

Jegyzőkönyv

Adatbázisrendszerek I.

Féléves feladat

Összeszerelő üzem

Készítette: Hegedüs Gábor

Neptunkód: DWCPH5

Dátum: 2021. 11. 30.

A feladat leírása:

A feladatban egy összeszerelő üzem adatbázisát hozom létre, az alábbi jellemzőkkel:

Company egyed maga a cég, tulajdonságai: Name cég neve, TIN cég adószáma, Industry a cég iparága és az address összetett tulajdonság, ami a címet adja meg négy részből, ZIP, city, street és door épül fel.

Employee egyed az alkalmazott, tulajdonságai: Eid egyedi azonosító, Name név, DOB születési dátum, Job munkakör, Sex nem. Job többértékű tulajdonság.

Machines egyed a cég tulajdonában lévő gépek, tulajdonságai: SN sorozatszám, Next_maint következő karbantartás dátuma, Last_maint legutolsó karbantartás dátuma, CP jelenleg gyártott termék, Name név.

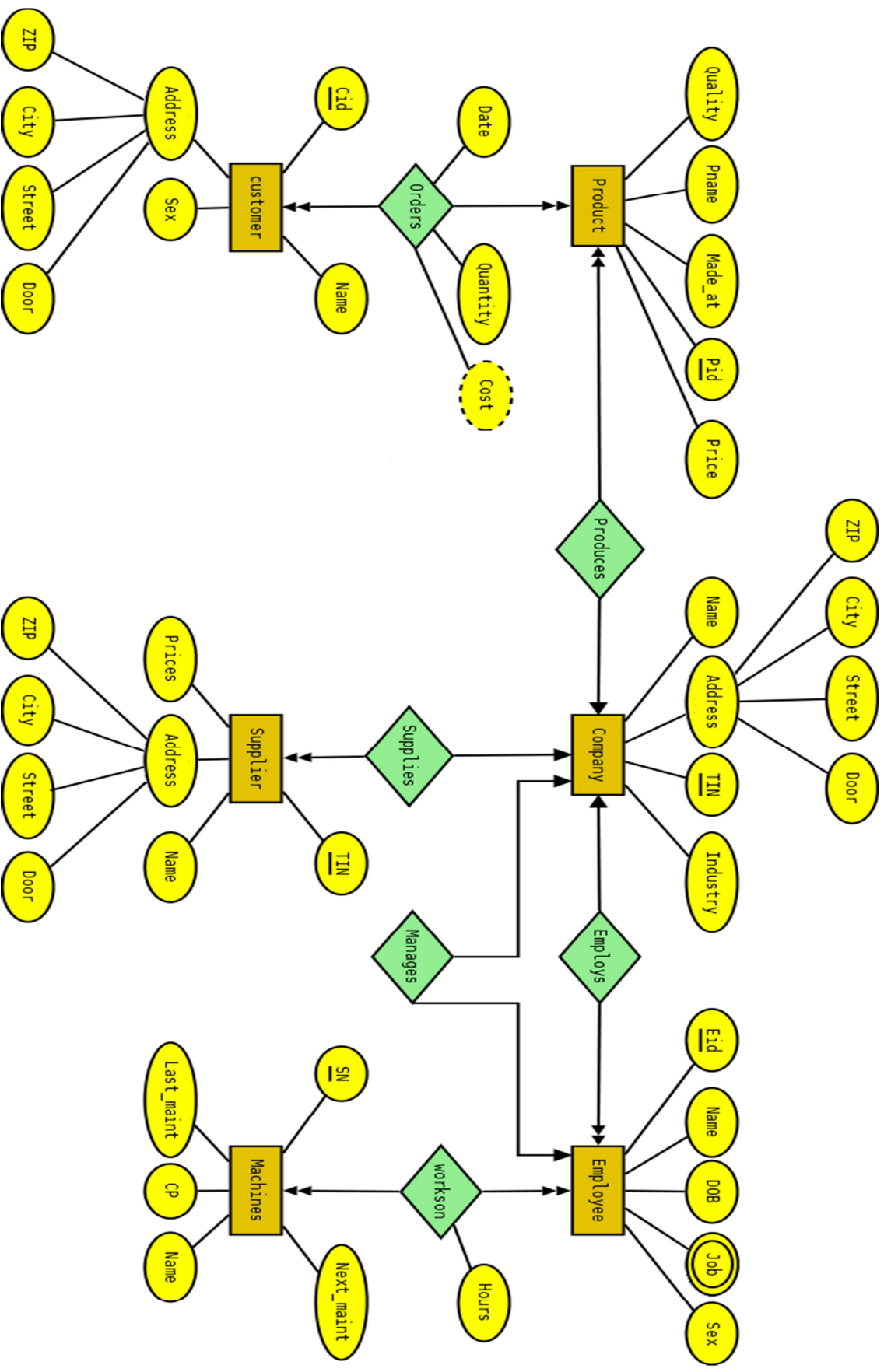
Product egyed a gyártott termék, tulajdonságai: Order összetett tulajdonság, ami a rendelést adja meg két részből, Date rendelés dátuma és Quantity mennyiség épül fel, Pname termék neve, Made_at elkészülés ideje, Pid egyedi azonosító, Quality minőség, Price a termék ára.

