

Adatbáziskezelés
INSA032LMSB_Gyakorlat:G01
2023/2024 ŐSZ

Házi dolgozat

Háziorvosi rendelő működési adatait
tartalmazó adatbázis elkészítése

Készítette:
Simon Járóka Heléna Lilla
XY társ

Tartalom

I. Adatbázis bemutatása	2
II. Adatbázis tervezése	2
III. Az adatbázis felépítése	6
IV. Lekérdezések.....	7
V. Kimutatások-Riportok.....	14
VI. Mellékletek	16

I. Adatbázis bemutatása

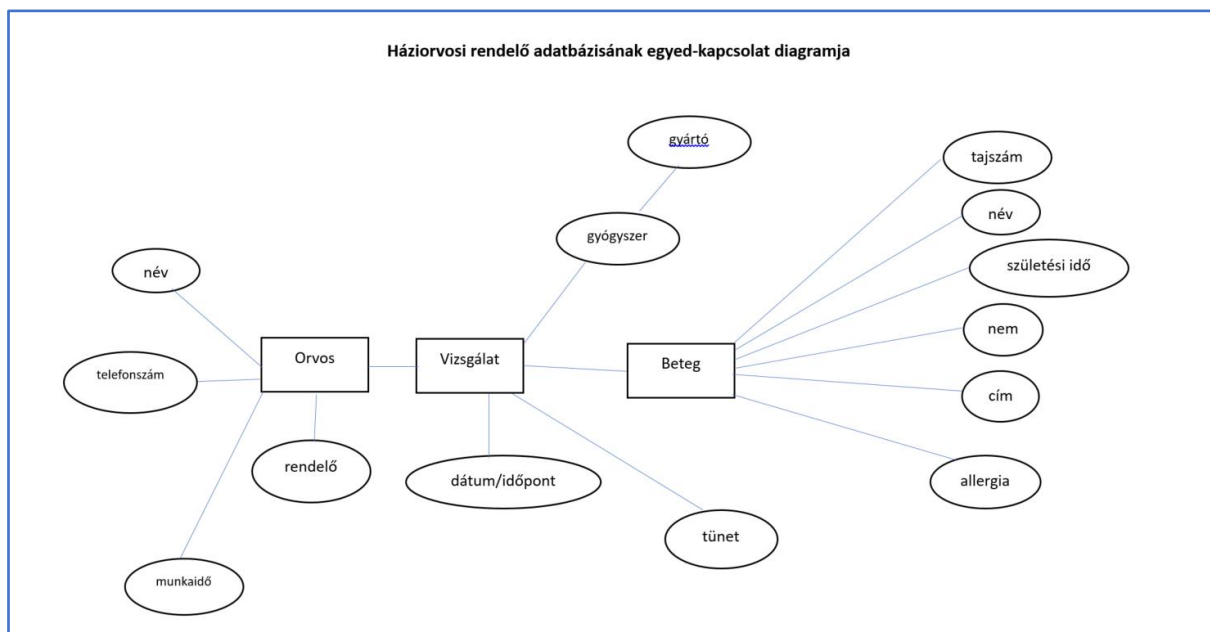
A házidolgozat elkészítése során kialakított adatbázis egy háziorvosi rendelő működése kapcsán keletkező adatokat tartalmazza egy hónap intervallumban. Az adatbázis célja, hogy rögzítse a háziorvosi rendelőben megforduló páciensek alapvető adatait, a rendelőbe történő megjelenésük célját, az esetleges tüneteiket, a részükre felírt gyógyszereket, az őket kezelő orvos adatait és lehetővé tegye a betegek, betegségekre, orvosokra és gyógyszerekre vonatkozó adatok különböző szempontok alapján történő értékelését, elemzését.

Az adatbázis elkészítése során SQLite adatbáziskezelő szoftver 3.12.2 verzióját használtuk, az adatok elemzését POWER BI elemző eszközzel végeztük. Ez utóbbi használatához szükség volt egy SQLite 64 bit ODBC program telepítésére ([sqliteodbc_w64.exe](#)) az adatmodel POWER BI-ba történő beolvasása érdekében.

Az adatbázis elkészítése során egy relációs adatbázis modell került kialakításra, amely összesen 10 sémában tartalmazza az adatokat.

II. Adatbázis tervezése

Az adatbázis egy háziorvosi rendelőben dolgozó orvosok, az általuk végzett vizsgálatok és a vizsgált betegek adatait tartalmazza. Az adatbázisban tárolandó adatok körét és viszonyait az alábbi egyed-kapcsolat diagram ábrázolja.



A fenti diagramból kiindulva az adatok tárolásának három entitás (Orvos, Vizsgálat, Beteg) áll a középpontjában, amelyek kapcsolatban állnak egymással. Az adatbázis adatszerkezetének megtervezése során ezért a három entitás és attribútumaik tárolását biztosító táblákból indultunk ki, amelyek között a kapcsolatot idegen kulcsok alkalmazásával alakítottuk ki.

Beteg					
Beteg taj szám	Beteg név	Beteg születési idő	Beteg neme	Beteg címe	Beteg allergiája

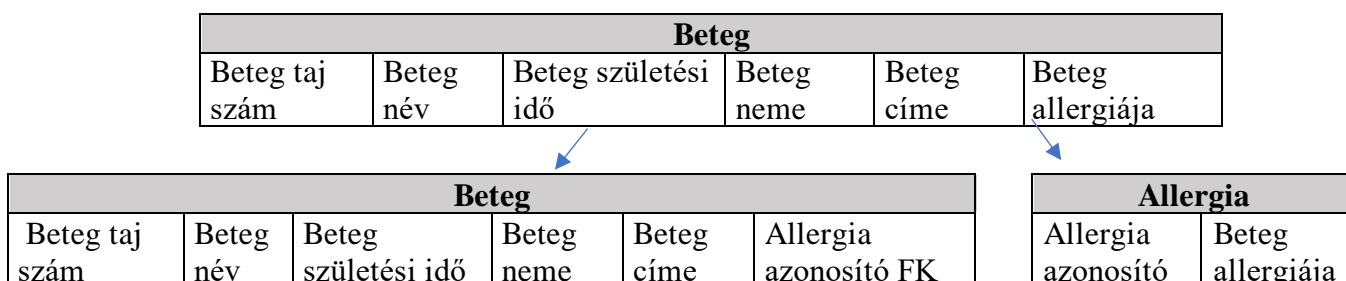
Orvos				
Orvos azonosító	Orvos név	Orvos telefonszám	Rendelési idő	Rendelő

Vizsgálat							
Vizsgálat azonosító	Vizsgálat dátum	Vizsgálat időpont	Tünet	Gyógyszer név	Gyógyszer gyártó	Beteg taj szám FK	Orvos azonosító FK

A redundancia csökkentése és az adatok integritásának növelése érdekében a három relációt normalizálás módszerével több részre volt szükséges felbontani.

