JEGYZŐKÖNYV

**Adatbázisrendszerek I.**

**Féléves feladat**

**Készítette: Pócsi Ákos**

**Neptunkód: ITUCXP**

**Gyakorlat ideje: Szerda 14-16**

**Gyakorlatvezető: Dr. Bednarik László**

**A feladat leírása**

A beadandó témája egy olyan adatbázis, amely egy élelmiszerüzletet kezel. Lekérdezhetjük a dolgozók, a termékek, a nagykerek, az üzlet, valamint a pontgyűjtő kártyával rendelkező vevők adatait is.

**A Nagyker egyed tulajdonságai**

Nagyker ID: A Nagyker egyed elsődleges kulcsa.

Név: A Nagyker neve.

Cím: Összetett tulajdonság, A nagyker címe.

Elérhetőség: Összetett tulajdonság, a nagyker telefonszáma, e-mail címe.

**A Termék egyed tulajdonságai**

Termék ID: A Termék egyed elsődleges kulcsa.

Ár: A termék beszerzési ára.

Bruttó ár: Származtatott tulajdonság, mert az árból kiszámolható.

Mennyiség: A termék darabszáma.

*A szállítás kapcsolat tulajdonsága a Dátum, mert több időpontban is érkeznek termékek.*

**A Nagyker és a Termék egyed N:M típusú kapcsolat, mert egy nagyker több terméket is szállíthat, egy terméket pedig több nagyker is szállíthat.**

**A Üzlet egyed tulajdonságai**

Üzlet ID: Az Üzlet egyed elsődleges kulcsa.

Név: Az üzlet neve.

Cím: Összetett tulajdonság, az üzlet címe.

Nyitvatartás: Az üzlet nyitvatartását tárolja.

Elérhetőség: Összetett tulajdonság, az üzlet telefonszáma, e-mail címe.

**Az Üzlet és a Termék egyed 1:N típusú kapcsolat, mert egy üzletbe több termék érkezhet, de egy termék csak egy üzletbe.**

**A Dolgozó egyed tulajdonságai**

Dolgozó ID: A Dolgozó egyed elsődleges kulcsa.

Név: A dolgozó neve.

Telefonszám: Több értékű tulajdonság, A dolgozó telefonszámai.

Lakcím: Összetett tulajdonság, a dolgozó lakcíme.

Adószám: A dolgozó adószáma.

Bankszámlaszám: A dolgozó bankszámlaszáma.

TB szám: A dolgozó TB száma.

**Vész esetén értesítendő egyed tulajdonságai**: Összetett tulajdonság, vész esetén értesítendő dolgozó hozzátartozóinak adatai.

**Az Üzlet és a Dolgozó egyed 1:N típusú kapcsolat, mert egy üzletben több alkalmazott dolgozik, de egy dolgozó csak egy üzletben.**

