

# 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:	12
1 F.) LEKÉRDEZÉSEK (SQL ÉS RELÁCIÓS ALGEBRA)	16

## 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.

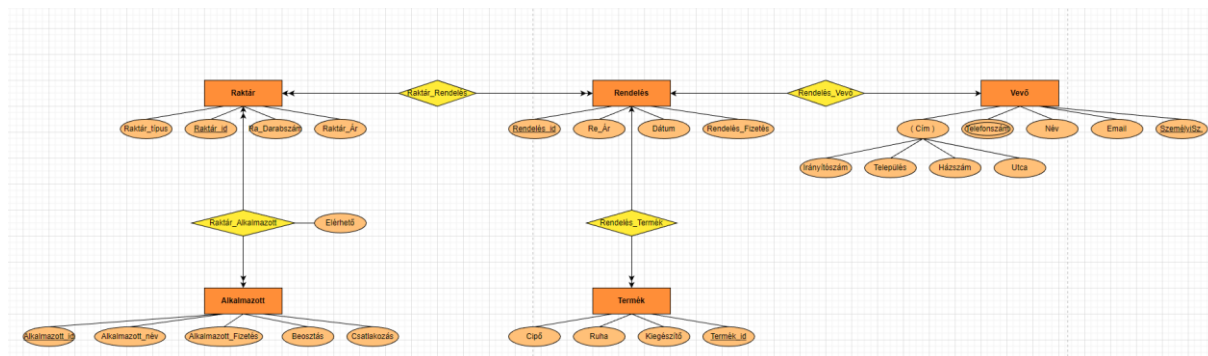
A beérkezett rendeléseket egy **Raktár** egyeddel kapcsolom össze, ahol több rendelés több raktárhoz tartozhat, tehát, hogyha az adott termék nincsen az egyik raktárban, viszont egy másikban készleten lehet. A Raktár tartalmazza a termék típusát, a Raktár\_ID-t, ami az elsődleges kulcs, ez tartja nyilván 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.

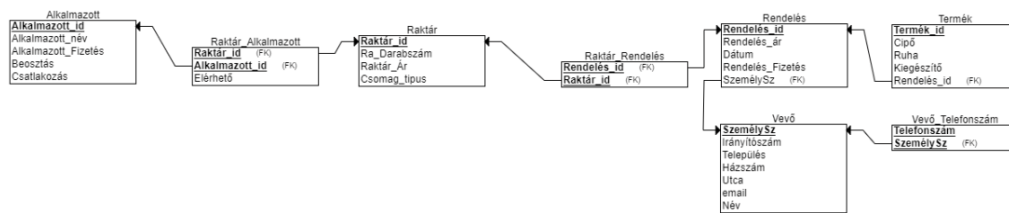
## 1 a.) Az adatbázis ER-modellje:

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(50) NOT NULL,  
    PRIMARY KEY (Raktár_id)  
);
```

	Field	Type	Null	Key	Default	Extra
►	Raktár_id	int(11)	NO	PRI	NULL	
	Ra_Darabszám	int(11)	NO		NULL	
	Raktár_Ár	int(11)	NO		NULL	
	Csomag_típus	varchar(50)	NO		NULL	

```
CREATE TABLE Vevő
```

```
(  
    Irányítószám INT NOT NULL,  
    Település Varchar(50) NOT NULL,  
    Házszám INT NOT NULL,  
    Utca Varchar(50) NOT NULL,  
    email Varchar(60) NOT NULL,  
    Név Varchar(50) NOT NULL,  
    SzemélySz INT NOT NULL,  
    PRIMARY KEY (SzemélySz)  
);
```

	Field	Type	Null	Key	Default	Extra
►	Irányítószám	int(11)	NO		NULL	
	Település	varchar(50)	NO		NULL	
	Házszám	int(11)	NO		NULL	
	Utca	varchar(50)	NO		NULL	
	email	varchar(60)	NO		NULL	
	Név	varchar(50)	NO		NULL	
	SzemélySz	int(11)	NO	PRI	NULL	



