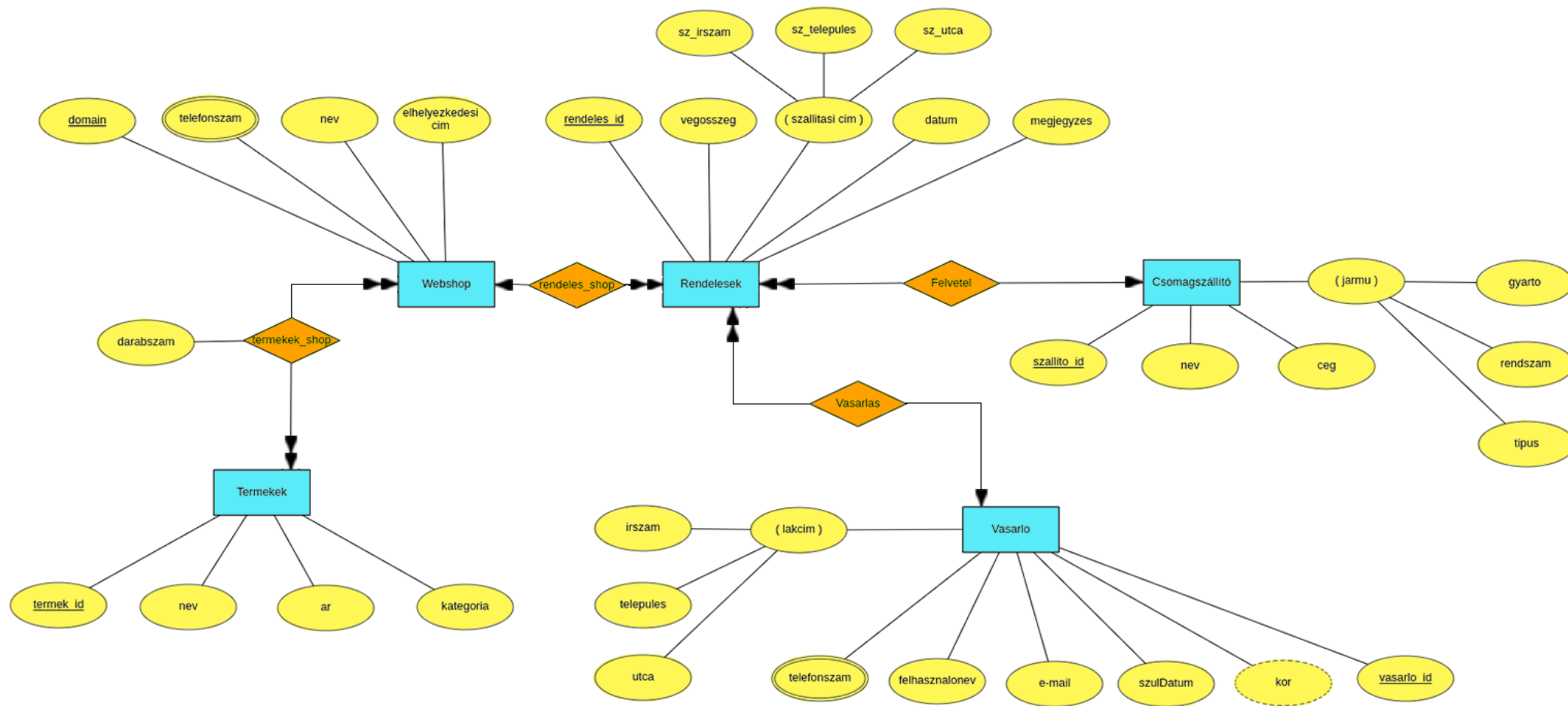
A Webshop egyed tartalmaz egy többértékű tulajdonságot 'telefonszam' néven, hiszen minden Webshop rendelkezik több telefonszámmal is, ha igénybe vennénk a technikai támogatást.

Ezt követően 1:N kapcsolatban van a Rendelések egyed, ami kulcsfontosságú hiszen ez képi az adatbázis gerincét, amin keresztül minden egyes táblát el tudunk érni. Emellett egy összetett tulajdonsággal is rendelkezik 'szallitasi cim' néven, amiben megtalálható az irányítószám, a település és az utca neve/száma. Mivel a vásárlók nem feltétlenül az állandó lakcímükre kérhetik a rendelést, így emiatt a rendelésben található szállítási cím változhat. A Rendelések egyed után két egyed külön ágra száll, így a Csomagszállító és a Vásárló egyed között nincs közvetlen kapcsolat.

