

JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves feladat

Futással foglalkozó sportbolt

Készítette: Karczub Roland

Neptunkód: KJSPMW

Dátum: 2022.11.20.

A FELADAT LEÍRÁSA:	3
1 A.) AZ ADATBÁZIS ER-MODELLJE:	5
1 B.) AZ ER MODELL KONVERTÁLÁSA RELÁCIÓS MODELLRE:	6
1 C.) AZ ADATBÁZIS RELÁCIÓS SÉMÁJA:	7
1 D.) TÁBLÁK LÉTREHOZÁSA:	8
1 E.) AZ ADATTÁBLÁK FELTÖLTÉSE:	11
1 F.) LEKÉRDEZÉSEK (SQL ÉS RELÁCIÓS ALGEBRA)	12

A feladat leírása:

A fél éves beadandó feladatomban egy futással foglalkozó sportbolt, ahol interneten keresztül lehet főképp futáshoz termékeket rendelni. Több raktár tárolja az adott termékeket, és több alkalmazott szerepel benne. Az említett adatokat ahhoz, hogy nyilvántartsam, öt egyedet hoztam létre, amik a következők:

- Vevő
- Rendelés
- Termék
- Raktár
- Alkalmazott

Elsősorban a **Vevő** egyedet szeretném bemutatni, ez tartalmazza a rendelő adatait, pontosítva egészen a címen belül az irányítószám, település, házsám, utcáig. Magába foglalja a telefonszámát, ami egy több értékű tulajdonságként lett létrehozva, nevét, email-címét és a személyi számát, ami ebben az egyedben megkapta az elsődleges kulcsot, mivel ez alapján van beazonosítva a vevő, ez az a tulajdonság, amiből minden vevőnek különböző van.

A második a sorban a **Rendelés** egyed, ami magába foglalja a rendeléseket, pontosabban egy Rendelés_ID-t, ami a konkrét rendelés száma, a nyomon követés céljából, ez maga az elsődleges kulcs. Tartalmazza még az egyed, a rendelés árát, a dátumot, amikor a rendelés leadásra került, illetve a rendelés típusát, hogy készpénzzel vagy bankkártyával fizetett a vevő. Egy vevő csak egy rendelést tud leadni.

A következő a **Termék** egyed, ami összeköttetésben áll a Rendelés egyeddel, és egy rendeléshez egyértelműen több termék tartozik. Tartalmazza a termék három féle típusát, ami a cipő, ruha, és kiegészítőket foglalja magába, illetve egy Termék_ID-t, ami az egyedben az elsődleges kulcs szerepét kapta meg, ez az ID a termékeket jelöli, mindegyik termék külön ID-val rendelkezik.

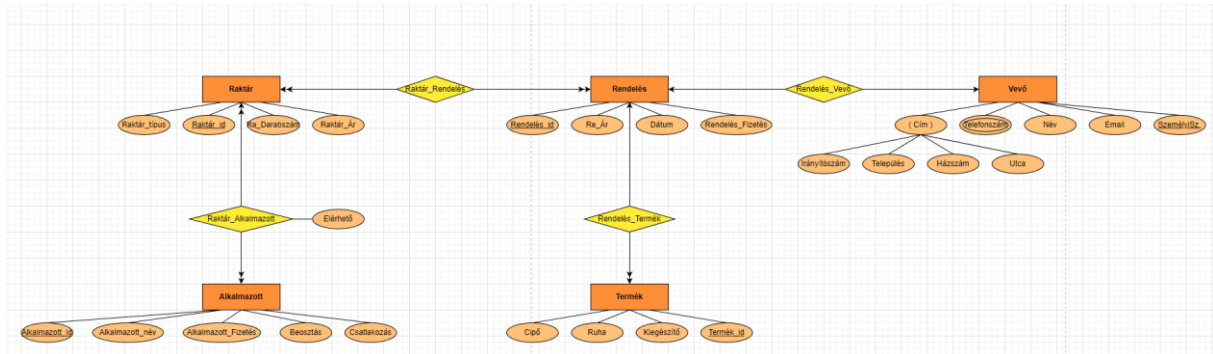
A beérkezett rendeléseket egy **Raktár** egyeddel kapcsolom össze, ahol több rendelés több raktárhoz tartozhat, tehát, hogyha az adott termék nincsen az egyik raktárban, viszont egy másikban készleten lehet. A Raktár tartalmazza a termék típusát, a Raktár_ID-t, ami az elsődleges kulcs, ez tartja nyilván külön a raktárakat. Szerepel még a darabszám, hogy a rendelt termékből hány darab érhető el, illetve a raktárban lévő termék árát birtokolja még.