```
CREATE TABLE Alkalmazott
```

```
(  
    Alkalmazott_név Varchar(50) NOT NULL,  
    Alkalmazott_Fizetés INT NOT NULL,  
    Beosztás Varchar(50) NOT NULL,  
    Csatlakozás Date NOT NULL,  
    Alkalmazott_id INT NOT NULL,  
    PRIMARY KEY (Alkalmazott_id)  
);
```

	Field	Type	Null	Key	Default	Extra
►	Alkalmazott_név	varchar(50)	NO		NULL	
	Alkalmazott_Fizetés	int(11)	NO		NULL	
	Beosztás	varchar(50)	NO		NULL	
	Csatlakozás	date	NO		NULL	
	Alkalmazott_id	int(11)	NO	PRI	NULL	

```
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)  
);
```

	Field	Type	Null	Key	Default	Extra
►	Elérhető	tinyint(1)	NO		NULL	
	Raktár_id	int(11)	NO	PRI	NULL	
	Alkalmazott_id	int(11)	NO	PRI	NULL	

```
CREATE TABLE Vevő_Telefonszám
```

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

	Field	Type	Null	Key	Default	Extra
►	Telefonszám	varchar(50)	NO	PRI	NULL	
	SzemélySz	int(11)	NO	PRI	NULL	

CREATE TABLE Rendelés

(

Rendelés\_id INT NOT NULL,

Rendelés\_ár INT NOT NULL,

Dátum Date NOT NULL,

Rendelés\_Fizetés Varchar(50) NOT NULL,

SzemélySz INT NOT NULL,

PRIMARY KEY (Rendelés\_id),

FOREIGN KEY (SzemélySz) REFERENCES Vevő(SzemélySz)

);

	Field	Type	Null	Key	Default	Extra
►	Rendelés_id	int(11)	NO	PRI	NULL	
	Rendelés_ár	int(11)	NO		NULL	
	Dátum	date	NO		NULL	
	Rendelés_Fizetés	varchar(50)	NO		NULL	
	SzemélySz	int(11)	NO	MUL	NULL	

CREATE TABLE Termék

(

Cipő Varchar(50) NOT NULL,

Ruha Varchar(50) NOT NULL,

Kiegészítő Varchar(50) NOT NULL,

Termék\_id INT NOT NULL,

Rendelés\_id INT NOT NULL,

PRIMARY KEY (Termék\_id),

FOREIGN KEY (Rendelés\_id) REFERENCES Rendelés(Rendelés\_id)

);

	Field	Type	Null	Key	Default	Extra
►	Cipő	varchar(50)	NO		NULL	
	Ruha	varchar(50)	NO		NULL	
	Kiegészítő	varchar(50)	NO		NULL	
	Termék_id	int(11)	NO	PRI	NULL	
	Rendelés_id	int(11)	NO	MUL	NULL	

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)  
);
```

	Field	Type	Null	Key	Default	Extra
►	Rendelés_id	int(11)	NO	PRI	NULL	
	Raktár_id	int(11)	NO	PRI	NULL	

## 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_DSEINTASK_SQL
Limit to 1000 rows
1 • Insert into 'Vevő' ('SzemélySz', 'Irányítószám', 'Település', 'Házszám', 'Utca', 'Email', 'Név') values (23468233, 3900, 'Szerencs', 24, 'József Attila', 'kukorica12@gmail.com', 'Kukor Ica');
2 • Insert into 'Rendelés' ('Rendelés_id', 'Rendelés_ár', 'Dátum', 'Rendelés_Fizetés', 'SzemélySz') values (1, 22400, '2022-11-10', 'készpénz', 23468233);
3 • Insert into 'Termék' ('Termék_id', 'Cím', 'Ruhák', 'Kiegészítő', 'Rendelés_id') values (10, 'Huka', 'Nike téli futónadrág', 'Petzl Fejlámpa', 1);
4 • Insert into 'Raktár' ('Raktár_id', 'Ra_Darabszám', 'Raktár_ár', 'Csomag_típus') values (5, 97, 23454, 'Törékény');
5 • Insert into 'Alkalmazott' ('Alkalmazott_id', 'Csatlakozás', 'Alkalmazott_név', 'Alkalmazott_Fizetés', 'Beosztás') values (101, '2018-03-12', 'Ady Sándor', 413675, 'raktárvezető');
6
7 • Insert into 'Vevő_Telefonszám' ('Telefonszám', 'SzemélySz') values ('0630-456-7890', 23468233);
8 • Insert into 'Raktár_Rendelés' ('Rendelés_id', 'Raktár_id') values (1, 5);
9 • Insert into 'Raktár_Alkalmazott' ('Alkalmazott_id', 'Raktár_id', 'Elérhető') values (101, 5, True);
```

1 • `select * from Vevő;`

	Irányítószám	Település	Házszám	Utca	email	Név	SzemélySz
▶	3900	Szerencs	24	József Attila	kukorica12@gmail.com	Kukor Ica	23468233
	3517	Miskolc	45	Duna	pestibözsi34@citromail.hu	Pesti Erzsébet	56259348
	3903	Bekecs	76	Virág	kisspistaHUN@freemail.hu	Kiss Pista	67346823
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1 • `select * from Rendelés;`

	Rendelés_id	Rendelés_ár	Dátum	Rendelés_Fizetés	SzemélySz
▶	1	22400	2022-11-10	készpénz	23468233
	2	43128	2022-11-03	kártya	67346823
	3	19846	2022-10-23	készpénz	56259348
*	NULL	NULL	NULL	NULL	NULL