Supplier egyed a beszállító, tulajdonságai: TIN egyedi azonosító, Name beszállító neve, Prices az árak amiért szállít és az address összetett tulajdonság, ami a címet adja meg négy részből, ZIP, city, street és door épül fel.

Egy cégnek több alkalmazottja is lehet ezért Employs 1:N kapcsolatú viszont egy cégnek csak egy alkalmazottja manager ezért a Manages 1:1 kapcsolatú. Mivel több alkalmazott van minden géphez többen is tartozhatnak ezért a workson M:N típusú kapcsolat. Supplies 1:N típusú kapcsolat mivel a cégnek lehet több beszállítója is, ez a Company-t és a Supplier-t köti össze. Produces kapcsolat összeköti a Company-t és a Product-ot és mivel egy cég előállíthat több terméket is ezért ez egy 1:N kapcsolat ahol Company 1, Product pedig N. Orders kapcsolat összeköti a Product-ot a megrendelővel, Customer. Ez a kapcsolat M:N típusú mivel egy terméket többen is rendelhetnek viszont egy megrendelő kérhet több terméket is.

A cost tulajdonság a rendelés mennyiségéből és a beszállító áraiból számítható így az leszarmaztatott.

Az adatbázis ER modellje:



Az adatbázis konvertálása relációs modellre:

Company egyedből egy tábla lesz, az Address mező kimarad, csak a ZIP, City, Street és a Door mezők szerepelnek majd a táblában. Name varchar típusú legfeljebb ötven méretű. ZIP szám négy számjegy, City és Street char típusú legfeljebb negyven karakterből állhat, Door szám négy számjegy maximum, TIN varchar maximum tíz karakter, Industry varchar maximum tíz karakter.

Employee egyedből tábla lesz, Eid int maximum tíz jegyű, Name varchar típusú legfeljebb ötven méretű, DOB dátum típusú, Sex varchar maximum tíz karakter. Job többértékű tulajdonság így külön táblába kerül egy idegen kulcs Eid és egy Jdesc nevű varchar(50) mezőkkel.

Product egyedből tábla lesz, Quality varchar maximum tíz karakter, Pname varchar maximum 50 karakter, Made_at dátum, Pid int maximum 10 jegyű, Price int nem lehet nulla.

Supplier egyedből tábla lesz, az Address mező kimarad, csak a ZIP, City, Street és a Door mezők szerepelnek majd a táblában. ZIP szám négy számjegy, City és Street char típusú legfeljebb negyven karakterből állhat, Door szám négy számjegy maximum, TIN varchar maximum tíz karakter, Prices int nem lehet nulla, Name varchar maximum 50 karakter.

Customer egyedből tábla lesz, az Address mező kimarad, csak a ZIP, City, Street és a Door mezők szerepelnek majd a táblában. ZIP szám négy számjegy, City és Street char típusú legfeljebb negyven karakterből állhat, Door szám négy számjegy maximum, Cid varchar maximum tíz karakter, Sex varchar maximum tíz karakter, Name varchar maximum ötven karakter.