Az végső a sorban, az **Alkalmazott** egyed, ami több-több kapcsolatban áll a Raktárral, tehát több raktárhoz, több alkalmazott tartozik. Az Alkalmazott egyedben szerepel az Alkalmazott_ID, ami elsődleges kulcs, és ez az alkalmazottak egyedi azonosítója. Tartalmazza még az alkalmazott nevét,

fizetését, beosztását, illetve a csatlakozás dátumát, ami egyfajta tapasztalatot is lefed, hogy ha az alkalmazott már régebb óta dolgozik ott, akkor nagy valószínűséggel nagyobb tapasztalattal is rendelkezik, mint az, aki nemrég kezdett.

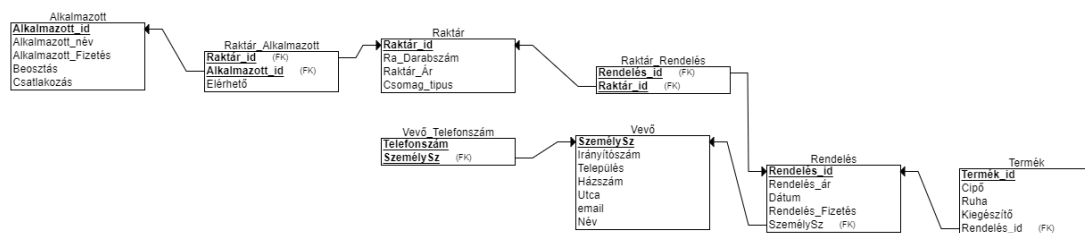
1 a.) Az adatbázis ER-modellje:

Egy öt egyedből álló kapcsolati rendszerrel, egy-egy, egy-több, és több-több kapcsolatokat tartalmaz. A raktár_alkalmazott egyedek között a raktár_alkalmazott kapcsolatnál létrehoztam egy elérhető tulajdonságot, ami azt tartja számon, hogy az alkalmazott elérhető-e. A telefonszámot neveztem meg többértékű tulajdonságnak, mivel egy vevő lehet több telefonja is, mindegyik egyedben szerepel egy elsődleges kulcs, a címet tettem összetett tulajdonságként, ami alatt további négy tulajdonság szerepel.



1 b.) Az ER modell konvertálása relációs modellre:

Próbáltam egy átlátható relációs modellt készíteni, figyeltem a sorrendre és az elrendezésre. Az Alkalmazott és a Raktár között létrehoztam egy Raktár_Alkalmazott táblát, ami tartalmaz két darab idegenkulcsot is. Szükségem volt a Raktár és Rendelés között is egy táblára, amik a Rendelés_ID-t, és a Raktár_ID-t foglalja magában. Illetve a harmadik tábla, amit még szükséges volt, az pedig a Vevő_Telefonszám volt, ami a többértékű tulajdonsággal bíró telefonszámot, és a SzemélySz egyedi kulcsot tartalmazza.



1 c.) Az adatbázis relációs sémája:

Termék [Termék_id, Cipő, Ruha, Kiegészítő, Rendelés_id]

Rendelés [Rendelés_id, Rendelés_ár, Dátum, Rendelés_Fizetés, SzemélySz]

Raktár_Rendelés [Rendelés_id, Raktár_id]

Raktár [Raktár_id, Ra_Darabszám, Raktár_Ár, Csomag_Típus]

Raktár_Alkalmazott [Raktár_id, Alkalmazott_id, Elérhető]

Vevő_telefonszám [Telefonszám, Személysz]

Vevő [SzemélySz, Név, Irányítószám, Település, Házsám, Utca, Email]

Alkalmazott [Alkalmazott_id, Alkalmazott_Név, Alkalmazott_Fizetes, Beosztás, Csatlakozás]

1 d.) Táblák létrehozása:

A create parancsot használtam az adattáblák elkészítéséhez. Fontos volt, hogy figyelnem kellett a helyes sorrendre, mikor létrehozom őket. Az SQL kódott XAMPP és MySQL segítségével hoztam létre, szerkesztettem.

```
CREATE TABLE Raktár
(
    Raktár_id INT NOT NULL,
    Ra_Darabszám INT NOT NULL,
    Raktár_Ár INT NOT NULL,
    Csomag_típus Varchar(20) NOT NULL,
    PRIMARY KEY (Raktár_id)
);
```

```
CREATE TABLE Vevő
(
    Irányítószám INT NOT NULL,
    Település Varchar(20) NOT NULL,
    Házsám INT NOT NULL,
    Utca Varchar(20) NOT NULL,
    email Varchar(60) NOT NULL,
    Név Varchar(20) NOT NULL,
    SzemélySz INT NOT NULL,
    PRIMARY KEY (SzemélySz)
);
```

```
CREATE TABLE Alkalmazott
(
    Alkalmazott_név Varchar(20) NOT NULL,
    Alkalmazott_Fizetés INT NOT NULL,
    Beosztás Varchar(20) NOT NULL,
    Csatlakozás Date NOT NULL,
    Alkalmazott_id INT NOT NULL,
    PRIMARY KEY (Alkalmazott_id)
);
```



```

CREATE TABLE Raktár_Alkalmazott
(
    Elérhető Boolean NOT NULL,
    Raktár_id INT NOT NULL,
    Alkalmazott_id INT NOT NULL,
    PRIMARY KEY (Raktár_id, Alkalmazott_id),
    FOREIGN KEY (Raktár_id) REFERENCES Raktár(Raktár_id),
    FOREIGN KEY (Alkalmazott_id) REFERENCES Alkalmazott(Alkalmazott_id)
);

CREATE TABLE Vevő_Telefonszám
(
    Telefonszám Varchar(20) NOT NULL,
    SzemélySz INT NOT NULL,
    PRIMARY KEY (Telefonszám, SzemélySz),
    FOREIGN KEY (SzemélySz) REFERENCES Vevő(SzemélySz)
);

CREATE TABLE Rendelés
(
    Rendelés_id INT NOT NULL,
    Rendelés_ár INT NOT NULL,
    Dátum Date NOT NULL,
    Rendelés_Fizetés Varchar(20) NOT NULL,
    SzemélySz INT NOT NULL,
    PRIMARY KEY (Rendelés_id),
    FOREIGN KEY (SzemélySz) REFERENCES Vevő(SzemélySz)
);

CREATE TABLE Termék
(
    Cipő Varchar(20) NOT NULL,
    Ruha Varchar(20) NOT NULL,
    Kiegészítő Varchar(20) NOT NULL,
    Termék_id INT NOT NULL,

```

```
Rendelés_id INT NOT NULL,  
PRIMARY KEY (Termék_id),  
FOREIGN KEY (Rendelés_id) REFERENCES Rendelés(Rendelés_id)  
);  
  
CREATE TABLE Raktár_Rendelés  
(  
    Rendelés_id INT NOT NULL,  
    Raktár_id INT NOT NULL,  
    PRIMARY KEY (Rendelés_id, Raktár_id),  
    FOREIGN KEY (Rendelés_id) REFERENCES Rendelés(Rendelés_id),  
    FOREIGN KEY (Raktár_id) REFERENCES Raktár(Raktár_id)  
);
```

1 e.) Az adattáblák feltöltése:

Szintén nagy szerepe volt annak, hogy milyen sorrendben töltöttem fel a táblákat, mert ha elvétem, akkor nem megfelelően hivatkoznak egymás tulajdonságaira. Ehhez a feltöltéshez az Insert to parancsot használtam.

```
KUSPMV_BESZTELAK_Sql
Limit to 1000 rows
1 • Insert into 'Vevő' ('SzemélySz', 'Irányítószám', 'Település', 'Házszám', 'Utc', 'Email', 'Név') values (23468233, 3900, 'Szerencs', 24, 'József Attila', 'kukoricai2@gmail.com', 'Kukor Ica');(67346823, 3
2 • Insert into 'Rendelés' ('Rendelés_id', 'Rendelés_ár', 'Dátum', 'Rendelés_Fizetés', 'SzemélySz') values (1, 22400, '2022-11-10', 'készpénz', 23468233);(2, 43128, '2022-11-03', 'kártya', 67346823);(3, 19846
3 • Insert into 'Termék' ('Termék_id', 'Cipő', 'Ruha', 'Kiegészítő', 'Rendelés_id') values (10, 'Hoka', 'Nike téli futónadrág', 'Petzl Fejlámpa', 1);(20, 'Saucony', 'Kalenji rövid futópóló', 'kompressziós szár',
4 • Insert into 'Raktár' ('Raktár_id', 'Ra_Darabszám', 'Raktár_ár', 'Csomag_típus') values (5, 97, 23454, 'Törékony'); (15, 98, 45565, 'Polyékony'); (25, 99, 12345, 'Törött');
5 • Insert into 'Alkalmazott' ('Alkalmazott_id', 'Csatlakozás', 'Alkalmazott_név', 'Alkalmazott_Fizetés', 'Beosztás') values (101, '2018-03-12', 'Ady Sándor', 413675, 'raktárvezető'); (102, '2008-07-10', 'Grizes
6
7 • Insert into 'Vevő_Telefonszám' ('Telefonszám', 'SzemélySz') values ('0630-456-7890', 23468233), ('0620-987-6543', 67346823), ('06-70-567-2345', 56259348);
8 • Insert into 'Raktár_Rendelés' ('Rendelés_id', 'Raktár_id') values (1, 5), (2, 15), (3, 25);
9 • Insert into 'Raktár_Alkalmazott' ('Alkalmazott_id', 'Raktár_id', 'Elérhető') values (101, 5, True), (102, 15, False), (103, 25, True);
```

1 f.) Lekérdezések (SQL és relációs algebra)

1.Kérjük le, annak az alkalmazottnak a nevét, aki a legtöbb fizetést kapta!

select Alkalmazott_név, max(Alkalmazott_fizetés) from alkalmazott;

π *alkalmazott_név, MAX (alkalmazott_fizetés)*
 γ *MAX (alkalmazott_fizetés) alkalmazott*

	Alkalmazott_név	max(Alkalmazott_Fizetés)
▶	Ady Sándor	413675

2.Kérdezze le azokat a vevőket, akik készpénzzel fizették a rendelésüket!

select név, rendelés.Rendelés_Fizetés from vevő

join rendelés on rendelés.SzemélySz = vevő.SzemélySz

where (rendelés.Rendelés_Fizetés = 'készpénz');

π *név, rendelés . rendelés_fizetés*
 σ *rendelés . rendelés_fizetés = "készpénz" (vevo ⋈ rendelés . személysz = vevő . személysz rendelés)*

	név	Rendelés_Fizetés
▶	Kukor Ica	készpénz
	Pesti Erzsébet	készpénz

3.Írja ki annak a vevőnek az adatait, akinek a rendelése a legolcsóbb volt!

select * from vevő

join rendelés on rendelés.SzemélySz = vevő.Személysz

where rendelés_ár = (select min(rendelés_ár) from rendelés);

π *vevo, MIN (rendelés_ár)*
 γ *MIN (rendelés_ár)*
 σ *rendelés_ár = MIN (rendelés_ár) (vevo × (rendelés ⋈ rendelés . személysz = vevp . személysz rendelés))*

	Irányítószám	Település	Házszám	Utca	email	Név	SzemélySz	Rendelés_id	Rendelés_ár	Dátum	Rendelés_Fizetés	SzemélySz
▶	3517	Miskolc	45	Duna	pestibözsi34@citromail.hu	Pesti Erzsébet	56259348	3	19846	2022-10-23	készpénz	56259348

4.Írja ki annak a vevőnek a nevét, és személyszámát, akinek az emailcíme freemail-t tartalmaz!

select név, SzemélySz from vevő

where email like '%freemail%';

π *név, személysz*
 σ *email LIKE "%freemail%" vevő*

	név	SzemélySz
▶	Kiss Pista	67346823

5. Írjuk ki azokat az elérhető alkalmazottakat, akiknek a fizetése nagyobb, mint 300 ezer!

```
select * from alkalmazott
```

```
join raktár_alkalmazott on raktár_alkalmazott.Alkalmazott_id = alkalmazott.Alkalmazott_id
```

```
where Elérhető = True and Alkalmazott_Fizetés > 300000;
```

π alkalmazott
 σ elérhető = true AND alkalmazott_fizetés > 300000 (alkalmazott \bowtie raktár_alkalmazott . alkalmazott_id = alkalmazott . alkalmazott_id raktár_alkalmazott)