A Csomagszállító egyed tulajdonság készlete hasonló a Rendelésekhez, mivel itt is van egy összetett tulajdonság 'jarmu' néven. Külön attribútumokra lehet osztani a gépjármű gyártóját, típusát és rendszámát.

Végül a Vasarlo egyedben egyszerre három speciális tulajdonság típus is jelen van. A 'lakcim' összetett attribútum azonos a Rendelésekben található 'szallitasi cim' tulajdonság felépítésével, de értékei nem feltétlenül lehetnek azonosak. A Webshop egyednél is megtalálható 'telefonszam' többértékű tulajdonság is jelen van a vásárlónál, természetesen eltérő értékekkel. Utoljára egy 'kor' származtatott tulajdonság is megtalálható, ami a relációs modellnél nem lesz elérhető, hiszen ennek az értéke később kiszámítható lesz az egyedben megtalálható 'szulDatum' tulajdonsággal.

## Az adatbázis ER modellje:



## Az adatbázis konvertálása relációs modellre:

Balról jobbra haladva a **'Termékek'** egyednél három normális tulajdonság és egy elsődleges kulcs található, így nincs szükség egynél több táblára a kialakításhoz.

A vele kapcsolatban lévő **'Webshop'** egyed N:M kapcsolatban van, így szükség van egy kapcsoló táblára, amelyben két idegen kulcs mező tartja a kapcsolatot, plusz egy a **'darabszám'** attribútum is ide fog kerülni.

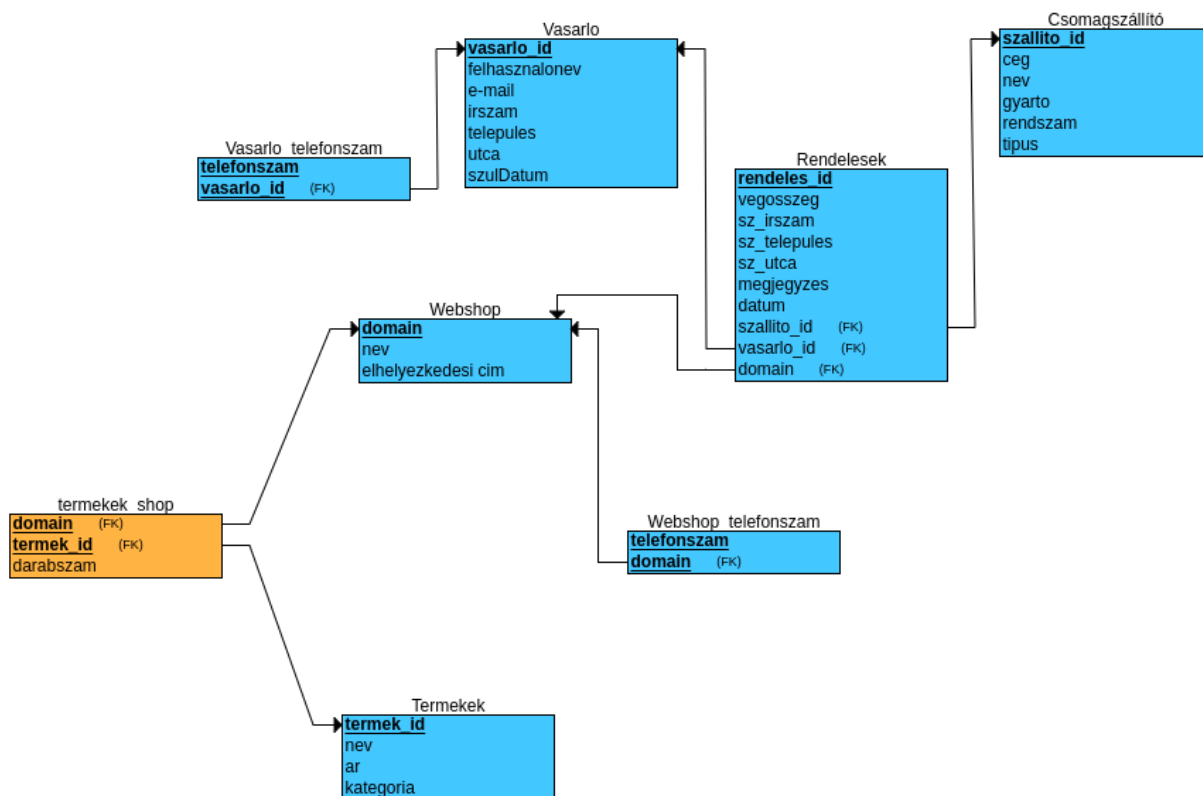
A **'Webshop'** egyednél kettő normális, egy elsődleges kulcs, és egy többértékű tulajdonság található, így a **'telefonszam'**-on kívül, mindegyik attribútum a **'Webshop'** táblába kerül egy újabb kulccsal együtt. A **'telefonszam'** tulajdonság egy külön táblába fog kerülni **'Webshop\_telefonszam'** néven.