Machines egyedből tábla lesz, SN szám maximum tizenöt jegyű, Next_maint és Last_maint dátum, Name és CP char típusú legfeljebb húsz karakter.

Workson N:M kapcsolat, ezért ebből tábla lesz, amelyben két idegen kulcs tartja majd a kapcsolatot az Employee és Machines táblákkal. A táblába bekerül a Hours mező.

Orders N:M kapcsolat, ezért ebből tábla lesz, amelyben két idegen kulcs tartja majd a kapcsolatot a Product és a Customer táblákkal. A táblába bekerülnek a Date és Quantity mezők.

Az adatbázis relációs modellje:

Product					
Quality	Pname	Made_at	PId	Price	Coid
C10	C50	D	I10	I	C10

Company					
ZIP	City	Street	Door	TIN	Name
I4	C40	C40	I4	C10	C50
					Industry
					C10

Employee				
Coid	Name	DOB	Sex	Eid
C10	C50	D	C10	I10

Job	
Eid	JDesc
I10	C50

Workson		
Eid	MId	Hours
I15	I10	I2

Orders			
CId	PId	Date	Quantity
I10	I10	D	I3

Customer					
Sex	ZIP	City	Street	Door	Name
C10	I4	C40	C40	I4	C50
					CId
					C10

Supplier					
Prices	ZIP	City	Street	Door	Name
I	I4	C40	C40	I4	C10
					TIN
					C10
					Coid
					C10

Machines				
SN	Next_maint	Last_maint	Name	CP
I15	D	D	C20	C20

Az adatbázis relációs sémái:

Company [ZIP, City, Street, Door, TIN, Name, Industry]

Product [Quality, Name, Made_at, Pid, Price, Coid]

Employee [Coid, Name, DOB, Sex, Eid]

Job [Eid, description]

Orders [Cid, Pid, Date, Quality]

Workson [Eid, Mid, Hours]

Customer [Sex, ZIP, City, Street, Door, Name, Cid]

Supplier [Prices, ZIP, City, Street, Door, Name, TIN, Coid]

Machines [SN, Next_maint, Last_maint, Name, CP]

A táblák létrehozása:

Create table Company (

ZIP int(4),

City varchar(40),

Street varchar(40),

Door int(4),

TIN varchar(10),

Name varchar(50),

Industry varchar(10),

PRIMARY KEY (TIN));

Create table Employee (

Coid varchar(10),

Name varchar(50),

DOB date,

Sex varchar(10),

Eid int(10),

PRIMARY KEY (Eid),

FOREIGN KEY(Coid) REFERENCES Company(TIN));

Create table Supplier (

Prices int NOT NULL,

ZIP int(4),

City varchar(40),

Steet varchar(40),

Door int(4),

Name varchar(50),

TIN varchar(10),

Coid varchar(10),

PRIMARY KEY(TIN),

FOREIGN KEY(Coid) REFERENCES Company(TIN));

Create table Product (

Quality varchar(10),

Pname varchar(50),

Made_at date,

Pid int(10),

Price int NOT NULL,

Coid varchar(10),

PRIMARY KEY(Pid),

FOREIGN KEY(Coid) REFERENCES Company(TIN));

Create table Machines (

SN int(15),

Next_maint date,

Last_maint date,

Name varchar(20),

CP varchar(20),

PRIMARY KEY(SN));

Create table Customer (

Sex varchar(10),

```
ZIP int(4),  
City varchar(40),  
Steet varchar(40),  
Door int(4),  
Name varchar(50),  
Cid int(5),  
PRIMARY KEY(Cid));
```

Create table Workson (

```
Eid int(15),  
Mid int(10),  
Hours int(2),  
FOREIGN KEY(Eid) REFERENCES Employee(Eid),  
FOREIGN KEY(MID) REFERENCES Machines(SN));
```

Create table Orders (

```
Cid int(10),  
Pid int(10),  
Date date,  
Quantity int(3));
```