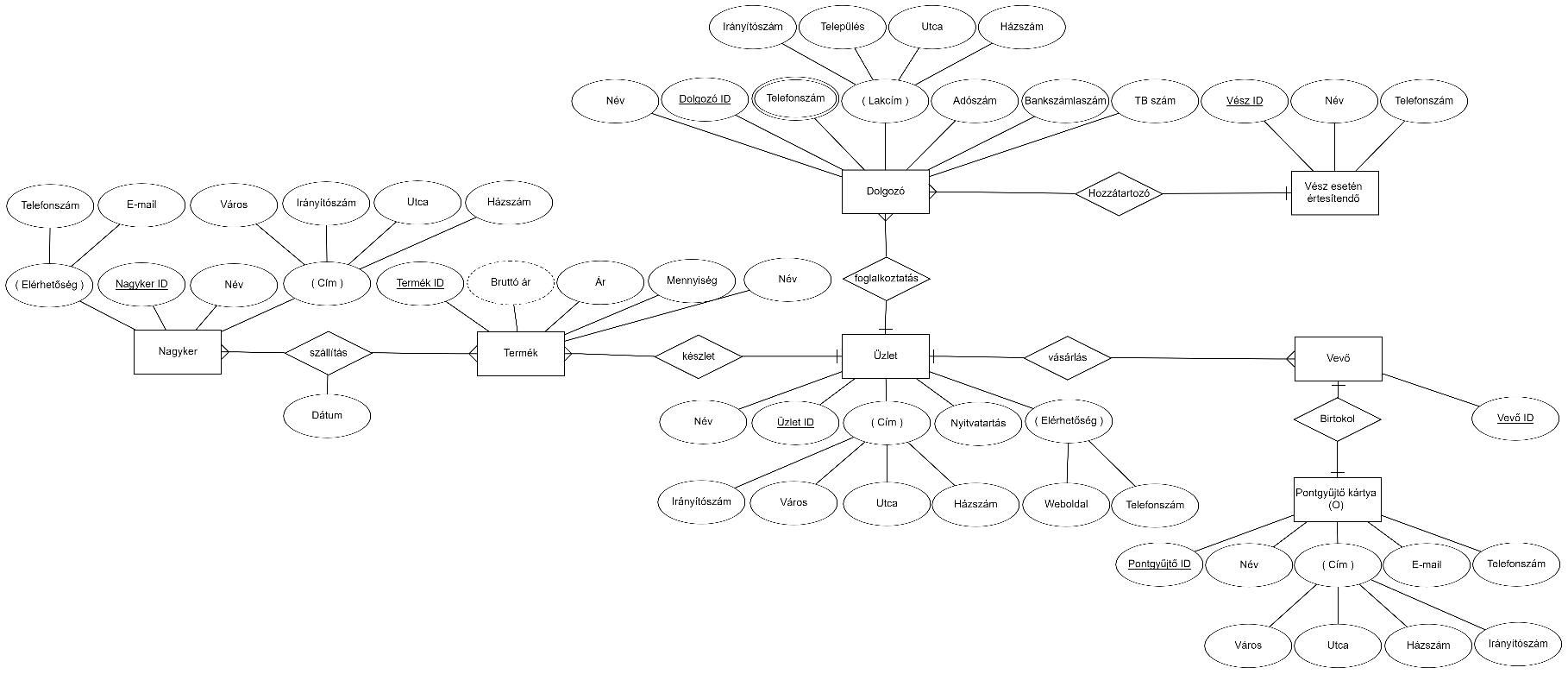
**A Vevő egyed tulajdonságai**

Vevő ID: A Vevő egyed elsődleges kulcsa

Pontgyűjtő kártya: Opcionális összetett tulajdonság, mert nem minden vevő rendelkezik vele.

Név, cím és e-mail adatokat tárol.

**Az Üzlet és a Vevő egyed 1:N típusú kapcsolat, mert egy üzletben több vásárló is vásárolhat, viszont egy vásárló egyszerre csak egy üzletben vásárolhat.**

**Az adatbázis ER-modellje**

**Adatbázis konvertálása Relációs modellre**

A **Nagyker** egyedből tábla lesz. Elsődleges kulcs a **Nagyker ID.** Tulajdonságai az *Irányítószám, Utca, Házszám, Város, Név, Telefonszám, E-mail.*

A **Termék** egyedből tábla lesz. Elsődleges kulcs a **Termék ID.** Tulajdonságai a *Mennyiség* és az *Ár.*

A **Nagyker** és a **Termék** táblák közötti kapcsolat egy segédtábla segítségével történik. A szállítás segédtáblában a két idegenkulcs, a **Nagyker ID**-ra és a **Termék ID**-ra mutat. Tulajdonsága a *Dátum* lesz.

Az **Üzlet** egyedből tábla lesz. Elsődleges kulcs az **Üzlet ID.** Három idegen kulcsa, az első a **Dolgozó ID**-re mutat a **Dolgozó** táblán, a második a **Vevő ID**-re mutat a **Vevő** táblán, a harmadik a **Termék ID**-re mutat a **Termék** táblán. Tulajdonságai a *Név, Weboldal,* *Telefonszám, Nyitvatartás, Irányítószám, Város, Utca és Házszám* mezők.

A **Dolgozó** egyedből tábla lesz. Elsődleges kulcs a **Dolgozó ID.** Tulajdonságai a *Név, Irányítószám, Település, Utca, Házszám, Adószám, Bankszámlaszám, TB szám, Név, Telefonszám.*

A **Dolgozó** táblára mutat két segédtábla. A **Dolgozó\_Telefonszám** segédtábla elsődleges kulcsa a **Telefonszám**. Az idegen kulcs a **Dolgozó ID**-re mutat. A **Vész esetén érrtesítendő** segédtábla idegen kulcsa szintén a **Dolgozó ID**-re mutat. Tulajdoságai a *Név* és a *Telefonszám* mezők.

A **Vevő** egyedből tábla lesz. Elsődleges kulcs a **Vevő ID.** Tulajdonságai a Név, Cím, E-mail.