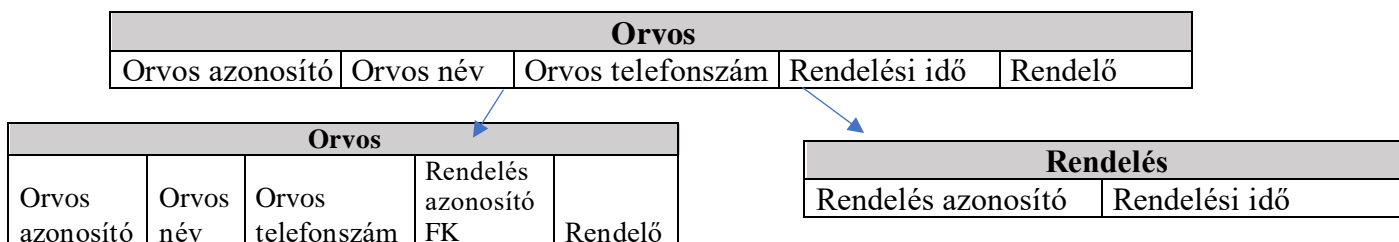
Az elsődleges normálforma elérése érdekében felbontottuk a relációkat rész attribútumokkal, illetve új relációs sémákra úgy, hogy minden egyes attribútumhoz egy érték tartozzon.

- I. A *Beteg* relációban minden attribútum csak egy értéket vehet fel, így reláció megfelel az első normálforma követelményeinek. A betegek azonban különböző allergiákkal rendelkezhetnek ezért a 2. normálforma elérése érdekében az betegekre vonatkozó allergia egy új relációba került szétválasztásra.



Ebben a formában a betegekre vonatkozó sémák nem tartalmaznak tranzitív függéseket, ezért megfelelnek a harmadik normálforma követelményeinek.

- II. Az *Orvos* relációban minden attribútum csak egy értéket vehet fel, így reláció megfelel az első normálforma követelményeinek. Az orvosok azonban különböző rendelési időpontokkal rendelkezhetnek, ezért a 2. normálforma elérése érdekében a rendelési idő egy új relációba került szétválasztásra.



Az orvos és a rendelő között tranzitív függés van ezért a rendelőre vonatkozó adatok egy új relációban kerültek tárolásra.

Orvos				
Orvos azonosító	Orvos név	Orvos telefonszám	Rendelés azonosító FK	Rendelő azonosító

Orvos				
Orvos azonosító	Orvos név	Orvos telefonszám	Rendelés azonosító FK	Rendelő azonosító FK

Rendelő	
Rendelő azonosító	Rendelő neve

Az Orvosra vonatkozó sémák (Orvos, rendelés, rendelő) így megfelelnek a harmadik normálforma követelményeinek.

III. A *Vizsgálat* relációban a tünet és a gyógyszer attribútum esetében szükséges lehetővé tenni, hogy legalább két értéket felvegyenek, ezért esetükben rész attribútumok kerültek kialakításra.

Vizsgálat									
Vizsgálat azonosító	Vizsgálat dátum	Vizsgálat időpont	Tünet	Gyógyszer név	Gyógyszer gyártó	Beteg taj szám FK	Orvos azonosító FK	Táppénz igény	Orvos látogatás célja

Vizsgálat											
Vizsgálat azonosító	Vizsgálat dátum	Vizsgálat időpont	Tünet	Tünet2	Gyógyszer név2	Gyógyszer név2	Gyógyszer gyártó	Beteg taj szám FK	Orvos azonosító FK	Táppénz igény	Orvos látogatás célja

A tünet, a gyógyszer és az orvos látogatás célja attribútumok azonban különböző értékeket is felvehetnek, ezért a 2 normálforma elérése érdekében új relációba kerültek szétválasztásra.

Vizsgálat											
Vizsgálat azonosító	Vizsgálat dátum	Vizsgálat időpont	Tünet	Tünet2	Gyógyszer név2	Gyógyszer név2	Gyógyszer gyártó	Beteg taj szám FK	Orvos azonosító FK	Táppénz igény	Orvos látogatás célja FK

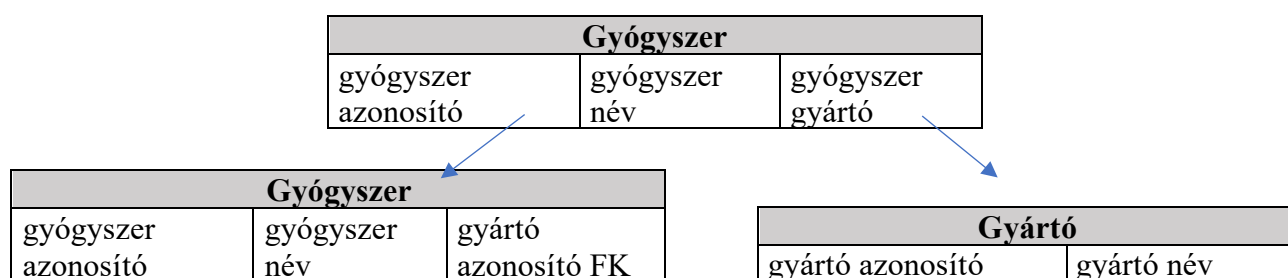
Vizsgálat											
Vizsgálat azonosító	Vizsgálat dátum	Vizsgálat időpont	Tünet	Tünet2	Gyógyszer név2	Gyógyszer név2	Beteg taj szám FK	Orvos azonosító FK	Táppénz igény	Orvos látogatás célja	