Create table Job (

```
Eid int(10),  
Jdesc varchar(50),  
FOREIGN KEY(Eid) REFERENCES Employee(Eid);
```

)

A táblák feltöltése:

Cég:

Insert into Company values(3600, 'Ózd', 'Nemzetor_ut', 1, 'AE1234567', 'Labda and Co', 'Sportszer');

Alkalmazott:

Insert into Employee values ('AE1234567', 'Mata Anna', '1987-09-22', 'Female', 1);

Insert into Employee values ('AE1234567', 'Albert Zsolt', '1970-01-15', 'Male', 2);

Insert into Employee values ('AE1234567', 'Kalmar Bela', '1965-02-02', 'Male', 3);

Insert into Employee values ('AE1234567', 'Anna Maria', '1989-03-14', 'Female', 4);

Insert into Employee values ('AE1234567', 'Ella Fitzgerald', '1985-05-10', 'Female', 5);

Insert into Employee values ('AE1234567', 'Kiss Gyula', '1992-01-01', 'Male', 6);

Job:

insert into Job values (1, 'Mernok');

insert into Job values (2, 'Sor');

insert into Job values (3, 'Darus');

insert into Job values (4, 'HR');

insert into Job values (5, 'Fordító');

insert into Job values (6, 'CEO');

Gépek:

Insert into Machines values ('123456789', '2021-12-30', '2020-12-30', 'Froccsonto', 'muanyag szelep');

Insert into Machines values ('143456789', '2022-02-30', '2020-11-30', 'Daru', 'raklap');

Insert into Machines values ('125456789', '2022-05-30', '2020-10-30', 'Eszterga', 'Bowling Pin');

Insert into Machines values ('123466789', '2022-03-30', '2020-09-30', 'Pres', 'Ping pong uto');

Insert into Machines values ('123456769', '2022-04-30', '2020-08-30', 'Festo gep', 'Labda boritas');

Workson kapcsolat táblája:

Insert into Workson values (1, 123456789, 6);

Insert into Workson values (1, 125456789, 6);

Insert into Workson values (3, 143456789, 12);

Insert into Workson values (2, 123466789, 6);

Insert into Workson values (2, 123456769, 6);

Termékek:

Insert into Product values ('normal', 'Ping pong uto', '2021-11-30', 1, 1500, 'AE1234567');

Insert into Product values ('high', 'Labda', '2020-10-20', 2, 2000, 'AE1234567');

Insert into Product values ('normal', 'Bowling Pin', '2020-11-25', 3, 2500, 'AE1234567');

Insert into Product values ('high', 'teniszuto', '2021-10-12', 4, 5000, 'AE1234567');

Insert into Product values ('normal', 'teniszlabda', '2021-09-10', 5, 100, 'AE1234567');

Vásárló:

Insert into Customer values ('Male', 3600, 'Ózd', 'Marcius 15. utca.', 1, 'AE1234567', 'Ala Bard', '001');

Insert into Customer values ('Male', 3600, 'Ózd', 'Petofi ter', 1, 'AE1234567', 'Falo Balazs', '002');

Insert into Customer values ('Male', 3600, 'Ózd', 'Nemzetor ut', 60, 'AE1234567', 'Karo Antal', '003');

Insert into Customer values ('Male', 3600, 'Ózd', 'Marcius 15. utca', 15, 'AE1234567', 'Horvath Zsolt', 'C004');

Insert into Customer values ('Male', 3600, 'Ózd', 'Gyar utca', 12, 'AE1234567', 'Kala Pal', '005');

Orders kapcsolat táblája:

Insert into Orders values (001, 1, '2021-11-30', 2);

Insert into Orders values (002, 2, '2021-11-30', 50);

Insert into Orders values (003, 4, '2021-10-30', 2);

Insert into Orders values (003, 5, '2021-10-30', 50);

Insert into Orders values (004, 3, '2021-11-25', 100);

Beszállító:

Insert into Supplier values (1000, 3527, 'Miskolc', 'Sajo ut', 1, 'En es En szállit', 'BE1234567', 'AE1234567');