A **Pontgyűjtő kártya** egyedből tábla lesz. Egyetlen hamis kulcs a **Vevő** tábla **Vevő ID**-re mutat. Tulajdonságai: *Név, Város, Utca, Házszám, Irányítószám, E-mail.*

**Relációs modell**

**Relációs séma**

**Nagyker [ Nagyker ID, Irányítószám, Utca, Házszám, Város, Név, Telefonszám, E-mail ]**

**szállítás [ Nagyker ID, Termék ID, Dátum ]**

**Pontgyűjtő kártya [Pontgyüjtő ID, Név, Város, Utca, Házszám, Irányítószám, E-mail, Vevő ID ]**

**Termék [ Termék ID, Mennyiség, Ár ]**

**Üzlet [ Üzlet ID, Név, Weboldal, Telefonszám, Nyitvatartás, Irányítószám, Város, Utca, Házszám, Dolgozó ID, Vevő ID, Termék ID ]**

**Dolgozó [ Dolgozó ID, Név, Irányítószám, Település, Utca, Házszám, Adószám, Bankszámlaszám, TB szám, Név, Telefonszám ]**

**Vevő [ Vevő ID, Név, Cím, E-mail ]**

**Dolgozó\_Telefonszám [ Telefonszám, Dolgozó ID ]**

**Vész esetén értesítendő [ Vész ID, Név, Telefonszám ]**

**A Táblák létrehozása**

CREATE TABLE Uzlet

(

Nev TINYTEXT NOT NULL,

Uzlet\_ID INT NOT NULL,

Weboldal TINYTEXT NOT NULL,

Telefonszam CHAR(12) NOT NULL,

Nyitvatartas TINYTEXT NOT NULL,

Iranyitoszam INT NOT NULL,

Varos TINYTEXT NOT NULL,

Utca TINYTEXT NOT NULL,

Hazszam TINYTEXT NOT NULL,

PRIMARY KEY (Uzlet\_ID)

);

CREATE TABLE Nagyker

(

Nagyker\_ID INT NOT NULL,

Iranyitoszam INT NOT NULL,

Utca TINYTEXT NOT NULL,

Hazszam TINYTEXT NOT NULL,

Varos TINYTEXT NOT NULL,

Nev TINYTEXT NOT NULL,

Telefonszam CHAR(12) NOT NULL,

Email TINYTEXT NOT NULL,

PRIMARY KEY (Nagyker\_ID)

);

CREATE TABLE Termek

(

Termek\_ID INT NOT NULL,

Mennyiseg INT NOT NULL,

Ar INT NOT NULL,

Uzlet\_ID INT NOT NULL,

PRIMARY KEY (Termek\_ID),

FOREIGN KEY (Uzlet\_ID) REFERENCES Uzlet(Uzlet\_ID)

);

CREATE TABLE Vevo

(

Vevo\_ID INT NOT NULL,

Uzlet\_ID INT NOT NULL,

PRIMARY KEY (Vevo\_ID),

FOREIGN KEY (Uzlet\_ID) REFERENCES Uzlet(Uzlet\_ID)

);

CREATE TABLE Pontgyujto\_kartya

(

Pontgyujto\_ID INT NOT NULL,

Nev TINYTEXT NOT NULL,

Varos TINYTEXT NOT NULL,

Utca TINYTEXT NOT NULL,

Hazszam TINYTEXT NOT NULL,

Iranyitoszam INT NOT NULL,

Email TINYTEXT NOT NULL,

Telefonszam CHAR(12) NOT NULL,

Vevo\_ID INT NOT NULL,

PRIMARY KEY (Pontgyujto\_ID),

FOREIGN KEY (Vevo\_ID) REFERENCES Vevo(Vevo\_ID)

);

CREATE TABLE Szallitas

(

Datum DATE NOT NULL,

Nagyker\_ID INT NOT NULL,

Termek\_ID INT NOT NULL,

FOREIGN KEY (Nagyker\_ID) REFERENCES Nagyker(Nagyker\_ID),

FOREIGN KEY (Termek\_ID) REFERENCES Termek(Termek\_ID)

);