Tünet	
Tünet azonosító	tünet

Orvos látogatás Célja	
Orvos látogatás célja azonosító	Orvos látogatás célja

Gyógyszer		
gyógyszer azonosító	gyógyszer név	gyógyszer gyártó

Ebben a formában a gyógyszer sémát a tranzitív függés megszüntetése érdekében szükséges felbontani a gyógyszergyártókra vonatkozó új séma kialakításával

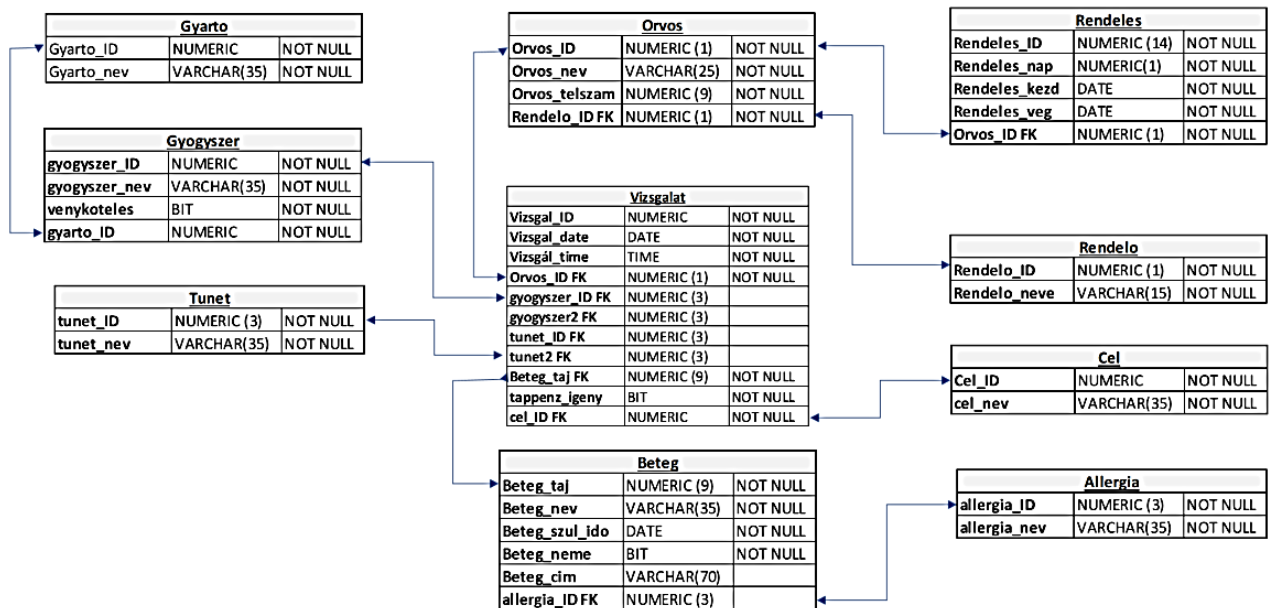


Ebben a formában már a vizsgálat reláció felbontásával keletkezett valamennyi reláció teljesíti a harmadik normálforma követelményeit.

Minden séma tartalmaz egy elsődleges kulcsot, a tény táblákhoz (Orvos, Vizsgálat, Beteg) pedig idegen kulcsként kerültek hozzárendelésre a dimenziótáblák elsődleges kulcsai, megteremtve így a kapcsolatot az egyes sémák között.

III. Az adatbázis felépítése

Normalizálást követően az adatmodell a következő szerkezettel került kialakításra.



Az adatbázisban három tény tábla és hét dimenzió tábla tartalmazza az adatokat, a sémák közötti kapcsolatokat az idegen kulcsok alkalmazása biztosítja.

Az Orvos nevű sémába a rendelőben dolgozó 3 orvos adatai kerültek rögzítésre.

A Rendelő sémába a rendelőben található 3 rendelőhelyiség került rögzítésre.

A Rendelés sémába a három orvos heti rendelési ideje került rögzítésre összesen 15 rekordban

A Beteg nevű sémába 155 beteg adatai kerültek rögzítésre.

Az Allergia nevű sémába a betegeknek felmerülő lehetséges allergiák kerültek tárolásra összesen 6 rekordban.

A Vizsgálata sémába 287 elvégzett vizsgálat adatai kerültek rögzítésre.

A Tünet nevű sémába az orvoshoz forduló betegek lehetséges tünetei kerültek rögzítésre összesen 12 rekordban.

A Gyógyszer nevű sémába az orvosok által felírható gyógyszerek kerültek rögzítésre összesen 26 rekordban.

A Gyártó nevű sémában a gyógyszereket gyártó vállalatok kerültek rögzítésre összesen 5 rekordban.

A Cél nevű sémában az orvosi rendelő felkeresésének az indokai kerültek rögzítésre összesen 7 rekordban.

Az adatbázis létrehozása során készített SQL kódokat az 1. sz. Melléklet tartalmazza.

IV. Lekérdezések

A lekérdezések végrehajtása érdekében szükséges volt két darab nézet létrehozása a gyógyszerek és a tünetek táblák adataiból. A VIEW-k létrehozását tartalmazó kódokat a mellékletek tartalmazzák.

Az elkészített adatbázisból az alábbi lekérdezések kerültek végrehajtásra:

1. Az alábbi SELECT eredménye bemutatja, hogy az egyes orvosok hány vizsgálatot végeztek és ennek során hány beteget láttak el. Van 6 olyan beteg, akik nem csak egy orvosnál jártak, ők minden őket kezelő orvosnál elszámolásra kerültek.

```
SELECT o.Orvos_nev AS 'Az orvos neve', count(v.Vizsgalat_ID) AS 'Az elvégzett vizsgálatok száma', count(DISTINCT v.Beteg_taj) AS 'A kezelt betegek száma'
FROM Vizsgalat v JOIN orvos o on v.Orvos_ID = o.Orvos_ID
GROUP BY o.Orvos_nev
```

	Az orvos neve	Az elvégzett vizsgálatok száma	A kezelt betegek száma
1	dr. Bessenyei Péter	91	51
2	dr. Mészáros Géza	81	46
3	dr. Szilágyi Ilona	115	64

A lekérdezés eredménye szerint dr. Szilágyi Ilona végezte a legtöbb vizsgálatot, és ő kezelte a legtöbb beteget.

