

JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves feladat

Éttermek

Készítette: **Nyíri Levente**

Neptunkód: **F023QC**

Dátum: **2022.11.20. Miskolc**

Tartalomjegyzék:

1. [A feladat leírása](#)
2. [Az adatbázis ER modellje](#)
3. [Az ER modell konvertálása relációs modellre](#)
4. [Az adatbázis relációs modellje](#)
5. [Az adatbázis relációs séma](#)
6. [Táblák létrehozása](#)
7. [Táblák feltöltése](#)
8. [Lekérdezések \(SQL és relációs algebra\)](#)

A feladat leírása:

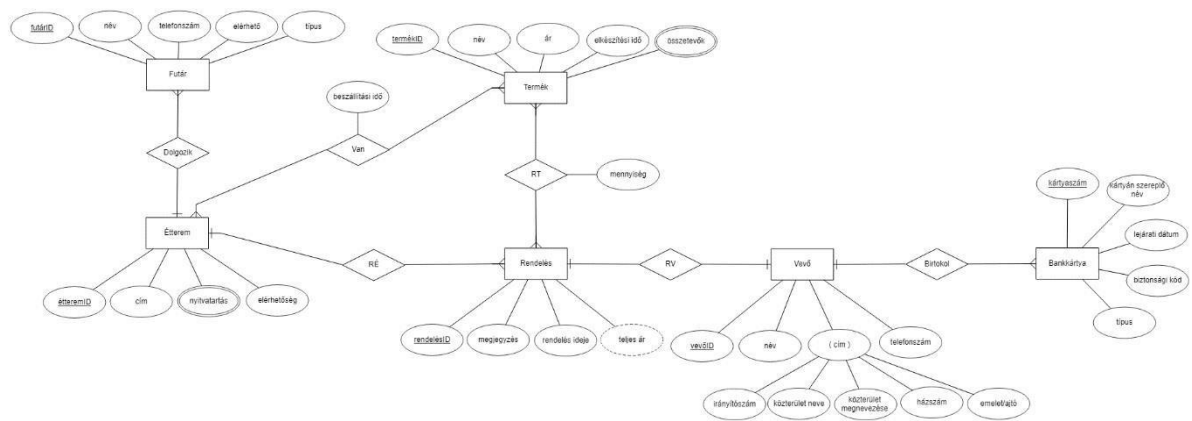
Adatbázisom egy étteremhálózatot dolgoz fel, melyben egyaránt szerepet kapnak az étterem futárjai is úgy, mint a vevők, a vevők bankkártyái, rendelései és ételei.

A szemantikai adatmodell 6 egyede:

- Étterem
- Futár
- Vevő
- Bankkártya
- Rendelés
- Termék

Az adatbázis ER modellje

Az **Étterem** egyed lesz a fő pont. Ide fog befutni a futár egyed egy 1:N kapcsolaton keresztül. Saját azonosítója az étteremID, lesz címe, elérhetősége és nyitva tartása. A nyitva tartás egy többértékű tulajdonság az ER modellben, így a relációs modellben önálló táblát fog kapni. A **Futár** egyed, ahogy már írtam, az Étterem egyed egy idegenkulcsát fogja tartalmazni, ezen kívül lesz egyedi azonosítója, a futárID, lesznek név, telefonszám, elérhető és típus tulajdonságai is. A **Vevő** tartalmaz vevőID-t, nevet, telefonszámot, illetve egy többágú tulajdonságot is, a címet. Ez szintén a relációs táblák készítésekor lesz fontos, ugyanis a cím elemei (az irányítószám, a közterület neve, a közterület megnevezése, a házszám és az emelet/ajtó tulajdonságok) külön-külön kapnak majd helyet a Vevő táblában. A **Bankkártya** tartalmaz egy kártyaszám tulajdonságot, ami az egyedi azonosítója lesz ennek az entitásnak. A kártyán szereplő név, a lejárat dátum, a biztonsági kód illetve a típus szintén elengedhetetlen tulajdonságok az egyedben. **Rendelés** egyedünkben szintén lesz egy rendelésID nevű tulajdonság, ami az elsődleges kulcs lesz, ezen kívül lesz egy származtatott tulajdonsága, a teljes ár. Többi tulajdonsága a megjegyzés és a rendelés ideje. Nem utolsó sorban a **Termék** egyedben is lesz egy elsődleges kulcs, név, ár, elkészítési idő és összetevők tulajdonságok. Az összetevők itt is többértékű lesz, ami külön táblát fog kapni a relációs adatmodellben. Fontos szót ejteni a kapcsolatokról is. Az Étterem és a Futár egyedek közt egy 1:N kapcsolat található, ugyanis egy étteremben több futár is dolgozik, viszont egy futár csak egy étteremben dolgozhat. Az Étterem és a Termék között jön az első N:M kapcsolat, melynek egyetlen tulajdonsága a szállítási idő. Egy étteremben több termék is található és egy termék több étteremben is megtalálható. A Termék és a Rendelés egyedek között található a második N:M kapcsolat, melynek szintén egyetlen tulajdonsága van, a mennyiség. Egy terméket többen is rendelhetnek és egy rendelésben több termék is lehet. A Rendelés és a Vevő között egy 1:1 kapcsolat található, ugyanis egy rendelés csak egy vevőhöz tartozhat és egy vevő csak egy rendelést adhat le egy időben. A Vevő és a Bankkártya egyedek között lesz az utolsó kapcsolat, ami 1:N. Egy vevőnek több bankkártyája lehet, de egy bankkártya csak egy vevőhöz tartozhat.