CREATE TABLE Dolgozo

(

Nev TINYTEXT NOT NULL,

Dolgozo\_ID INT NOT NULL,

Iranyitoszam INT NOT NULL,

Telepules TINYTEXT NOT NULL,

Utca TINYTEXT NOT NULL,

Hazszam TINYTEXT NOT NULL,

Adoszam CHAR(13) NOT NULL,

Bankszamlaszam CHAR(26) NOT NULL,

TB\_szam CHAR(11) NOT NULL,

Uzlet\_ID INT NOT NULL,

PRIMARY KEY (Dolgozo\_ID),

FOREIGN KEY (Uzlet\_ID) REFERENCES Uzlet(Uzlet\_ID)

);

CREATE TABLE Vesz\_Eseten\_ertesitendo

(

Vesz\_ID INT NOT NULL,

Nev TINYTEXT NOT NULL,

Telefonszam CHAR(12) NOT NULL,

Dolgozo\_ID INT NOT NULL,

PRIMARY KEY (Vesz\_ID),

FOREIGN KEY (Dolgozo\_ID) REFERENCES Dolgozo(Dolgozo\_ID)

);

CREATE TABLE Dolgozo\_Telefonszam

(

Telefonszam CHAR(12) NOT NULL,

Dolgozo\_ID INT NOT NULL,

PRIMARY KEY (Telefonszam, Dolgozo\_ID),

FOREIGN KEY (Dolgozo\_ID) REFERENCES Dolgozo(Dolgozo\_ID)

);

**A táblák feltöltése**

INSERT INTO Uzlet VALUES ("Baziscsemege",1, "www.baziscsemege.hu","+36303254867","6-18","3565", "Tiszaluc", "Szechenyi ut", "12");

INSERT INTO Uzlet VALUES ("Forrascsemege",2, "www.forrascsemege.hu","+36701554897","8-16","3922", "Taktaharkany", "Deak utca", "5");

INSERT INTO Dolgozo VALUES ("Kis Pista", 1,"3529", "Miskolc", "Petofi utca", "12 2.emelet 1.ajto", "0123456789", "11112222-33334444-55556666", "111-333-555", 1);

INSERT INTO Dolgozo VALUES ("Kis Pista", 2,"3529", "Miskolc", "Petofi utca", "12 2.emelet 1.ajto", "0123456789", "11112222-33334444-55556666", "111-333-555", 2);

INSERT INTO Dolgozo VALUES ("Nagy Maria", 3,"3565", "Tiszaluc", "Arany Janos utca", "10", "9876543210", "99998888-77776666-55556666", "222-333-444", 2);

INSERT INTO Dolgozo VALUES ("Kovacs Andras", 4, "3922", "Taktaharkany", "Mezo utca", "5", "0246897531", "11112222-11110000-00000000", "123-234-555", 1);

INSERT INTO Dolgozo VALUES ("Kis Monika", 5,"3922", "Taktharkany", "Posta utca", "22", "0556816789", "12512752-33934547-57256876", "181-373-455", 1);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+36301325181", 1);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+3670123456", 1);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+36301111122", 2);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+3620141151", 2);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+36401235656", 3);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+3670123556", 3);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+36301327851", 4);

INSERT INTO Dolgozo\_Telefonszam VALUES ("+36705525178", 5);

INSERT INTO Vesz\_Eseten\_ertesitendo VALUES (1,"Kis Pista anyukaja", "+36301987654", 1);

INSERT INTO Nagyker VALUES (1, "3410", "Nagyker utca", "12", "Ebes", "Elelmiszernagyker", "+361123456", "elelmiszernagyker@gmail.com");

INSERT INTO Nagyker VALUES (2, "3550", "Tejes utca", "10", "Debrecen", "Tejtermek nagyker", "+361123453", "tejtermeknagyker@gmail.com");

INSERT INTO Nagyker VALUES (3, "3555", "Erdei utca", "20", "Nyiregyhaza", "Vegyes nagyker", "+3611233455", "vegyesnagyker@gmail.com");

INSERT INTO Termek VALUES (1, 1000, 250, 1, "Babkonzerv");

INSERT INTO Termek VALUES (2, 2000, 2550, 1, "Mososzer");

INSERT INTO Termek VALUES (3, 500, 125, 2, "Tejberizs");

INSERT INTO Termek VALUES (4, 600, 250, 2, "Kindertejszelet");

INSERT INTO Termek VALUES (5, 60, 2500, 1, "Kovaszosuborka");

INSERT INTO Termek VALUES (6, 800, 1050, 1, "Csirkemell");

INSERT INTO Szallitas VALUES ("2021-02-02", 1, 1);

INSERT INTO Szallitas VALUES ("2021-02-10", 1, 2);

INSERT INTO Szallitas VALUES ("2021-03-16", 1, 3);