A **'Rendelések'** egyedből ugyanúgy egy tábla lesz, de mivel a **'szállítási cím'** egy összetett tulajdonság, így ez a mező kimarad, és csak a többi kisebb tulajdonság kerül be a táblába.

A **'Vásárló'** ágnál hasonló a helyzet, de itt az összetett tulajdonság és a többértékű tulajdonság mellett, még egy származtatott tulajdonság is jelen van. Ezért a **'kor'** mező el lesz távolítva modellből, a **'telefonszam'** külön táblába fog kerülni **'Vasarlo\_telefonszam'** néven, miközben a Vásárló táblában a **'lakcím'** tulajdonság elvész, de azon leszármaztatott tulajdonságok megmaradnak.

A **'Csomagszállító'** egyednél ugyanaz a helyzet, mint a **'Rendelések'** egyednél, úgyhogy a **'jarmu'** összetett tulajdonság kimarad, az azon leszármaztatott mezők kivételével.

## Az adatbázis relációs modellje:



## Az adatbázis relációs sémája:

Termékek [ termek\_id, nev, ar, kategoria ]

termekek\_shop [ domain, termék\_id, darabszam ]

Webshop [ domain, nev, elhelyezkedesi cim ]

Webshop\_telefonszam [ telefonszam, domain ]

Rendelesek [ rendeles\_id, vegosszeg, sz\_irszam, sz\_telepules, sz\_utca, megjegyzes, datum, szallito\_id, vasarlo\_id, domain ]

Vasarlo [ vasarlo\_id, felhasznalonev, e-mail, irszam, telepules, utca, szulDatum ]

Vasarlo\_telefonszam [ telefonszam, vasarlo\_id ]

Csomagszállító [ szallito\_id, ceg, nev, gyarto, rendszam, tipus ]

## A táblák létrehozása:

A táblák létrehozásánál ügyelni kell a sorrendre, hogy először az idegen kulcs nélküli táblák készüljenek el, majd sorra jöhetnek azok, amelyek már tartalmaznak idegen kulcsot problémamentesen.

```
CREATE TABLE Vasarlo
(
    felhasznalonev varchar(25) NOT NULL,
    e-mail varchar(50) NOT NULL,
    vasarlo_id number,
    irszam number(4),
    telepules varchar(40),
    utca varchar(80),
    szulDatum DATE,
    PRIMARY KEY (vasarlo_id)
);
```

```
CREATE TABLE Termekek
(
    termék_id number,
    nev varchar(40),
    ar number,
    kategoria varchar(40),
    PRIMARY KEY (termek_id)
);
```

CREATE TABLE Csomagszallito

```
(
    szallito_id number,
    ceg varchar(10),
    nev varchar(30),
    gyarto varchar(10),
    rendszam varchar(7),
    tipus varchar(20),
    PRIMARY KEY (szallito_id)
);
```

CREATE TABLE Webshop

```
(
    nev varchar(20),
    domain varchar(30),
    elhelyezkedesi_cim varchar(80),
    PRIMARY KEY (domain)
);
```

CREATE TABLE Rendelesek

```
(
    rendeles_id number,
    vegosszeg number,
    sz_irszam number(4),
    sz_telepules varchar(40),
    sz_utca varchar(80),
    megjegyzes varchar(1000),
    datum date,
    domain varchar(30),
    szallito_id number,
    vasarlo_id number,
    PRIMARY KEY (rendeles_id),
    FOREIGN KEY (domain) REFERENCES Webshop(domain),
    FOREIGN KEY (szallito_id) REFERENCES Csomagszallito(szallito_id),
    FOREIGN KEY (vasarlo_id) REFERENCES Vasarlo(vasarlo_id)
);
```

CREATE TABLE termekek\_shop

```
(
    darabszam number,
    termék_id number,
    domain varchar(30),
    FOREIGN KEY (termék_id) REFERENCES Termekek(termék_id),
    FOREIGN KEY (domain) REFERENCES Webshop(domain)
);
```