Az ER modell konvertálása relációs modellre

Haladjunk jobbról balra. Vesszük a Bankkártya egyedet, melynek az elsődleges kulcsa egy VARCHAR(20) lesz, a lejárat dátuma DATE típusú, a biztonsági kód INT(3), a kártyán szereplő név VARCHAR(30), típusa VARCHAR(15) és lesz egy idegenkulcsa is, a vevőID, ami INT. Az összes elem integritási feltétele NOT NULL.

A Vevő tábla következik, melynek elsődleges kulcsa INT. A név tulajdonság VARCHAR(30) és NOT NULL. A telefonszám VARCHAR(30) és lehet NULL, mivel nem biztos, hogy az adott vevőnek éppen lesz telefonkészüléke. Most jön a cím többágú tulajdonság, ami a relációs táblában szétszóródik több tulajdonságra. Az irányítószám INT(4) NOT NULL integritási feltétellel. A közterület neve VARCHAR(30) szintén NOT NULL, a közterület megnevezése VARCHAR(10) ebbe kevesebb karakternek kell beleférnie. NOT NULL lesz ő is, végül a házszám, ami egy INT értéket fog felvenni NOT NULL integritással és az emelet/ajtó, ami VARCHAR(5) és lehet NULL, mert van olyan vevő, aki kertesházban lakik.

A Futár táblában első helyen a futárID kap szerepet, ami az elsődleges kulcs lesz. Utána jön egy BOOLEAN tulajdonság, az elérhető. A név VARCHAR(30) értéket kapott, ahogy a telefonszám és a típus is. Egyetlen idegenkulcsa az étteremID lesz, ami INT. A telefonszámot kivéve mindegyik tulajdonság NOT NULL integritással rendelkezik.

Jön az Étterem tábla. PRIMARY KEY-je az étteremID, a cím VARCHAR(50) értéket vesz fel, mivel elég hosszú címeket is meg lehet adni az adatbázisban. Az elérhetőség pedig VARCHAR(100) lesz, ide pedig a telefonszámtól elkezdve a honlap domain nevén át mindenféle adatot be lehet írni. Természetesen mindegyik integritási feltétel NOT NULL. Az Étterem táblához tartozik az Étterem_nyitvatartás tábla is, ami az adott étterem nyitva tartását tartja nyilván. Ő neki VARCHAR(30) NOT NULL lesz a nyitvatartás oszlop értéke és lesz egy idegenkulcsa az étteremID, ami az Étterem táblára mutat.

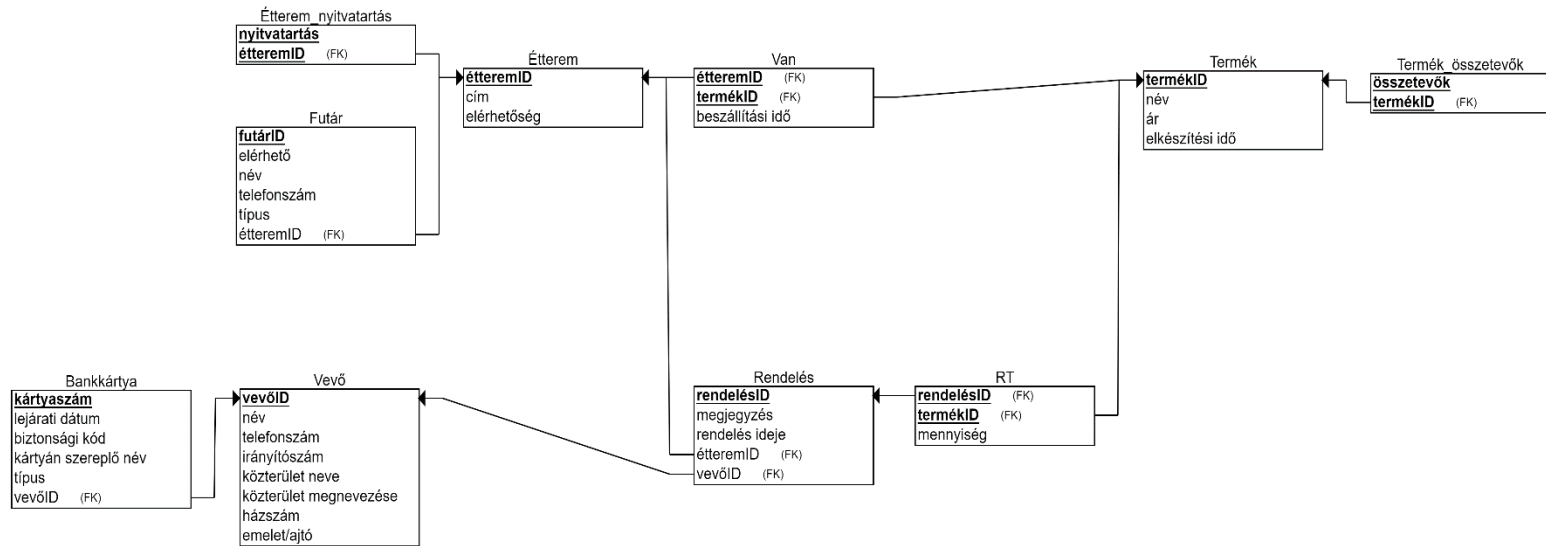
A Rendelés tábla jön, melynek szintén lesz elsődleges kulcsa a rendelésID formájában. A megjegyzés itt is egy hosszabb karakterláncot követel, így VARCHAR(100) lesz. Ez lehet NULL. A rendelés_ideje tulajdonság VARCHAR(30) NOT NULL értékkel fog bírni, illetve két idegenkulcsa lesz a táblának. Először az étteremID, ami az Étterem táblára mutat, majd a vevőID, ami a vevő táblára.