```
1 • select * from Termék;
```

Result Grid					
		Filter Rows:			
		Edit:			
	Cipő	Ruha	Kiegészítő	Termék_id	Rendelés_id
▶	Hoka	Nike téli futónadrág	Petzl Fejlámpa	10	1
	Saucony	Kalenji rövid futópó	Kompressziós szár	20	2
	Salomon	Felső alöltözet	Salomon mellény	30	3
*	NULL	NULL	NULL	NULL	NULL

```
1 select * from Raktár;
```

Result Grid				
		Filter Rows:		
		Edit:		
	Raktár_id	Ra_Darabszám	Raktár_Ár	Csomag_típus
▶	5	97	23454	Törékeny
	15	98	45565	Folyékony
	25	99	12345	Törött
*	NULL	NULL	NULL	NULL

```
1 • select * from Alkalmazott;
```

Result Grid					
		Filter Rows:			
		Edit:			
	Alkalmazott_név	Alkalmazott_Fizetés	Beosztás	Csatlakozás	Alkalmazott_id
▶	Ady Sándor	413675	raktárvezető	2010-03-12	101
	Grízes Kálmán	206342	raktáros	2008-07-10	102
	Gyári Zsolt	345678	Műszakvezető	1988-09-30	103
*	NULL	NULL	NULL	NULL	NULL

```
1 • select * from Vevő_Telefonszám;
```

Result Grid			Filter Rows:
	Telefonszám	SzemélySz	
▶	06-70-567-2345	56259348	
	0620-987-6543	67346823	
	0630-456-7890	23468233	
✱	NULL	NULL	

```
1 select * from Raktár_Rendelés;
```

```
2
```

Result Grid			Filter Rows:
	Rendelés_id	Raktár_id	
▶	1	5	
	2	15	
	3	25	
✱	NULL	NULL	

```
1 select * from Raktár_Alkalmazott;  
2
```

Result Grid			
Filter Rows:			
	Elérhető	Raktár_id	Alkalmazott_id
▶	1	5	101
	0	15	102
	1	25	103

## 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;

$\pi$  *alkalmazott\_név, MAX (alkalmazott\_fizetés)*  
 $\gamma$  *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');

$\pi$  *név, rendelés . rendelés\_fizetés*  
 $\sigma$  *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);

$\pi$  *vevo, MIN (rendelés\_ár)*  
 $\gamma$  *MIN (rendelés\_ár)*  
 $\sigma$  *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	pestibözi34@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%';

$\pi$  *név, személysz*  
 $\sigma$  *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;

$\pi$  alkalmazott  
 $\sigma$  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);

$\pi$  ruha, AVG (rendelés\_ár)  
 $\gamma$  AVG (rendelés\_ár)  
 $\sigma$  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';

$\pi$  vevo  
 $\sigma$  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	Házzszám	Utca	email	Név	SzemélySz	Rendelés_id	Rendelés_ár	Dátum	Rendelés_Fizetés	SzemélySz	Cipő	Ruha	Kiegészítő	Termék_id	Rendelés_id
▶	3903	Bekecs	76	Virág	kisspista@freemail.hu	Kiss Pista	67346823	2	43128	2022-11-03	kártya	67346823	Saucony	Kalenji rövid futópóló	Kompressziós szőr	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)
γ 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