CREATE TABLE Vasarlo\_telefonszam

```
(
    telefonszam varchar(15),
    vasarlo_id number,
    FOREIGN KEY (vasarlo_id) REFERENCES Vasarlo(vasarlo_id)
);
```

```
CREATE TABLE Webshop_telefonszam
(
    telefonszam varchar(15),
    domain varchar(30),
    FOREIGN KEY (domain) REFERENCES Webshop(domain)
);
```

## A táblák feltöltése:

```
INSERT INTO Vasarlo VALUES('csikocska30', 'csikocska30@gmail.com',1, 3530,
'Miskolc', 'Tátra utca 3.', '12.13.1996');
INSERT INTO Vasarlo VALUES('zambokiraly', 'jimmy3555@gmail.com',2, 5624,
'Doboz', 'Arany János utca 24.', '06.15.2000');
INSERT INTO Vasarlo VALUES('BIOhunter25', 'bio.hunter25@c2.hu',3, 5520,
'Szeghalom', 'Petőfi Sándor utca 50.', '04.10.1998');
INSERT INTO Vasarlo VALUES('SuperDrBubo', 'bubo.doktor30@gmail.com',4, 1061,
'Budapest', 'Jókai utca 15.', '12.03.1999');
INSERT INTO Vasarlo VALUES('Pet5050', 'peter.hajdu5050@hotmail.com',5, 9400,
'Sopron', 'Vörösbegy utca 65.', '04.28.2001');
```

```
INSERT INTO Termek VALUES (1,'porszívó',27990,'haztartasi');
INSERT INTO Termek VALUES (2,'televízió',52900,'szorakozas');
INSERT INTO Termek VALUES (3,'tornalabda',690,'sport');
INSERT INTO Termek VALUES (4,'páraelszívó',31290,'haztartasi');
INSERT INTO Termek VALUES (5,'hangprojektor',34990,'szorakozas');
INSERT INTO Termek VALUES (6,'DVD lejátszó',20090,'szorakozas');
INSERT INTO Termek VALUES (7,'akkus_fűró',32490,'barkacs');
INSERT INTO Termek VALUES (8,'tricikli',24500,'kulteri');
```

```
INSERT INTO Csomagszallito VALUES(1, 'GLS', 'Béla', 'Fiat', 'EVY-386', 'Ducato');
INSERT INTO Csomagszallito VALUES(2, 'GLS', 'Viktor', 'Fiat', 'MDO-130', 'Ducato');
INSERT INTO Csomagszallito VALUES(3, 'DPD', 'Gergely', 'Citroen', 'PLA-075',
'Jumper');
INSERT INTO Csomagszallito VALUES(4, 'MPL', 'András', 'Volkswagen', 'NLE-327',
'Crafter');
INSERT INTO Csomagszallito VALUES(5, 'GLS', 'Endre', 'Opel', 'KQE-247', 'Vivaro');
INSERT INTO Csomagszallito VALUES(6, 'MPL', 'Dénes', 'Fiat', 'NDC-973', 'Ducato');
```

```
INSERT INTO Webshop VALUES('eMAG', 'www.emag.hu', 'Székesfehérvár');
INSERT INTO Webshop VALUES('Alza', 'www.alza.hu', 'Budapest');
INSERT INTO Webshop VALUES('Auchan', 'www.auchan.hu', 'Miskolc');
```

```
INSERT INTO Webshop_telefonszam VALUES('+36-1-123-4567', 'www.emag.hu');
INSERT INTO Webshop_telefonszam VALUES('+36-1-987-6543', 'www.alza.hu');
INSERT INTO Webshop_telefonszam VALUES('+36-1-222-2222', 'www.auchan.hu');
```

```
INSERT INTO termekek_shop VALUES('www.emag.hu', 1, 12);
INSERT INTO termekek_shop VALUES('www.alza.hu', 2, 9);
INSERT INTO termekek_shop VALUES('www.auchan.hu', 3, 26);
INSERT INTO termekek_shop VALUES('www.emag.hu', 4, 10);
INSERT INTO termekek_shop VALUES('www.alza.hu', 5, 4);
INSERT INTO termekek_shop VALUES('www.auchan.hu', 6, 11);
INSERT INTO termekek_shop VALUES('www.auchan.hu', 7, 8);
INSERT INTO termekek_shop VALUES('www.emag.hu', 8, 2);
```

```
INSERT INTO Vasarlo_telefonszam VALUES('+36-20-1234567',1);
INSERT INTO Vasarlo_telefonszam VALUES('+36-70-7654321',2);
INSERT INTO Vasarlo_telefonszam VALUES('+36-20-1233210',3);
INSERT INTO Vasarlo_telefonszam VALUES('+36-30-5656567',4);
INSERT INTO Vasarlo_telefonszam VALUES('+36-30-1111111',5);
```

```
INSERT INTO Rendelesek VALUES(1, 9990, 3530, 'Miskolc', 'Tátra utca 3.', 'A
kaputelefon nem működik. Kapukód: xxxx', '11/20/2021', 'www.alza.hu', 2, 1);
INSERT INTO Rendelesek VALUES(2, 31290, 1061, 'Budapest', 'Jókai utca 15.', null,
'11/19/2021', 'www.emag.hu', 1, 4);
INSERT INTO Rendelesek VALUES(3, 20090, 5624, 'Doboz', 'Arany János utca 24.',
'Hétfőn a szomszéd tudja átvenni', '10/25/2021', 'www.auchan.hu', 5, 2);
INSERT INTO Rendelesek VALUES(4, 2070, 3881, 'Abaújszántó', 'Varga utca 44.', null,
'11/25/2021', 'www.emag.hu', 1, 5);
INSERT INTO Rendelesek VALUES(5, 32490, 3597, 'Hejőkeresztúr', 'Petőfi Sándor utca
67.', null, '11/24/2021', 'www.auchan.hu', 4, 3);
INSERT INTO Rendelesek VALUES(6, 52900, 9400, 'Sopron', 'Vörösbegy utca 65.',
null, '11/25/2021', 'www.alza.hu', 6, 5);
INSERT INTO Rendelesek VALUES(7, 27990, 5520, 'Szeghalom', 'Petőfi Sándor utca
50.', 'Délelőtt munkahelyemen vagyok, 2-kor vagyok elérhető.', '11/22/2021',
'www.emag.hu', 2, 3);
```