A Termék tábla termékID-ja lesz az elsődleges kulcs, ami INT, a név tulajdonság az étel neve, ami VARCHAR(50) és NOT NULL, az ár DECIMAL értéket fog felvenni és az elkészítési idő VARCHAR(15) lesz. A Termék táblához kapcsolódik a Termék_összetevők tábla is, ami a Termék egyed többértékű tulajdonsága. Neki két tulajdonsága lesz, az első a VARCHAR(30) értéket felvevő összetevők, amik akár lehetnek ugyan azok is, így nem lehet PRIMARY KEY, a második pedig a Termék tábla idegenkulcsa.

Végül, de nem utolsó sorban érkeznek az N:M kapcsolatokat kapcsolótáblái. Először az RT nevű kapcsolat jön, melynek két idegenkulcsa, az rendelésID és a termékID, mindkettő INT NOT NULL. Egyetlen saját tulajdonsága a mennyiség, ami VARCHAR(30) NOT NULL integritással.

A Van kapcsolat étteremID és termékID tulajdonságai lesznek az idegenkulcsok, a beszállítási_idő pedig VARCHAR(30) NOT NULL integritással bíró egyéni tulajdonság.

Az adatbázis relációs modellje



Az adatbázis relációs séma

Étterem [étteremID, cím, elérhetőség]

Étterem_nyitvatartás [nyitvatartás, étteremID]

Futár [futárID, elérhető, név, telefonszám, típus, étteremID]

Vevő [vevőID, név, telefonszám, irányítószám, közterület_neve, közterület_megnevezése, házsám, emelet_ajtó]

Bankkártya [kártyaszám, lejárat_i dátum, biztonsági_kód, kártyán_szereplő_név, típus, vevőID]

Rendelés [rendelésID, megjegyzés, rendelés_ideje, étteremID, vevőID]

Termék [termékID, név, ár, elkészítési_idő]

Termék_összetevők [összetevők, termékID]

RT [rendelésID, termékID, mennyiség]

Van [étteremID, termékID, beszállítási_idő]

A táblák létrehozása

A táblák SQL-beli implementálásánál a helyes sorrend számít. Nem lehet összevissza létrehozni a táblákat, ezért én a következő sorrendet állítottam fel az adatbázis létrehozása után:

- éttermek.étterem
- éttermek.étterem_nyitvatartás
- éttermek.futár
- éttermek.vevő
- éttermek.bankkártya
- éttermek.rendelés
- éttermek.termék
- éttermek.termék_összetevők
- éttermek.rt
- éttermek.van

Az SQL adatbázist a MySQL Workbench fejlesztői környezettel készítettem el.

```

DROP DATABASE IF EXISTS éttermek;
CREATE DATABASE éttermek;

DROP TABLE IF EXISTS éttermek.étterem;
CREATE TABLE éttermek.étterem(
    étteremID INT PRIMARY KEY,
    cím VARCHAR(50) NOT NULL,
    elérhetőség VARCHAR(100) NOT NULL
);

DROP TABLE IF EXISTS éttermek.étterem_nyitvatartás;
CREATE TABLE éttermek.étterem_nyitvatartás(
    nyitvatartás VARCHAR(30) PRIMARY KEY,
    étteremID INT NOT NULL,
    FOREIGN KEY (étteremID) REFERENCES éttermek.étterem(étteremID)
);

DROP TABLE IF EXISTS éttermek.futár;
CREATE TABLE éttermek.futár(
    futárID INT PRIMARY KEY,
    elérhető BOOLEAN NOT NULL,
    név VARCHAR(30) NOT NULL,
    telefonszám VARCHAR(30),
    típus VARCHAR(30) NOT NULL,
    étteremID INT NOT NULL,
    FOREIGN KEY (étteremID) REFERENCES éttermek.étterem(étteremID)
);

DROP TABLE IF EXISTS éttermek.vevő;
CREATE TABLE éttermek.vevő(
    vevőID INT PRIMARY KEY,
    név VARCHAR(30) NOT NULL,
    telefonszám VARCHAR(30),
    irányítószám INT(4) NOT NULL,
    község_neve VARCHAR(30) NOT NULL,
    község_megnevezése VARCHAR(10) NOT NULL,
    házszám INT NOT NULL,
    emelet_ajtó VARCHAR(5)
);

DROP TABLE IF EXISTS éttermek.bankkártya;
CREATE TABLE éttermek.bankkártya(
    kártyaszám VARCHAR(20) PRIMARY KEY,
    lejárat dátum DATE NOT NULL,
    biztonsági_kód INT(3) NOT NULL,
    kártyán_szerelő_név VARCHAR(30) NOT NULL,
    típus VARCHAR(15) NOT NULL,
    vevőID INT NOT NULL,
    FOREIGN KEY (vevőID) REFERENCES éttermek.vevő(vevőID)
);