2. A Következő SELECT lehetővé teszi egy kiválasztott orvos, jelen esetben dr. Bessenyei Péter tekintetében annak leválogatását, hogy a betegei milyen céllal keresték fel az orvosi rendelőt.

```
SELECT o.Orvos_nev AS 'Orvos neve', c.Cel_nev AS 'Orvos látogatás célja',
count(Vizsgalat_ID) AS 'Orvos látogatás darabszáma'
FROM Vizsgalat v JOIN orvos o on v.Orvos_ID = o.Orvos_ID
JOIN Cel c on v.cel_ID = c.Cel_ID
GROUP BY o.Orvos_nev, c.Cel_nev
HAVING o.Orvos_nev Like 'dr. Bessenyei%'
ORDER BY o.Orvos_nev, count(Vizsgalat_ID) DESC
```

	Orvos neve	Orvos látogatás célja	Orvos látogatás darabszáma
1	dr. Bessenyei Péter	Betegség	31
2	dr. Bessenyei Péter	Kontroll vizsgálat	22
3	dr. Bessenyei Péter	Sport orvosi vizsgálat	19
4	dr. Bessenyei Péter	Gyógyszer felírás	9
5	dr. Bessenyei Péter	Munka alkalmassági vizsgálat	4
6	dr. Bessenyei Péter	Szakrendelés/Kórház beutaló	3
7	dr. Bessenyei Péter	Jogosítvány szerzés/hosszabítás	3

A lekérdezés eredménye szerint 31 páciens betegség kezelése miatt érkezett, 22 keresték fel kontroll vizsgálat, 19-en sportorvosi vizsgálat céljából.

3. A következő SELECT lehetővé teszi, hogy láthatjuk, hogy mely betegek allergiásak:

```
SELECT NULL AS "Beteg_nev" , count(allergia_ID) AS "Allergias betegek szama", NULL AS
"Allergia_neve"
FROM Beteg
UNION
SELECT Beteg_nev , COUNT(allergia_ID), allergia_nev
FROM Beteg
JOIN Allergia using (allergia_ID)
GROUP BY (Beteg_nev)
HAVING COUNT(allergia_ID) > 0
ORDER BY Beteg_nev ASC
```

	Beteg_nev	Allergias betegek szama	Allergia_neve
1	NULL	67	NULL
2	Békés Éva	1	Pollen
3	Béres András	1	Rovarcspép
4	Béres Elvira	1	Pollen
5	Böhm Ignác	1	Pollen
6	Farkas Enikő	1	Glutén
7	Farkas Ágnes	1	Glutén
8	Fazekas Lőrinc	1	Glutén
9	Fekete Margit	1	Pollen
10	Fekete Éva	1	Pollen
11	Halmi Klára	1	Penicilin

A lekérdezés eredménye szerint összesen 67 betegnek van valamilyen allergiája.

4. A következő SELECT lekérdezésben láthatjuk, hogy a betegek hány alkalommal mentek el vizsgálatra

```
SELECT b.beteg_nev, count(s.Vizsgalat_ID) AS 'Latogatasok alkalma'
FROM Vizsgalat as s
JOIN Beteg as b using (Beteg_taj)
GROUP BY s.Beteg_taj
ORDER BY count(s.Vizsgalat_ID) DESC
```

	Beteg_nev	Latogatasok alkalma
1	Pozsgai Béla	5
2	Lakatos Béla	4
3	Lovas Renátó	3
4	Szikla Brigitta	3
5	Szalai Emőke	3
6	Székes Krisztián	3
7	Böhm Ignác	3

Az eredményből látható, hogy Pozsgai Béla illetve Lakatos Béla nevű betegek jártak leggyakrabban az orvosi rendelőben.

5. A következő SELECT az allergiák előfordulási gyakoriságát mutatja be:

```
SELECT allergia_nev AS "Allergia neve:", count (Beteg_nev) AS "Allergiás betegek száma:"  
FROM Beteg  
JOIN Allergia using (allergia_ID)  
GROUP BY (allergia_nev)  
Order by count (Beteg_nev) DESC
```

	Allergia neve:	Allergiás betegek száma:
1	Glutén	28
2	Pollen	21
3	Rovarcspítés	8
4	Penicilin	4
5	Laktóz	3
6	Földimogyoró	3

A lekérdezés eredménye szerint a glutén és pollen allergia a legjellemzőbb a betegek között.

6. Az alábbi SELECT leválogatja az egyes orvosok betegeinek korcsoportonkénti megoszlását, három korcsoport megkülönböztetésével: 1960 előtt született betegek - idősek, 1960 és 1990 között született betegek – középkorúak, 1990 után született betegek – fiatalok. Van 6 olyan beteg, akik nem csak egy orvosnál jártak, ők minden őket kezelő orvosnál elszámolásra kerültek.

```
SELECT Orvos_nev AS 'Orvos neve', Korcsoport, count(Beteg_taj) AS 'Korcsoportba tartozó  
betegek száma'  
FROM  
    (SELECT DISTINCT b.Beteg_taj, b.Beteg_szulido, CASE  
        WHEN Beteg_szulido <'1960.01.01' THEN 'Idős'  
        WHEN Beteg_szulido >'1990.01.01' THEN 'Fiatal'  
        ELSE 'Középkorú'  
        END AS 'Korcsoport', v.Orvos_ID, o.Orvos_nev  
    FROM Beteg b JOIN vizsgalat v on b.Beteg_taj = v.Beteg_taj  
    JOIN orvos o on v.Orvos_ID = o.Orvos_ID)  
GROUP BY Orvos_nev, Korcsoport  
ORDER BY Orvos_nev, count(Beteg_taj) DESC
```

	Orvos neve	Korcsoport	Korcsoportba tartozó betegek száma
1	dr. Bessenyei Péter	Fiatal	25
2	dr. Bessenyei Péter	Középkorú	19
3	dr. Bessenyei Péter	Idős	7
4	dr. Mészáros Géza	Idős	21
5	dr. Mészáros Géza	Középkorú	18
6	dr. Mészáros Géza	Fiatal	7
7	dr. Szilágyi Ilona	Középkorú	29
8	dr. Szilágyi Ilona	Idős	20
9	dr. Szilágyi Ilona	Fiatal	15

A lekérdezés eredménye szerint a legtöbb idős beteg dr. Mészáros Gézánál járt, a legtöbb középkorú beteg dr. Szilágyi Ilonánál, a legtöbb fiatal pedig dr. Bessenyei Péternél.