Insert into Supplier values (1500, 3527, 'Miskolc', 'Sajo ut', 4, 'A es B is szállit', 'BE2234567', 'AE1234567');

Insert into Supplier values (1700, 3527, 'Miskolc', 'Sajo ut', 7, 'Botos es tarsa', 'BE2334567', 'AE1234567');

Insert into Supplier values (20000, 3527, 'Miskolc', 'Sajo ut', 9, 'Karcsi bt', 'BE4234567', 'AE1234567');

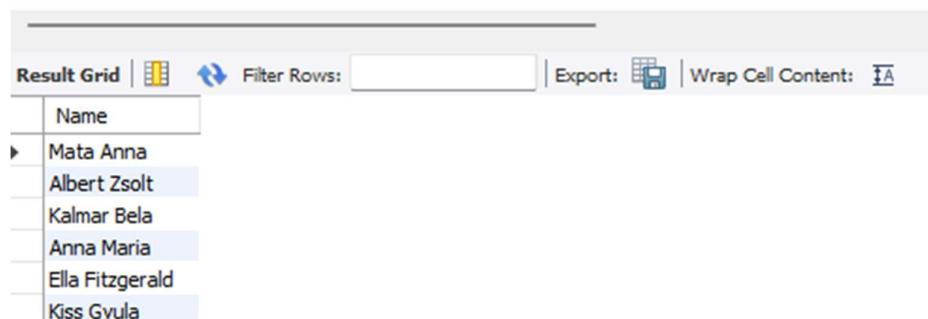
Lekérdezések:

1. Dolgozók neve

Select Name from Employee;

Π Name (Employee)

```
1 • Select Name from Employee;A
```



The screenshot shows a database interface with a 'Result Grid' tab. Below the tab, there is a toolbar with 'Filter Rows:', 'Export:', and 'Wrap Cell Content:' options. The main area displays a table with one column, 'Name', containing the following rows: 'Mata Anna', 'Albert Zsolt', 'Kalmar Bela', 'Anna Maria', 'Ella Fitzgerald', and 'Kiss Gyula'.

Name
Mata Anna
Albert Zsolt
Kalmar Bela
Anna Maria
Ella Fitzgerald
Kiss Gyula

2. Kiválasztja a férfi dolgozók nevét és születési dátumát

Select Name,DOB From Employee where Sex like 'Male';

Π Name, DOB (σ Sex like Male (Employee))



```
1 Select Name, DOB From Employee where Sex like 'Male';
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Name	DOB			
▶	Albert Zsolt	1970-01-15			
	Kalmar Bela	1965-02-02			
	Kiss Gyula	1992-01-01			

3. Kiválasztja a Gépek táblából a Név oszlopot amit gépek neve néven ír ki, minden értékből csak egy darabot.

Select distinct Name as Gepek_neve from Machines;

Π Name(Machines);

 Limit to 1000 rows 

```
1 Select distinct Name as Gepek_neve from Machines;
```

Open a script file in this editor

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Gepek_neve			
▶	Festo gep			
	Froccsonto			
	Pres			
	Eszterga			
	Daru			

4. Kiválasztja a 2500-nál olcsóbb termékek nevét és árát

Select Pname, Price from Product where Price<2500;

Π Pname, Price (σ Price<2500 (Products))

1 `Select Pname, Price from Product where Price<2500;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Pname	Price
▶	Pinguto	1500
	Labda	2000
	labda2	100

5. Kiválasztja a Beszállító tábla egész tartalmát:

`select * from Supplier;`

Π (Supplier)

1 • `select * from Supplier;`

Result Grid | Filter Rows: | Edit: | Export/Impo

	Prices	ZIP	City	Steet	Door	Name	TIN	Coid
▶	1000	3527	Miskolc	Sajo ut	1	En es En	BE2234567	AE1234567
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Kiválasztja nevét és mennyit rendelt az Orders és a Customer táblákból

`select Customer.Name, Orders.Quantity`

`from Customer`

`inner join Orders on Customer.Cid = Orders.Cid;`

$\Pi_{\text{customer . name, orders . quantity}} (\text{customer} \bowtie_{\text{customer . cid = orders . cid}} \text{orders})$