```



```

DROP TABLE IF EXISTS éttermek.rendelés;
CREATE TABLE éttermek.rendelés(
    rendelésID INT PRIMARY KEY,
    megjegyzés VARCHAR(100),
    rendelés_ideje VARCHAR(30) NOT NULL,
    étteremID INT NOT NULL,
    vevőID INT NOT NULL,
    FOREIGN KEY (étteremID) REFERENCES éttermek.étterem(étteremID),
    FOREIGN KEY (vevőID) REFERENCES éttermek.vevő(vevőID)
);

DROP TABLE IF EXISTS éttermek.termék;
CREATE TABLE éttermek.termék(
    termékID INT PRIMARY KEY,
    név VARCHAR(50) NOT NULL,
    ár DECIMAL NOT NULL,
    elkészítési_idő VARCHAR(15) NOT NULL
);

DROP TABLE IF EXISTS éttermek.termékösszetevők;
CREATE TABLE éttermek.termékösszetevők(
    összetevők VARCHAR(30) PRIMARY KEY,
    termékID INT NOT NULL,
    FOREIGN KEY (termékID) REFERENCES éttermek.termék(termékID)
);

DROP TABLE IF EXISTS éttermek.RT;
CREATE TABLE éttermek.RT(
    rendelésID INT NOT NULL,
    termékID INT NOT NULL,
    mennyiség VARCHAR(30) NOT NULL,
    FOREIGN KEY (rendelésID) REFERENCES éttermek.rendelés(rendelésID),
    FOREIGN KEY (termékID) REFERENCES éttermek.termék(termékID)
);

DROP TABLE IF EXISTS éttermek.van;
CREATE TABLE éttermek.van(
    étteremID INT NOT NULL,
    termékID INT NOT NULL,
    beszállítási_idő VARCHAR(30) NOT NULL,
    FOREIGN KEY (étteremID) REFERENCES éttermek.étterem(étteremID),
    FOREIGN KEY (termékID) REFERENCES éttermek.termék(termékID)
);

```

A táblák feltöltése

```
1 INSERT INTO ettermek.etterem VALUES( 1, '3531 Miskolc, Györi kapu 45-47', 'Telefon: +3646428114, tokajkifozde.hu' );
2 INSERT INTO ettermek.etterem VALUES( 2, '3433 Nyékládháza, Vasút utca 13', 'Telefon: +36203423888, etnapizza.hu' );
3 INSERT INTO ettermek.etterem VALUES( 3, '3910 Tokaj, Hősök tere 1', 'Telefon: +36702960530, tokajtaverna.hu' );
4 INSERT INTO ettermek.etterem VALUES( 4, '1203 Budapest, Lajtha László utca 2', 'Telefon: +36207770001, artpub.hu' );
5 INSERT INTO ettermek.etterem VALUES( 5, '3860 Encs, Petőfi Sándor út 57', 'Telefon: +3646587340, anyukammondta.hu' );
6 INSERT INTO ettermek.etterem VALUES( 6, '8230 Balatonfüred, Tagore sétány 1', 'Telefon: +3687580070, borcsaetterem.hu' );
7 SELECT * FROM ettermek.etterem;
```

étteremID	cím	elérhetőség
1	3531 Miskolc, Györi kapu 45-47	Telefon: +3646428114, tokajkifozde.hu
2	3433 Nyékládháza, Vasút utca 13	Telefon: +36203423888, etnapizza.hu
3	3910 Tokaj, Hősök tere 1	Telefon: +36702960530, tokajtaverna.hu
4	1203 Budapest, Lajtha László utca 2	Telefon: +36207770001, artpub.hu
5	3860 Encs, Petőfi Sándor út 57	Telefon: +3646587340, anyukammondta.hu
6	8230 Balatonfüred, Tagore sétány 1	Telefon: +3687580070, borcsaetterem.hu
NULL	NULL	NULL

```
1 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '11.45-22.30', 6 );
2 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '11.00-16.00', 1 );
3 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '11.00-17.00', 3 );
4 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '11.00-22.00', 2 );
5 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '12.00-24.00', 4 );
6 INSERT INTO ettermek.etterem_nyitvatartas VALUES( '12.00-22.00', 5 );
7 SELECT * FROM ettermek.etterem_nyitvatartas;
```

nyitvatartás	étteremID
11.00-16.00	1
11.00-22.00	2
11.00-17.00	3
12.00-24.00	4
12.00-22.00	5
11.45-22.30	6
NULL	NULL

```
INSERT INTO ettermek.futar VALUES( 1, false, 'Kovács István', '+3670444888', 'Molt', 2 );
INSERT INTO ettermek.futar VALUES( 2, true, 'Kis László', NULL, 'Foodpanda', 3 );
INSERT INTO ettermek.futar VALUES( 3, true, 'Mogyes György', '+3687565433', 'Molt', 6 );
INSERT INTO ettermek.futar VALUES( 4, false, 'Gulyás Szabolcs', '+361321111', 'feletöl', 4 );
INSERT INTO ettermek.futar VALUES( 5, false, 'Gulyás György', '+361999888', 'Foodexpress', 4 );
INSERT INTO ettermek.futar VALUES( 6, false, 'Kiss Bálint', '+3620456423', 'Cantain', 1 );
INSERT INTO ettermek.futar VALUES( 7, true, 'Széplaki Eszter', '+3630321123', 'Molt', 3 );
INSERT INTO ettermek.futar VALUES( 8, true, 'Virág Zsolt', '+3687126765', 'Dofalat', 5 );
INSERT INTO ettermek.futar VALUES( 9, false, 'Kerényi Zsófia', NULL, 'Foodpanda', 1 );
INSERT INTO ettermek.futar VALUES( 10, false, 'Ökrös Marianna', NULL, 'Dofalat', 5 );
SELECT * FROM ettermek.futar;
```