7. Az alábbi SELECT lehetővé teszi a betegek névtöredéke alapján annak lekérdezését, hogy mikor, milyen céllal, tünetekkel látogattak az orvosi rendelőbe, és ott milyen gyógyszereket írtak fel a számukra:

```
SELECT v.Vizsgal_date AS 'Vizsgálat dátuma', v.Vizsgal_time AS 'Vizsgálat időpontja',
       o.Orvos_nev AS 'Orvos neve', b.Beteg_taj AS 'Beteg tajszáma',
       b.Beteg_nev AS 'Beteg neve', b.Beteg_szulido AS 'Beteg születési ideje',
       v.tappenz_ AS 'Táppézn igény', c.Cel_nev AS 'Rendelő felkeresésének célja',
       t.tunet_nev AS 'Tünet', tv.tunetnev2 AS 'Tünet 2',
       g.gyogyszer_nev AS 'gyógyszer', gv.gyogyszer_nev2 AS 'gyógyszer2'
FROM Vizsgalat v LEFT join Beteg b on v.Beteg_taj=b.Beteg_taj
Left join Cel c ON v.cel_ID=c.cel_ID
Left join Tunet t on v.tunet_ID = t.tunet_ID
Left join tunetView tv on v.tunet2 = tv.tunet_ID
Left join Gyogyszer g on v.gyogyszer_ID= g.gyogyszer_ID
Left join gyogyszerView gv on v.gyogyszer2 = gv.gyogyszer_ID
Left join Orvos o on v.Orvos_ID=o.Orvos_ID
Where b.Beteg_nev Like "Horváth%"
```

A lekérdezés eredménye szerint öt alkalommal került sor Horváth családnévű személy vizsgálatára a rendelőben:

	Vizsgálat dátuma	Vizsgálat időpontja	Orvos neve	Beteg tajszáma	Beteg neve	Beteg születési ideje	Táppézn igény	Rendelő felkeresésének célja	Tünet	Tünet 2	gyógyszer	gyógyszer2
1	2023.01.03	14:35	dr. Szilágyi Ilona	112233686	Horváth Ignác	1967.08.23	0	Kontroll vizsgálat	Nincs tünet	NULL	NULL	NULL
2	2023.01.17	12:50	dr. Mészáros Géza	476786584	Horváth Lajos	1980.04.10	0	Munka alkalmassági vizsgálat	Nincs tünet	NULL	NULL	NULL
3	2023.01.27	14:45	dr. Mészáros Géza	476786584	Horváth Lajos	1980.04.10	1	Betegség	Fejfájás	Nátha	Algonflex	Xilomare
4	2023.01.18	13:23	dr. Bessenyei Péter	476786585	Horváth Viktor	1977.04.10	1	Betegség	Láz	Köhögés	Algopyrin	Paxirasol
5	2023.01.24	08:40	dr. Bessenyei Péter	476786585	Horváth Viktor	1977.04.10	0	Kontroll vizsgálat	Nincs tünet	NULL	NULL	NULL

8. A következő SELECT lehetővé teszi az orvosok által felírt gyógyszerek lekérdezését gyakoriság szerint csökkenő sorrendben:

```
SELECT gyogyszer_nev AS 'Gyógyszer neve', count(beteg_taj) AS 'Hányszor írták fel:'  
FROM(  
    SELECT v.Beteg_taj, v.gyogyszer_ID, g.gyogyszer_nev  
    FROM Vizsgalat v LEFT join Gyogyszer g on v.gyogyszer_ID=g.gyogyszer_ID  
    UNION ALL  
    SELECT v.Beteg_taj, v.gyogyszer2, g.gyogyszer_nev  
    FROM Vizsgalat v LEFT join Gyogyszer g on v.gyogyszer2=g.gyogyszer_ID)  
WHERE gyogyszer_nev<> "  
GROUP BY gyogyszer_nev  
ORDER BY count(beteg_taj) DESC
```

	Gyógyszer neve	Hányszor írták fel:
1	Eurovit C vitamin	39
2	Algoflex	36
3	Algopyrin	34
4	Rinathiol	15
5	Paxirasol	12
6	Contramal	12
7	Ambroxol	12
8	Xilomare	10
9	Insulin Aspart	10
10	Alfetim	10
11	Quarelin	7
12	Metformin	7
13	Amlipin	7
14	Magne B6	6
15	Diclofenat	6

A lekérdezés eredménye szerint a leggyakrabban C vitamint Algoflex és Algopyrin gyógyszereket írták fel az orvosok

9. A következő SELECT lehetővé teszi annak lekérdezését, hogy melyek azok az utcák, ahol a legtöbb (jelenlegi beállítás szerint több mint 4) azonos tünetet mutató beteg lakik (ebből esetleg lehet járvány kitörésére következtetni):

```
SELECT Város,SUBSTR(utca2,0, (instr(utca2, " " ))) AS 'utca', tunet_nev AS 'Tünet',
COUNT(Beteg_taj) AS 'A tüneteket hordozó betegek száma'
FROM
(SELECT Beteg_taj, tunet_ID, tunet_nev,Beteg_cime, SUBSTR(Beteg_cime, 0,
INSTR(Beteg_cime, ',')) AS 'Város',
SUBSTR(Beteg_cime, (INSTR(Beteg_cime, ',')+2) ) AS 'utca2'
FROM (SELECT v.Beteg_taj, v.tunet_ID, t.tunet_nev, b.Beteg_cime
FROM
(SELECT Beteg_taj, tunet_ID
FROM Vizsgalat
UNION ALL

SELECT Beteg_taj, tunet2
FROM Vizsgalat) v LEFT JOIN Beteg b
on b.Beteg_taj = v.Beteg_taj
LEFT JOIN tunet t on t.tunet_ID = v.tunet_ID
))
WHERE tunet_ID > 1
GROUP BY Város, utca, tunet_nev
HAVING COUNT(Beteg_taj) > 3
ORDER BY COUNT(Beteg_taj) DESC, utca
```