INSERT INTO Szallitas VALUES ("2021-01-05", 2, 4);

INSERT INTO Szallitas VALUES ("2021-01-10", 2, 5);

INSERT INTO Szallitas VALUES ("2020-12-21", 3, 6);

INSERT INTO Vevo VALUES (1,1);

INSERT INTO Vevo VALUES (2,2);

INSERT INTO Pontgyujto\_kartya VALUES (1,"Meszaros Sándor", "Tiszaluc", "Csokonai utca", "18", 3565, "meszaros@gmail.hu","+3630532187", 1);

INSERT INTO Pontgyujto\_kartya VALUES (2,"Kozma Sára", "Taktaharkany", "Jokai utca", "36", 3922, "kozma@gmail.hu","+3630857474", 2);

**Lekérdezések**

**Listázza ki azokat a termékeket amelyek drágábbak mint 350.**

SELECT Nev FROM Termek WHERE Ar > 350;

πnev σar > 350termek

**Listázza azon vásárlók nevét és telefonszámát, akik rendelkeznek pontgyűjtő kártyával.**

SELECT Nev, Telefonszam FROM Pontgyujto\_kartya ORDER BY Nev ASC;

τnev πnev, telefonszampontgyujto\_kartya

**Listázza azon vásárlók nevét, akik rendelkeznek pontgyűjtő kártyával és tiszalúciak.**

SELECT Nev FROM Pontgyujto\_kartya WHERE Varos = "Tiszaluc";

πnev σvaros = "Tiszaluc"pontgyujto\_kartya

**Listázza azt az üzletet amelyik minden nap 6-18 óráig van nyitva.**

SELECT \* FROM Uzlet WHERE Nyitvatartas = "6-18";

σnyitvatartas = "6-18"uzlet

**Listázza ki azokat a termékeket melyek tejtermékek.**

SELECT \* FROM Termek WHERE Nev LIKE '%tej%';

σnev LIKE "%tej%"termek

**Listázza ki azokat a termékeket melyek 2021-02-15 előtt érkeztek.**

SELECT \* FROM Termek LEFT JOIN Szallitas ON Szallitas.Termek\_ID = Termek.Termek\_ID WHERE Szallitas.Datum < "2021-02-15";

σ Szallitas.Datum < "2021-02-15"⨝ Szallitas.Termek\_ID = Termek.Termek\_ID Szallitas

**Listázza ki azon dolgozók összes adatát, akik a Báziscsemege üzletben dolgoznak.**

SELECT \* FROM Dolgozo LEFT JOIN Uzlet ON Dolgozo.Uzlet\_ID = Uzlet.Uzlet\_ID WHERE Uzlet.Nev = "Baziscsemege";

σ Uzlet.Nev = "Baziscsemege"⨝ Dolgozo.Uzlet\_ID = Uzlet.Uzlet\_ID Uzlet

**Listázza ki az átlagosnál drágább termékek összes adatát.**

SELECT \* FROM Termek WHERE Ar > (SELECT AVG(Ar) FROM Termek);

σar  > πAVG (ar) γAVG (ar)termek

**Listázza ki üzletenként a dolgozók számát csökkenő sorrendben.**

SELECT Uzlet.Nev, (SELECT COUNT(\*) FROM Dolgozo WHERE Dolgozo.Uzlet\_ID=Uzlet.Uzlet\_ID) as Dolgozok\_Szama FROM Uzlet ORDER BY Dolgozok\_Szama DESC;

τdolgozok\_szama ↓ (π COUNT (\*) γCOUNT (\*) σdolgozo . uzlet\_id = uzlet . uzlet\_iddolgozo)πuzlet . nev → dolgozok\_szamauzlet

**Listázza ki melyik beszállító melyik üzletbe szállított már termékeket.**

SELECT DISTINCT Uzlet.Nev, Nagyker.Nev FROM Uzlet JOIN Termek ON Termek.Uzlet\_ID = Uzlet.Uzlet\_ID JOIN Szallitas ON Szallitas.Termek\_ID = Termek.Termek\_ID JOIN Nagyker ON Nagyker.Nagyker\_ID = Szallitas.Nagyker\_ID;

δ πuzlet . nev, nagyker . nev(uzlet ⋈termek . uzlet\_id = uzlet . uzlet\_idtermek ⋈szallitas . termek\_id = termek . termek\_idszallitas ⋈nagyker . nagyker\_id = szallitas . nagyker\_idnagyker)