futárID	elérhető	név	telefonszám	típus	étteremID
1	0	Kovács István	+3670444888	Wolt	2
2	1	Kis László	NULL	Foodpanda	3
3	1	Medgyes György	+3687565433	Wolt	6
4	0	Gulyás Szabolcs	+361321111	Teletál	4
5	0	Gulyás György	+361999888	FoodExpress	4
6	0	Kiss Bálint	+3620456423	Cantain	1
7	1	Széplaki Eszter	+3630321123	Wolt	3
8	1	Virág Zsolt	+3687126765	Jófalat	5
9	0	Kerényi Zsófia	NULL	Foodpanda	1
10	0	Ókrös Marianna	NULL	Jófalat	5
NULL	NULL	NULL	NULL	NULL	NULL

```

INSERT INTO ettermek.vevő VALUES( 1, 'Gonda György', NULL, 3526, 'Mohostó', 'utca', 12, NULL );
INSERT INTO ettermek.vevő VALUES( 2, 'Kiss János', '+3620321999', 3911, 'Jánosik', 'tér', 32, '3/2' );
INSERT INTO ettermek.vevő VALUES( 3, 'Kerekes László', '+3670242998', 2533, 'Szabadság', 'útja', 55, NULL );
INSERT INTO ettermek.vevő VALUES( 4, 'Kékes Zsolt', NULL, 1230, 'Nagy Sándor József', 'utca', 21, 'Fsz/2' );
INSERT INTO ettermek.vevő VALUES( 5, 'György István', '+36467550203', 3534, 'Jósika', 'utca', 33, '1/3' );
INSERT INTO ettermek.vevő VALUES( 6, 'Bók Mária', '+3687953345', 2619, 'Örökösök', 'útja', 63, NULL );
INSERT INTO ettermek.vevő VALUES( 7, 'Szerencsés Veronika', NULL, 1133, 'Lakosok', 'kertje', 55, NULL );
INSERT INTO ettermek.vevő VALUES( 8, 'Virág László', '+361555666', 1345, 'Rózsadomb', 'utca', 63, NULL );
SELECT * FROM ettermek.vevő;

```

vevőID	név	telefonszám	irányítószám	község_neve	község_megnevezése	házsám	emelet_ajtó
1	Gonda György	NULL	3526	Mohostó	utca	12	NULL
2	Kiss János	+3620321999	3911	Jánosik	tér	32	3/2
3	Kerekes László	+3670242998	2533	Szabadság	útja	55	NULL
4	Kékes Zsolt	NULL	1230	Nagy Sándor József	utca	21	Fsz/2
5	György István	+36467550203	3534	Jósika	utca	33	1/3
6	Bók Mária	+3687953345	2619	Örökösök	útja	63	NULL
7	Szerencsés Veronika	NULL	1133	Lakosok	kertje	55	NULL
8	Virág László	+361555666	1345	Rózsadomb	utca	63	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

INSERT INTO ettermek.bankkártya VALUES( '1983-2345-2112-4433', '2022-11-11', 325, 'Kiss János', 'MasterCard', 2 );
INSERT INTO ettermek.bankkártya VALUES( '1983-1343-2212-1234', '2024-12-12', 555, 'Bók Mária', 'MasterCard', 6 );
INSERT INTO ettermek.bankkártya VALUES( '9877-1333-2213-4241', '2023-03-05', 423, 'Virág László', 'VisaCard', 8 );
INSERT INTO ettermek.bankkártya VALUES( '9877-4555-2333-1231', '2025-01-13', 455, 'Szerencsés Veronika', 'VisaCard', 7 );
INSERT INTO ettermek.bankkártya VALUES( '3112-1343-4556-2123', '2022-12-31', 566, 'Gonda György', 'PayPal', 1 );
INSERT INTO ettermek.bankkártya VALUES( '1983-1445-5233-1212', '2023-03-12', 666, 'Szerencsés Veronika', 'MasterCard', 7 );
INSERT INTO ettermek.bankkártya VALUES( '9877-1342-1234-4231', '2024-05-23', 121, 'György István', 'VisaCard', 5 );
INSERT INTO ettermek.bankkártya VALUES( '3112-4422-3124-2425', '2025-02-28', 142, 'Bók Mária', 'PayPal', 6 );
INSERT INTO ettermek.bankkártya VALUES( '1983-5323-2356-4231', '2022-12-02', 975, 'Kékes Zsolt', 'MasterCard', 4 );
SELECT * FROM ettermek.bankkártya;

```

kártyaszám	lejárati_dátum	biztonsági_kód	kártyán_szereplő_név	típus	vevőID
1983-1343-2212-1234	2024-12-12	555	Bók Mária	MasterCard	6
1983-1445-5233-1212	2023-03-12	666	Szerencsés Veronika	MasterCard	7
1983-2345-2112-4433	2022-11-11	325	Kiss János	MasterCard	2
1983-5323-2356-4231	2022-12-02	975	Kékes Zsolt	MasterCard	4
3112-1343-4556-2123	2022-12-31	566	Gonda György	PayPal	1
3112-4422-3124-2425	2025-02-28	142	Bók Mária	PayPal	6
9877-1333-2213-4241	2023-03-05	423	Virág László	VisaCard	8
9877-1342-1234-4231	2024-05-23	121	György István	VisaCard	5
9877-4555-2333-1231	2025-01-13	455	Szerencsés Veronika	VisaCard	7
NULL	NULL	NULL	NULL	NULL	NULL