	Város	utca	Tünet	A tüneteket hordozó betegek száma
1	Diósd	Duna	Fejfájás	10
2	Diósd	Fenyősor	Láz	6
3	Diósd	Gizella	Fejfájás	6
4	Diósd	Gizella	Köhögés	6
5	Diósd	Szabadság	Köhögés	5
6	Diósd	Teréz	Köhögés	5
7	Diósd	Teréz	Láz	5
8	Diósd	Duna	Köhögés	4
9	Diósd	Duna	Magas vérnyomás	4
10	Diósd	Petőfi	Láz	4
11	Diósd	Zsuzsa	Köhögés	4
12	Diósd	Zsuzsa	Láz	4
13	Diósd	Éva	Cukorbetegség	4

A lekérdezés eredménye szerint a legtöbb azonos tünetet mutató beteg a Diósd, Duna, Fenyősor illetve Gizella utcákban lakik.

10. A következő SELECT lehetővé teszi, annak lekérdezését, hogy az egyes orvosok esetében mely betegek igényelték a legtöbb alkalommal táppénzt.

```
SELECT DISTINCT v.Beteg_taj,
               b.Beteg_nev,
               o.Orvos_nev,
               count (tappenzenz_) OVER (
                 PARTITION BY b.Beteg_nev
                 ORDER by Orvos_ID
                 RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
               ) AS "Tappenzenz igeenylesek szama"
FROM Vizsgalat as v
JOIN Beteg as b using (Beteg_taj)
JOIN Orvos as o using (Orvos_ID)
WHERE tappenzenz_ = 1
ORDER BY "Tappenzenz igeenylesek szama" DESC
```

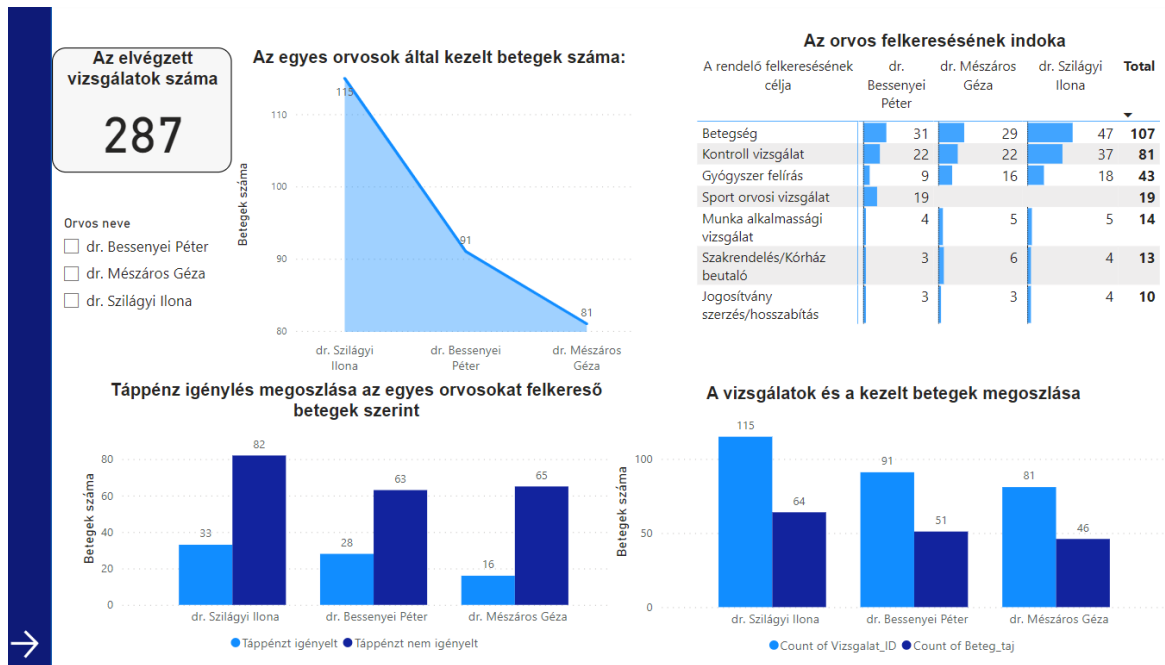
	Beteg_taj	Beteg_nev	Orvos_nev	Tappenzenz igeenylesek szama
1	476686567	Szemes Klára	dr. Bessenyei Péter	3
2	476786578	Vörös Erika	dr. Bessenyei Péter	3
3	121545683	Böhm Zsolt	dr. Szilágyi Ilona	2
4	775675789	Lengyel Péter	dr. Mészáros Géza	2
5	734545684	Pozsgai Béla	dr. Szilágyi Ilona	2
6	734545684	Pozsgai Béla	dr. Bessenyei Péter	2
7	476565466	Szűcs Vera	dr. Bessenyei Péter	2
8	476786599	Borsodi László	dr. Szilágyi Ilona	1
9	474564465	Békés Éva	dr. Bessenyei Péter	1
10	473564455	Béres András	dr. Szilágyi Ilona	1

A lekérdezésben összesítve láthatjuk, hogy maximum három alkalommal kért egy beteg táppénzt ugyanattól az orvostól, de volt olyan beteg is Pozsgai Béla, aki több orvostól is több alkalommal kért táppénzt.