```

1  select Customer.Name, Orders.Quantity
2  from Customer
3  inner join Orders on Customer.Cid = Orders.Cid;
4

```

Result Grid			Filter Rows:	Export:	Wrap Cell
	Name	Quantity			
▶	Ala Bard	2			
	Kata Pult	50			
	Karo Antal	2			
	Karo Antal	50			

7. Kiválasztja nevét és mennyit rendelt az Orders és a Customer táblákból és csökkenő sorrendben kiírja név szerint

```

select Customer.Name, Orders.Quantity
from Customer
inner join Orders on Customer.Cid = Orders.Cid
group by Name
order by Name desc;

```

$\tau_{name} \downarrow$

$\pi_{customer.name, orders.quantity}$

$\gamma_{name, (customer \bowtie_{customer.cid = orders.cid})}$

```

2  from Customer
3  inner join Orders on Customer.Cid = Orders.Cid
4  group by Name
5  order by Name desc;
6

```

Result Grid			Filter Rows:	Export:	Wrap C
	Name	Quantity			
▶	Kata Pult	50			
	Karo Antal	2			
	Ala Bard	2			

8. Kiválasztja a dolgozókat, a gépeiket és kiírja mennyit dolgoztak
- ```

SELECT Employee.Name, Machines.Name, Workson.Hours
From ((Workson
INNER JOIN Machines ON Workson.Mid = Machines.SN)
INNER JOIN Employee ON Workson.Eid = Employee.Eid);

```

$\pi_{employee.name, machines.name, workson.hours} (workson \bowtie_{workson.mid = machines.sn} machines \bowtie_{workson.eid = employee.eid} employee)$

```

1 • SELECT Employee.Name, Machines.Name, Workson.Hours
2 From ((Workson
3 INNER JOIN Machines ON Workson.Mid = Machines.SN)
4 INNER JOIN Employee ON Workson.Eid = Employee.Eid);
5

```

| Name         | Name       | Hours |
|--------------|------------|-------|
| Albert Zsolt | Festo gep  | 6     |
| Mata Anna    | Froccsonto | 6     |
| Mata Anna    | Froccsonto | 6     |
| Albert Zsolt | Pres       | 6     |
| Albert Zsolt | Pres       | 6     |
| Mata Anna    | Eszterga   | 6     |
| Mata Anna    | Eszterga   | 6     |
| Kalmar Bela  | Daru       | 12    |

9. Azon termékek neve amik ára meghaladja az 5000 értéket.

```

SELECT Product.Pname, sum(price) FROM Orders
JOIN Product ON Orders.Pid = Product.pid
GROUP BY Pname
HAVING sum(price) >= 5000;

```

$\pi_{product.pname, SUM(price)}$

$\sigma_{SUM(price) \geq 5000}$

$\gamma_{pname, SUM(price)} (orders \bowtie_{orders.pid = product.pid} product)$

```

1 • SELECT Product.Pname, sum(price) FROM Orders
2 JOIN Product ON Orders.Pid = Product.pid
3 GROUP BY Pname
4 HAVING sum(price) >= 5000;
5

```

| Pname     | sum(price) |
|-----------|------------|
| teniszuto | 5000       |

10. Irányítószáma és Városa annak a vásárlónak aki a legkevesebbet rendelte és férfi

```

SELECT ZIP, City FROM Customer
JOIN Orders ON Orders.Cid = Customer.Cid
Join Product On Orders.Pid = Product.Pid
WHERE Customer.Sex = 'Male' and Product.Quality= 'Normal'
AND Orders.quantity = (SELECT MIN(quantity) FROM Orders);

```

$\pi_{zip, city}$ 
$$\sigma_{customer.sex = "Male" \text{ AND } product.quality = "Normal" \text{ AND } orders.quantity = (\pi_{MIN}(quantity))}$$
$$Y_{MIN(quantity)} orders) (customer$$

$\bowtie_{orders.cid = customer.cid}$   $orders \bowtie_{orders.pid = product.pid}$   $product$ )