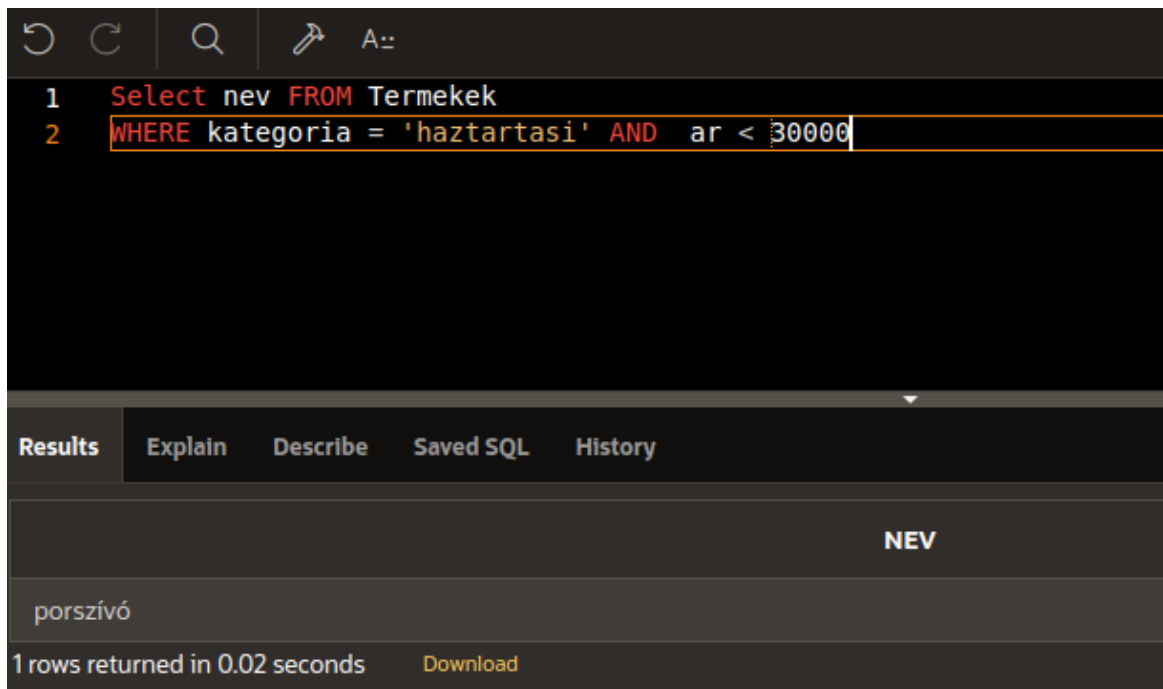
## Lekérdezések:

*1. Kérdezzük le azon termékek nevét, amelyeknek ára 30.000-nél kisebb és háztartási kategóriában van:*

```
SELECT nev FROM Termekek  
WHERE kategoria = 'háztartasi' AND ar < 30000;
```

$\pi$  nev

( $\sigma$  kategoria = "háztartasi" AND ar < 30000 termek)



```
1 Select nev FROM Termekek  
2 WHERE kategoria = 'háztartasi' AND ar < 30000
```

NEV
porszívó

1 rows returned in 0.02 seconds [Download](#)

*2. Minden egyes elérhető terméket számoljunk össze kategória szerint a Termékek táblából:*

```
SELECT kategoria, count(kategoria) "TERMÉKEK SZÁMA"  
FROM Termekek  
GROUP BY kategoria;
```

$\pi$  kategoria, count(kategoria)  $\rightarrow$  TERMÉKEK SZÁMA

( $\gamma$  kategoria, COUNT (kategoria) termek)



```
1 SELECT kategoria, count(kategoria) "TERMÉKEK SZÁMA"  
2 FROM Termekek  
3 GROUP BY kategoria
```

KATEGORIA	TERMÉKEK SZÁMA
sport	1
barkacs	1
kulteri	1
háztartasi	2
szórakozás	3

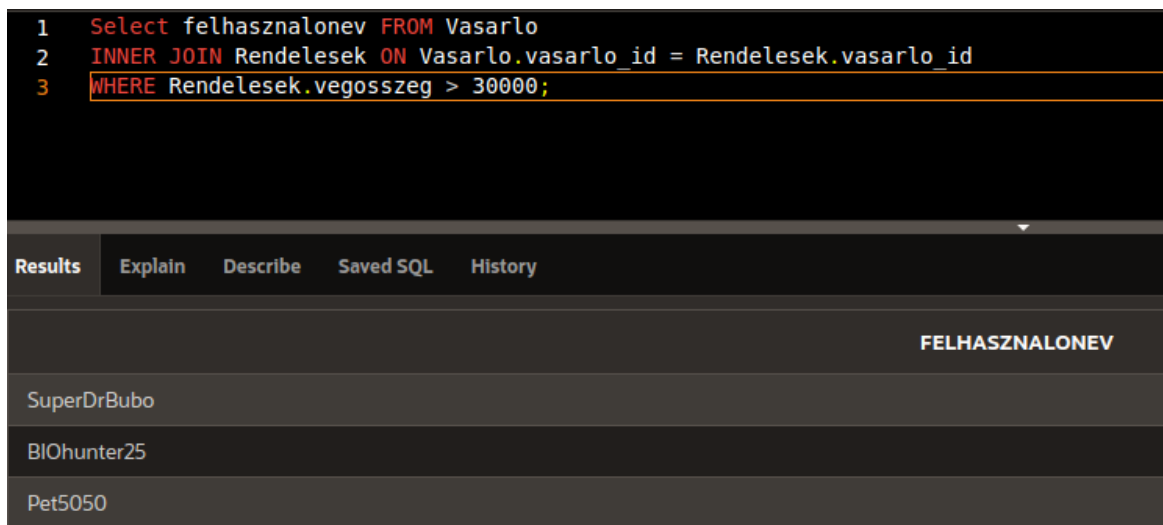
5 rows returned in 0.03 seconds [Download](#)

**3. Kérdezzük le azon vásárlók felhasználónevét, akik több mint 30.000-et fizettek a rendelésükért:**

```
SELECT felhasznalonev FROM Vasarlo  
INNER JOIN Rendelesek ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id  
WHERE Rendelesek.vegosszeg > 30000;
```

$\pi$  felhasznalonev

$(\sigma_{\text{rendelesek.vegosszeg} > 30000} (\text{vasarlo} \bowtie_{\text{vasarlo.vasarlo\_id} = \text{rendelesek.vasarlo\_id}} \text{rendelesek}))$



The screenshot shows a SQL query execution interface. The query is as follows:

```
1 Select felhasznalonev FROM Vasarlo  
2 INNER JOIN Rendelesek ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id  
3 WHERE Rendelesek.vegosszeg > 30000;
```

Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing a table with the following data:

FELHASZNALONEV
SuperDrBubo
BlOhunter25
Pet5050

**4. Kérdezzük le azon vásárlók felhasználónevét, akik rendelésükhöz írtak megjegyzést:**

```
SELECT felhasznalonev FROM Vasarlo  
JOIN Rendelesek ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id  
WHERE Rendelesek.megjegyzes IS NOT NULL;
```

$\pi$  felhasznalonev

$(\sigma_{\text{NOT}(\text{rendelesek.megjegyzes} = \text{NULL})} (\text{vasarlo} \bowtie_{\text{vasarlo.vasarlo\_id} = \text{rendelesek.vasarlo\_id}} \text{rendelesek}))$

1	Select felhasználonev FROM Vasarlo
2	JOIN Rendelesek ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id
3	WHERE Rendelesek.megjegyzes IS NOT NULL;

Results	Explain	Describe	Saved SQL	History
FELHASZNALONEV				
csikocska30				
zambokiraly				
BIOhunter25				

5. Kérdezzük le minden egyes Webshop nevét, és elérhető termékek darabszámának összegét:

```
SELECT nev, SUM(termekek_shop.darabszam) "OSSZES DARAB" FROM Webshop
JOIN termekek_shop ON Webshop.domain = termekek_shop.domain
GROUP BY nev
```

$\pi$  nev, SUM (darabszam)  $\rightarrow$  OSSZES DARAB

$(\gamma$  nev, SUM (darabszam) (webshop  $\bowtie$  webshop.domain = termekek\_shop.domain  
termekek\_shop))

```
1 SELECT nev, SUM(termekek_shop.darabszam) "OSSZES DARAB" FROM Webshop
2 JOIN termekek_shop ON Webshop.domain = termekek_shop.domain
3 GROUP BY nev
```

Results

ExplainDescribeSaved SQLHistory

NEV	OSSZES DARAB
eMAG	24
Auchan	45
Alza	13

**6. Írjuk ki azon vásárlók felhasználónevét és azonosítóját, akik csomagjait az MPL futárcég szállítja ki:**

```
SELECT Rendelesek.vasarlo_id, Vasarlo.felhasznalonev
FROM Rendelesek
INNER JOIN Csomagszallito ON Rendelesek.szallito_id = Csomagszallito.szallito_id
INNER JOIN Vasarlo ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id
WHERE Csomagszallito.ceg = 'MPL';
```

$\pi$  rendelesek.vasarlo\_id, vasarlo.felhasznalonev

$(\sigma_{\text{csomagszallito.ceg} = \text{"MPL"}}(\text{rendelesek} \bowtie \text{rendelesek.szallito\_id} = \text{csomagszallito.szallito\_id}$   
 $\text{csomagszallito} \bowtie \text{vasarlo.vasarlo\_id} = \text{rendelesek.vasarlo\_id} \text{ vasarlo}))$

```
1 SELECT Rendelesek.vasarlo_id, Vasarlo.felhasznalonev
2 FROM Rendelesek
3 INNER JOIN Csomagszallito ON Rendelesek.szallito_id = Csomagszallito.szallito_id
4 INNER JOIN Vasarlo ON Vasarlo.vasarlo_id = Rendelesek.vasarlo_id
5 WHERE Csomagszallito.ceg = 'MPL'
6
```

VASARLO_ID	FELHASZNALONEV
3	BIOhunter25
5	Pet5050

2 rows returned in 0.06 seconds Download

**7. Kérdezzük le a vásárlók életkorát:**

```
SELECT felhasznalonev, floor((sysdate-szulDatum)/365) "KOR"
FROM Vasarlo
```

$\pi$  felhasznalonev, floor((sysdate-szulDatum)/365)  $\rightarrow$  KOR Vasarlo

```
1 SELECT felhasznalonev, floor((sysdate-szulDatum)/365) "KOR" FROM Vasarlo
2
```

FELHASZNALONEV	KOR
csikocska30	24
zambokiraly	21
BIOhunter25	23
SuperDrBubo	22
Pet5050	20

5 rows returned in 0.00 seconds Download

**8. Kérdezzük le annak a vásárlónak az állandó utcáját, akinek a rendelésének a végösszege a legnagyobb volt, és 30-as telefon körzetszámmal rendelkezik:**

```
SELECT utca FROM Vasarlo
JOIN Vasarlo_telefonszam ON Vasarlo.vasarlo_id = Vasarlo_telefonszam.vasarlo_id
INNER JOIN Rendelesek ON Rendelesek.vasarlo_id = Vasarlo.vasarlo_id
WHERE Telefonszam LIKE '+36-30%'
AND Rendelesek.vegosszeg = (SELECT MAX(vegosszeg) FROM Rendelesek);
```