```

INSERT INTO éttermek.rendelés VALUES( 1, NULL, '12.03', 1, 5 );
INSERT INTO éttermek.rendelés VALUES( 2, 'Laktózmentes tejjel kérném a kávét', '13.55', 6, 7 );
INSERT INTO éttermek.rendelés VALUES( 3, 'Kizárólag szójjával kérném az ételek elkészítését', '15.06', 4, 8 );
INSERT INTO éttermek.rendelés VALUES( 4, NULL, '15.45', 1, 1 );
INSERT INTO éttermek.rendelés VALUES( 5, NULL, '11.02', 2, 1 );
INSERT INTO éttermek.rendelés VALUES( 6, NULL, '20.00', 5, 3 );
INSERT INTO éttermek.rendelés VALUES( 7, 'Tofuval kérem helyettesíteni a köretet', '15.50', 3, 4 );
INSERT INTO éttermek.rendelés VALUES( 8, 'Rukkola salátát kérem mellőzzék a többi közül', '22.00', 6, 6 );
INSERT INTO éttermek.rendelés VALUES( 9, NULL, '13.29', 6, 2 );
INSERT INTO éttermek.rendelés VALUES( 10, NULL, '21.00', 2, 5 );
SELECT * FROM éttermek.rendelés;

```

rendelésID	megjegyzés	rendelés_ideje	étteremID	vevőID
1	NULL	12.03	1	5
2	Laktózmentes tejjel kérném a kávét	13.55	6	7
3	Kizárólag szójjával kérném az ételek elkészítését	15.06	4	8
4	NULL	15.45	1	1
5	NULL	11.02	2	1
6	NULL	20.00	5	3
7	Tofuval kérem helyettesíteni a köretet	15.50	3	4
8	Rukkola salátát kérem mellőzzék a többi közül	22.00	6	6
9	NULL	13.29	6	2
10	NULL	21.00	2	5
NULL	NULL	NULL	NULL	NULL

```

2 INSERT INTO éttermek.termék VALUES( 2, 'Rákóczi fatányéros', 12500, '60 perc' );
3 INSERT INTO éttermek.termék VALUES( 3, 'Dubarri csirkemell édes kávéval', 2599, '20 perc' );
4 INSERT INTO éttermek.termék VALUES( 4, 'Pisztráng roston', 3250, '30 perc' );
5 INSERT INTO éttermek.termék VALUES( 5, 'Libacomb vöröskáposztával', 5540, '40 perc' );
6 INSERT INTO éttermek.termék VALUES( 6, 'Paprikás krumpli, ahogy mi szeretjük', 3999, '15 perc' );
7 SELECT * FROM éttermek.termék;

```

termékID	név	ár	elkészítési_idej
1	Konfitált kacsacomb párolt káposztával	6000	30 perc
2	Rákóczi fatányéros	12500	60 perc
3	Dubarri csirkemell édes kávéval	2599	20 perc
4	Pisztráng roston	3250	30 perc
5	Libacomb vöröskáposztával	5540	40 perc
6	Paprikás krumpli, ahogy mi szeretjük	3999	15 perc
NULL	NULL	NULL	NULL

```

INSERT INTO éttermek.termék_összetevők VALUES( 'Burgonya', 6 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Rántott csirkemell', 2 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Vöröskáposzta', 5 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Fehérbor', 5 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Karfiol', 3 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Kacsaszír', 1 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Kacsacomb', 1 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Pisztráng', 4 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Hal fűszerek', 4 );
INSERT INTO éttermek.termék_összetevők VALUES( 'Kolbász', 6 );
SELECT * FROM éttermek.termék_összetevők;

```

összetevők	termékID
Kacsacomb	1
Kacsaszír	1
Rántott csirkemell	2
Karfiol	3
Hal fűszerek	4
Pisztráng	4
Fehérbor	5
Vöröskáposzta	5
Burgonya	6
Kolbász	6
NULL	NULL

```

INSERT INTO éttermek.RT VALUES( 1, 4, '1 adag' );
INSERT INTO éttermek.RT VALUES( 2, 3, '1 adag' );
INSERT INTO éttermek.RT VALUES( 3, 1, '2 adag' );
INSERT INTO éttermek.RT VALUES( 4, 6, '1/2 adag' );
INSERT INTO éttermek.RT VALUES( 5, 2, '1/2 adag' );
INSERT INTO éttermek.RT VALUES( 6, 5, '2 adag' );
INSERT INTO éttermek.RT VALUES( 7, 2, '1 adag' );
INSERT INTO éttermek.RT VALUES( 8, 2, '1/2 adag' );
INSERT INTO éttermek.RT VALUES( 9, 1, '1 adag' );
INSERT INTO éttermek.RT VALUES( 10, 3, '1 adag' );
SELECT * FROM éttermek.RT;

```

rendelésID	termékID	mennyiség
1	4	1 adag
2	3	1 adag
3	1	2 adag
4	6	1/2 adag
5	2	1/2 adag
6	5	2 adag
7	2	1 adag
8	2	1/2 adag
9	1	1 adag
10	3	1 adag