	Alkalmazott_név	Alkalmazott_Fizetés	Beosztás	Csatlakozás	Alkalmazott_id	Elérhető	Raktár_id	Alkalmazott_id
▶	Ady Sándor	413675	raktárvezető	2010-03-12	101	1	5	101
	Gyári Zsolt	345678	Műszakvezető	1988-09-30	103	1	25	103

6. Kérdezze le azokat a ruhákat, amelyek futóruhák, és a rendelés ára kisebb, mint az átlag rendelésár!

```
select Ruha from termék
```

```
join rendelés on rendelés.Rendelés_id = termék.Rendelés_id
```

```
where Ruha like '%futó%' and rendelés.Rendelés_ár < (SELECT(avg(Rendelés_ár))from rendelés);
```

π ruha, AVG (rendelés_ár)
 γ AVG (rendelés_ár)
 σ ruha LIKE "%futó%" AND rendelés . rendelés_ár < AVG (rendelés_ár) (termék \times (rendelés \bowtie rendelés . rendelés_id = termék . rendelés_id rendelés))

	Ruha
▶	Nike téli futónadrág

7. Írd ki annak a vevőnek az adatait, aki pólót vett, és kártyával fizetett!

```
select * from vevő
```

```
join rendelés on rendelés.SzemélySz = vevő.Személysz
```

```
join termék on termék.Rendelés_id = rendelés.Rendelés_id
```

```
where termék.Ruha like '%futó%' and rendelés.Rendelés_Fizetés = 'kártya';
```

π vevo
 σ termék . ruha LIKE "%futó%" AND rendelés . rendelés_fizetés = "kártya" (vevo \bowtie rendelés . személysz = vevo . személysz rendelés \bowtie termék . rendelés_id = rendelés . rendelés_id termék)

	Irányítószám	Település	Házszám	Utca	email	Név	SzemélySz	Rendelés_id	Rendelés_ár	Dátum	Rendelés_Fizetés	SzemélySz	Cipő	Ruha	Kiegészítő	Termék_id	Rendelés_id
▶	3903	Bekecs	76	Virág	kisspistahun@freemail.hu	Kiss Pista	67346823	2	43128	2022-11-03	kártya	67346823	Saucony	Kalenji rövid futópóló	Kompressziós szár	20	2

8. Írja ki azt a cipőt, ahol a rendelés ár az átlag raktárár fölött van!

```
select Cipő from termék
```

```
join rendelés on rendelés.Rendelés_id = termék.Rendelés_id
```

```

join raktár_rendelés on raktár_rendelés.Rendelés_id = rendelés.Rendelés_id
join raktár on raktár.Raktár_id = raktár_rendelés.Raktár_id
where raktár.Raktár_Ár > (select(avg(rendelés.Rendelés_Ár))from rendelés);

```

```

π cipo, AVG (rendelés_ár)
Y AVG (rendelés_ár)
σ raktár . raktár_ár > AVG (rendelés_ár) (termék × (rendelés ⋈ rendelés . rendelés_id = termék . rendelés_id
rendelés ⋈ raktár_rendelés . rendelés_id = rendelés . rendelés_id raktár_rendelés ⋈ raktár . raktár_id =
raktár_rendelés . raktár_id raktár))

```

	Cipő
▶	Saucony

9.Írja ki azt az alkalmazott nevét, és beosztását, akinek a csatlakozás dátuma korábban van mint 2015!

```

select Alkalmazott_név, Beosztás from alkalmazott
where Csatlakozás < '2009-00-00';

```

```

π alkalmazott_név, beosztás
σ csatlakozás < "2009-00-00" alkalmazott

```

	Alkalmazott_név	Beosztás
▶	Grízes Kálmán	raktáros
	Gyári Zsolt	Műszakvezető

10.Írja ki azt a vevő nevet, és emailcímét, aki Szerencsen lakik és a telefonszáma 30-as!

```

select Név, email from vevő
join vevő_telefonszám on vevő_telefonszám.SzemélySz = vevő.SzemélySz
where vevő_telefonszám.Telefonszám like '%0630%' and vevő.Település = 'Szerencs';

```

```

π név, email
σ vevo_telefonszám . telefonszám LIKE "%0630%" AND vevo . település = "Szerencs" (vevo ⋈ vevo_telefonszám .
személysz = vevo . személysz vevo_telefonszám)

```

	Név	email
▶	Kukor Ica	kukorica12@gmail.com