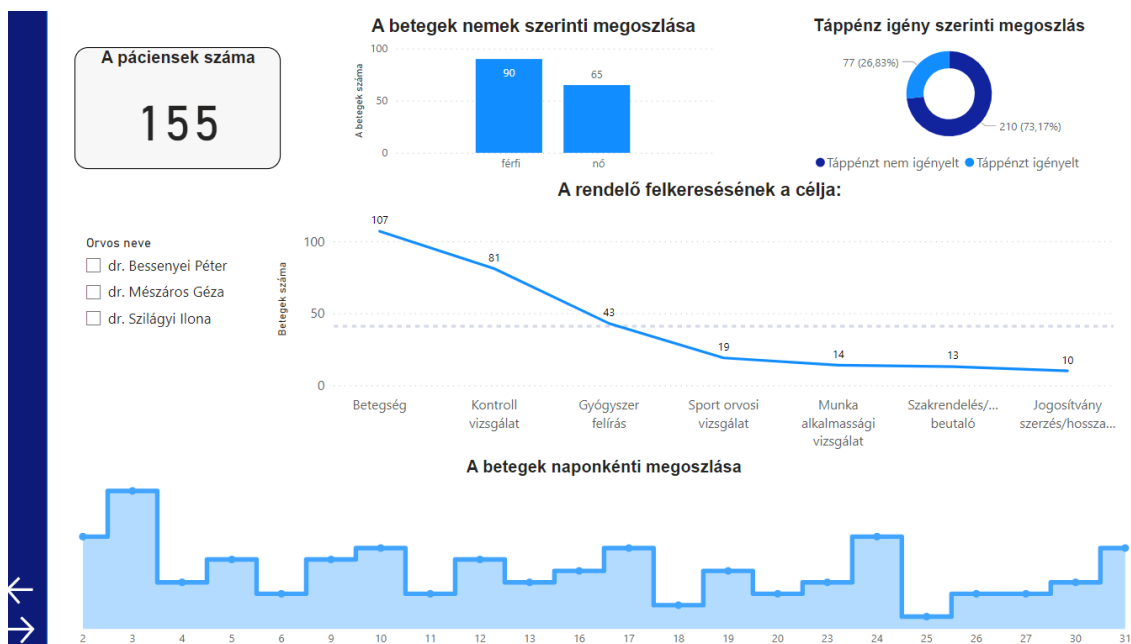
V. Kimutatások-Riportok

PowerBI szoftver segítségével az alábbi vizualizációkat készítettük az adatok ábrázolásához:

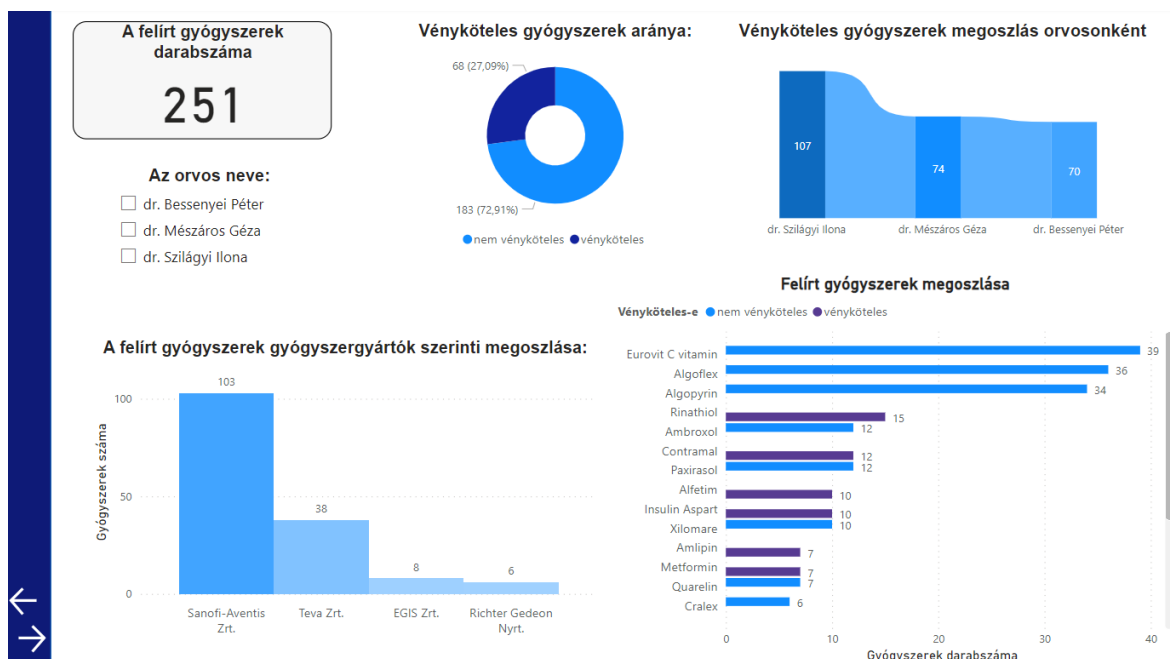
1. Az orvosokra vonatkozó adatok megjelenítése: Az alábbi riport bemutatja, hogy az egyes orvosok hány vizsgálatot végeztek, elvégzett vizsgálatok számát, milyen céllal keresték fel a rendelőt, igényelték-e táppénzt.