```

INSERT INTO éttermek.van VALUES( 1, 6, '1 nap' );
INSERT INTO éttermek.van VALUES( 2, 4, '2 nap' );
INSERT INTO éttermek.van VALUES( 3, 5, '1 nap' );
INSERT INTO éttermek.van VALUES( 4, 2, '2 nap' );
INSERT INTO éttermek.van VALUES( 5, 3, '3 nap' );
INSERT INTO éttermek.van VALUES( 6, 1, '1 nap' );
SELECT * FROM éttermek.van;

```

étteremID	termékID	beszállítási_idej
1	6	1 nap
2	4	2 nap
3	5	1 nap
4	2	2 nap
5	3	3 nap
6	1	1 nap

Lekérdezések (SQL és relációs algebra)

1. Kérdezzük le annak a futárnak a nevét, aki a Cantain-ban dolgozik.

$$\pi_{\text{név}}(\sigma_{\text{típus} = \text{'Cantain'}}(\text{futár}))$$

```
SELECT név FROM éttermek.futár  
WHERE típus = 'Cantain';
```

név
Kiss Bálint

2. Kérdezzük le annak az étteremnek a címét, ahol a Gulyás nevűek dolgoznak!

$$\gamma(\pi_{\text{cím}}(\sigma_{\text{név LIKE 'Gulyás\%'}}(\text{étterem} \bowtie_{\text{fut.étteremID} = \text{ét.étteremID}} \text{futár})))$$

```
SELECT DISTINCT cím FROM éttermek.étterem ét  
JOIN éttermek.futár fut ON fut.étteremID = ét.étteremID  
WHERE fut.név LIKE 'Gulyás%';
```

cím
1203 Budapest, Lajtha László utca 2

3. Számoljuk meg azokat a felhasználókat, akiknek nem MasterCard bankkártyájuk van!

$$\pi_{\text{COUNT(vevőID)} \rightarrow \text{nem_mastercard_kártyák}}(\gamma_{\text{COUNT(vevőID)}}(\sigma_{\text{NOT (bank.típus != 'MasterCard')}}(\text{vevő} \bowtie_{\text{bank.vevőID} = \text{vev.vevőID}} \text{bankkártya})))$$

```
SELECT COUNT(vev.vevőID) AS Nem_MasterCard_kártyák FROM éttermek.vevő vev  
JOIN éttermek.bankkártya bank ON bank.vevőID = vev.vevőID  
WHERE bank.típus != 'MasterCard';
```

Nem_MasterCard_kártyák
5

4. Kérdezzük le melyik ételt tart a legtovább elkészíteni!

$$\pi_{\text{elkészítési_idő}} \downarrow (\pi_{\text{név termék}})$$

```
SELECT név FROM éttermek.termék
ORDER BY elkészítési_idő DESC
LIMIT 1;
```

név
Rákóczi fatányéros

5. Kérdezzük le azon termékek nevét, melyeknek az ára több mint az átlag!

$$\pi_{\text{név} \rightarrow \text{Átlagnál_nagyobb_ár}} (\gamma_{\text{név}} (\sigma_{(\text{AVG(ár)} > (\pi_{\text{AVG(ár)}} \text{ termék}))})$$

```
SELECT név, AVG(ár) AS Átlagnál_nagyobb_ár FROM éttermek.termék
GROUP BY név
HAVING AVG(ár) > (SELECT AVG(ár) FROM éttermek.termék);
```

név	Átlagnál_nagyobb_ár
Konfitált kacsacomb párolt káposztával	6000.0000
Rákóczi fatányéros	12500.0000

6. Kérdezzük le azon felhasználók adatait, akik nem rendelkeznek bankkártyával!

$$\sigma_{\text{vev.vevőID NOT IN } (\gamma_{\text{vevőID}}) \text{ bankkártya}}$$

```
SELECT * FROM éttermek.vevő vev
WHERE vev.vevőID NOT IN (SELECT DISTINCT vevőID FROM éttermek.bankkártya);
```

vevőID	név	telefonszám	irányítószám	közterület_neve	közterület_megnevezése	házzszám	emelet_ajtó
3	Kerekes László	+3670242998	2533	Szabadság	útja	55	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. Kérdezzük le annak az étteremnek az elérhetőségét, melynél a termék beszállítási ideje a legmagasabb!

$$\pi_{\text{van.beszállítási_idő}} \downarrow (\pi_{\text{elérhetőség}}(\text{étterem} \bowtie \text{van.étteremID} = \text{ét.étteremID} \text{ van}))$$

```
SELECT elérhetőség FROM éttermek.étterem ét
JOIN éttermek.van van ON van.étteremID = ét.étteremID
ORDER BY van.beszállítási_idő DESC
LIMIT 1;
```

elérhetőség
Telefon: +3646587340, anyukammonda.hu

8. Kérdezzük le annak a terméknek a nevét, amelyik étteremnek a weboldala: borcsaetterem.hu

$$\pi_{\text{név}}(\sigma_{\text{ét.elérhetőség LIKE 'borcsaetterem.hu\%'}}(\text{termék} \bowtie \text{van.termékID} = \text{term.termékID} \text{ van} \bowtie \text{ét.termékID} = \text{van.étteremID} \text{ étterem}))$$