$\pi$  utca

$(\sigma_{\text{telefonszam LIKE "+36-30\%" AND Rendelesek.vegosszeg = } \pi \text{ MAX (vegosszeg) } \gamma \text{ MAX (vegosszeg) rendelesek (vasarlo } \bowtie \text{ vasarlo.vasarlo\_id = vasarlo\_telefonszam.vasarlo\_id vasarlo\_telefonszam } \bowtie \text{ rendelesek.vasarlo\_id = vasarlo.vasarlo\_id rendelesek))}$

The screenshot shows a SQL query execution interface. The query is as follows:

```
1 SELECT utca FROM Vasarlo
2 JOIN Vasarlo_telefonszam ON Vasarlo.vasarlo_id = Vasarlo_telefonszam.vasarlo_id
3 JOIN Rendelesek ON Rendelesek.vasarlo_id = Vasarlo.vasarlo_id
4 WHERE Telefonszam LIKE '+36-30%'
5 AND Rendelesek.vegosszeg = (SELECT MAX(vegosszeg) FROM Rendelesek);
```

The interface has tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing a single result:

UTCA
Vörösbegy utca 65.

At the bottom, it indicates "1 rows returned in 0.02 seconds" and provides a "Download" link.

**9. Kérdezzük le annak a vásárlónak az irányítószámát és településének nevét, akinek a rendelésének a végösszege a legalacsonyabb volt, és a GLS szállítja ki a csomagját:**

```
SELECT irszam, telepules FROM Vasarlo
JOIN Rendelesek ON Rendelesek.vasarlo_id = Vasarlo.vasarlo_id
JOIN Csomagszallito ON Rendelesek.szallito_id = Csomagszallito.szallito_id
WHERE Csomagszallito.ceg = 'GLS'
AND Rendelesek.vegosszeg = (SELECT MIN(vegosszeg) FROM Rendelesek);
```

$\pi$  irszam, telepules

$(\sigma_{\text{csomagszallito.ceg = "GLS" AND rendelesek.vegosszeg = } \pi \text{ MIN (vegosszeg) } \gamma \text{ MIN (vegosszeg) rendelesek (vasarlo } \bowtie \text{ rendelesek.vasarlo\_id = vasarlo.vasarlo\_id rendelesek } \bowtie \text{ rendelesek.szallito\_id = csomagszallito.szallito\_id csomagszallito))}$

1	SELECT irszam, telepules FROM Vasarlo
2	JOIN Rendelesek ON Rendelesek.vasarlo_id = Vasarlo.vasarlo_id
3	JOIN Csomagszallito ON Rendelesek.szallito_id = Csomagszallito.szallito_id
4	WHERE Csomagszallito.ceg = 'GLS'
5	AND Rendelesek.vegosszeg = (SELECT MIN(vegosszeg) FROM Rendelesek);

Results	Explain	Describe	Saved SQL	History
IRSZAM		TELEPULES		
9400		Sopron		

1 rows returned in 0.01 seconds [Download](#)

10. Írjuk ki a Webshopokon lévő aktív rendelések végösszegének szummáját, abban az esetben, ha több vagy egyenlő, mint 50.000:

```
SELECT Webshop.nev, sum(vegosszeg) FROM Rendelesek
JOIN Webshop ON Rendelesek.domain = Webshop.domain
GROUP BY nev
HAVING sum(vegosszeg) >= 50000;
```

$\pi$  webshop . nev, SUM (vegosszeg)

$(\sigma_{\text{SUM (vegosszeg) } \geq 50000} \gamma_{\text{nev, SUM (vegosszeg) (rendelesek} \bowtie \text{rendelesek.domain = webshop.domain webshop)})$

1	SELECT Webshop.nev, sum(vegosszeg) FROM Rendelesek
2	JOIN Webshop ON Rendelesek.domain = Webshop.domain
3	GROUP BY nev
4	HAVING sum(vegosszeg) >= 50000;

Results	Explain	Describe	Saved SQL	History
NEV		SUM(VEGOSSZEG)		
eMAG		61350		
Auchan		52580		
Alza		62890		

3 rows returned in 0.00 seconds [Download](#)