```
SELECT név FROM éttermek.termék term
JOIN éttermek.van van ON van.termékID = term.termékID
JOIN éttermek.étterem ét ON ét.étteremID = van.étteremID
WHERE ét.elérhetőség LIKE '%borcsaetterem.hu%';
```

név
Konfitált kacsacomb párolt káposztával

9. Kérdezzük le azon vevők neveit és telefonszámait, akik kertesházban laknak és van telefonjuk!

$\pi_{\text{név, telefonszám}}(\sigma_{\text{emelet_ajtó} = \text{NULL AND NOT (telefonszám} = \text{NULL)}} \text{vevő})$

```
SELECT név, telefonszám FROM éttermek.vevő
WHERE emelet_ajtó IS NULL AND telefonszám IS NOT NULL;
```

név	telefonszám
Kerekes László	+3670242998
Bók Mária	+3687953345
Virág László	+361555666

10. Kérdezzük le azon vevők címét és rendelésének megjegyzését, aki megjegyzést írt a rendeléséhez!

$\pi_{\text{irányítószám, közterület_neve, közterület_megnevezése, házszám, emelet_ajtó, rend.megjegyzés}}(\sigma_{\text{NOT (rend.megjegyzés} = \text{NULL)}}(\text{vevő} \bowtie \text{rend.vevőID} = \text{vev.vevőID} \text{ rendelés}))$

```
SELECT irányítószám,
       közterület_neve,
       közterület_megnevezése,
       házszám,
       emelet_ajtó,
       rend.megjegyzés
FROM éttermek.vevő vev
JOIN éttermek.rendelés rend ON rend.vevőID = vev.vevőID
WHERE rend.megjegyzés IS NOT NULL;
```

irányítószám	község_neve	község_megnevezése	házzám	emelet_ajtó	megjegyzés
1230	Nagy Sándor József	utca	21	Fsz/2	Tofuval kérem helyettesíteni a köretet
2619	Örökösök	útja	63	NULL	Rukkola salátát kérem mellőzzék a többi közül
1133	Lakosok	kertje	55	NULL	Laktómentes tejjel kérném a kávé
1345	Rózsadomb	utca	63	NULL	Kizárólag szójával kérném az ételek elkészítését

11. Kérdezzük le azon éttermek címeit, ahol a futárok nem elérhetőek!

$\gamma_{\text{cím}}(\sigma_{\text{fut.elérhető} = \text{false}}(\text{étterem} \bowtie_{\text{fut.étteremID} = \text{ét.étteremID}} \text{futár}))$

```
SELECT cím FROM éttermek.étterem ét
JOIN éttermek.futár fut ON fut.étteremID = ét.étteremID
WHERE fut.elérhető = false
GROUP BY cím;
```

cím
3433 Nyékládháza, Vasút utca 13
1203 Budapest, Lajtha László utca 2
3531 Miskolc, Győri kapu 45-47
3860 Encs, Petőfi Sándor út 57

12. Kérdezzük le azon termék adatait, melynek a legtöbb az ára és a vevő féladagot kér belőle!

$\gamma(\pi_{\text{MAX(ár)} \rightarrow \text{maximum_ár, név}}(\gamma_{\text{MAX(ár)}}(\sigma_{\text{rt.mennyiség} = '1/2 \text{ adag}}'(\text{termék} \bowtie_{\text{rt.termékID} = \text{term.termékID}} \text{rt}))))$

```
SELECT DISTINCT MAX(ár) AS Maximum_ár, név FROM éttermek.termék term
JOIN éttermek.RT rt ON rt.termékID = term.termékID
WHERE rt.mennyiség = '1/2 adag';
```

Maximum_ár	név
12500	Paprikás krumpli, ahogy mi szeretjük

13. Kérdezze le, hogy létezik-e PayPal kártyával rendelkező felhasználó!

$$\pi_{\text{név}}(\sigma(\pi_{\text{típus}}(\sigma(c.\text{típus} = \text{'PayPal'}))))$$

```
SELECT név FROM éttermek.vevő  
WHERE EXISTS (SELECT típus FROM éttermek.bankkártya c WHERE c.típus = 'PayPal');
```

név
Gonda György
Kiss János
Kerekes László
Kékes Zsolt
György István
Bók Mária
Szerencsés Veronika
Virág László

14. Listázza ki az összes terméknevet!

$$\pi_{\text{név}}(\sigma_{(\text{TRUE})})$$

```
SELECT ALL név FROM éttermek.termék  
WHERE TRUE;
```

név
Konfitált kacsacomb párolt káposztával
Rákóczi fatányéros
Dubbari csirkemell édes kávéval
Pisztráng roston
Libacomb vöröskáposztával
Paprikás krumpli, ahogy mi szeretjük

15. Listázza ki bármely termék árát, amely több, mint 6000.

$$\pi_{\text{név}}(\sigma_{\text{termékID} = (\pi_{\text{termékID}}(\sigma_{\text{ár} > 6000})))$$

```
SELECT név
FROM éttermek.termék
WHERE termékID = ANY (SELECT termékID FROM éttermek.termék WHERE ár > 6000);
```

név
Rákóczi fatányéros