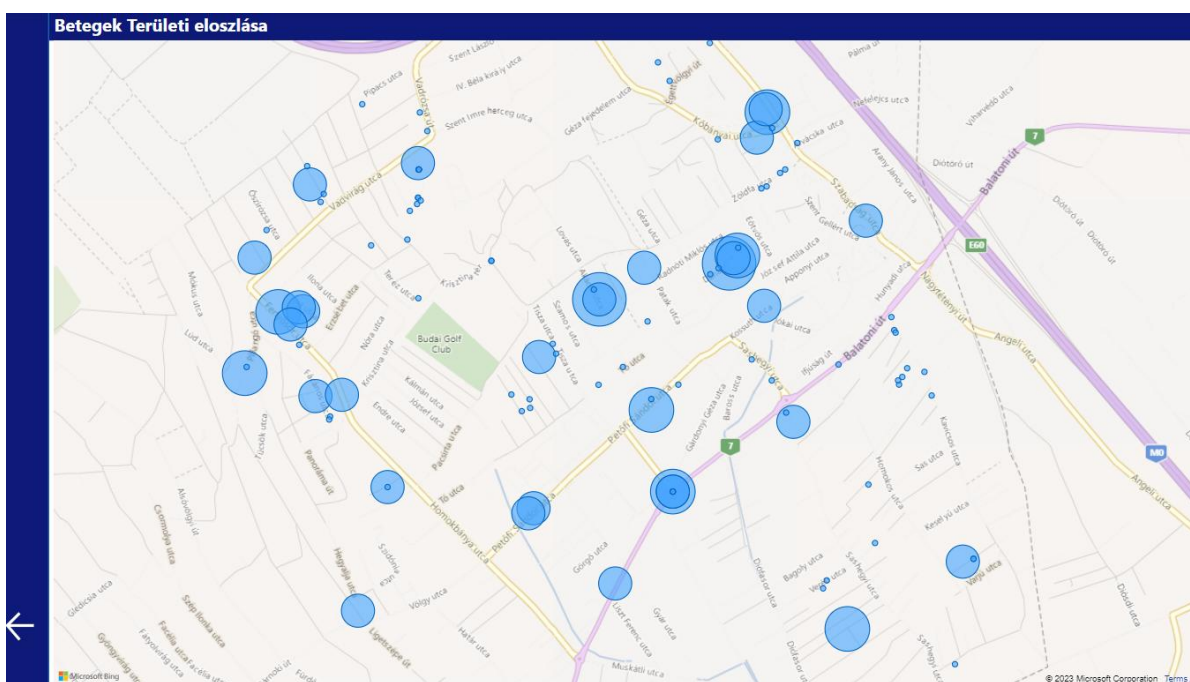
2. A betegekre vonatkozó adatok megjelenítése: Az alábbi riport bemutatja a betegek számát, azok nemekénti eloszlását, az egyes napokon hány beteg járt a rendelőben, milyen céllal keresték fel a rendelőt, igényelték-e táppénzt.
Az adatok tovább szűrhetők a kezelőorvos szerint.



3. A gyógyszerekre vonatkozó adatok megjelenítése: Az alábbi riport bemutatja, az összes felírt gyógyszer darabszámát, ezen belül a felírt gyógyszerek megoszlását, azon belül, hogy vénykötelesek-e. Utána láthatjuk, hogy a vényköteles gyógyszerek felírása hogyan oszlik meg orvosonként. Láthatunk gyártó szerinti eloszlást is a felírt gyógyszerek között, és végül magát a gyógyszerek eloszlását láthatjuk. Az adatok tovább szűrhetőek a kezelőorvos szerint.



4. A betegek területi eloszlására vonatkozó adatok megjelenítése: A betegek területi eloszlása, a településen belül, térképen ábrázolva. Ezen kívül láthatjuk mely utcák azok, ahonnan a legtöbbben látogatják az orvosi rendelőt.



VI. Mellékletek

Az adatbázis létrehozása során készített SQL kód:

```
CREATE TABLE IF NOT EXISTS "Allergia" (  
    "allergia_ID" NUMERIC(3) NOT NULL,  
    "allergia_nev" VARCHAR(35) NOT NULL,  
    PRIMARY KEY("allergia_ID")  
);  
CREATE TABLE IF NOT EXISTS "Beteg" (  
    "Beteg_taj" NUMERIC(9) NOT NULL UNIQUE,  
    "Beteg_nev" VARCHAR(35) NOT NULL,  
    "Beteg_szulido" DATE NOT NULL,  
    "Beteg_neme" BIT NOT NULL,  
    "Beteg_cime" VARCHAR(70),  
    "allergia_ID" NUMERIC(3),  
    PRIMARY KEY("Beteg_taj"),  
    FOREIGN KEY("allergia_ID") REFERENCES "Allergia"("allergia_ID")  
);  
CREATE TABLE IF NOT EXISTS "Rendelo" (  
    "Rendelo_ID" NUMERIC(1) NOT NULL UNIQUE,  
    "Rendelo_neve" VARCHAR(15) NOT NULL,  
    PRIMARY KEY("Rendelo_ID")  
);  
CREATE TABLE IF NOT EXISTS "Orvos" (  
    "Orvos_ID" INTEGER NOT NULL UNIQUE,  
    "Orvos_nev" VARCHAR(25) NOT NULL,  
    "Orvos_telszam" NUMERIC(9) NOT NULL,  
    "Rendelo_ID" NUMERIC(1) NOT NULL,  
    PRIMARY KEY("Orvos_ID" AUTOINCREMENT),  
    FOREIGN KEY("Rendelo_ID") REFERENCES "Rendelo"("Rendelo_ID")  
);  
CREATE TABLE IF NOT EXISTS "Tunet" (  
    "tunet_ID" INTEGER NOT NULL,  
    "tunet_nev" VARCHAR(35) NOT NULL UNIQUE,  
    PRIMARY KEY("tunet_ID" AUTOINCREMENT)  
);  
CREATE TABLE IF NOT EXISTS "Rendeles" (  
    "Rendeles_ID" INTEGER NOT NULL,  
    "Rendeles_nap" NUMERIC(1) NOT NULL,  
    "Rendeles_kezd" DATE NOT NULL,  
    "Rendeles_veg" DATE NOT NULL,  
    "Orvos_ID" INTEGER NOT NULL,  
    PRIMARY KEY("Rendeles_ID" AUTOINCREMENT),  
    FOREIGN KEY("Orvos_ID") REFERENCES "Orvos"("Orvos_ID")  
);
```

```

CREATE TABLE IF NOT EXISTS "Cel" (
    "Cel_ID"      INTEGER NOT NULL UNIQUE,
    "Cel_nev"     VARCHAR(35) NOT NULL UNIQUE,
    PRIMARY KEY("Cel_ID" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "Gyarto" (
    "gyarto_ID"   INTEGER NOT NULL UNIQUE,
    "gyarto_nev"  VARCHAR(35) NOT NULL UNIQUE,
    PRIMARY KEY("gyarto_ID" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "Gyogyszer" (
    "gyogyszer_ID"    INTEGER NOT NULL,
    "gyogyszer_nev"   VARCHAR(35) NOT NULL UNIQUE,
    "gyarto_ID"       NUMERIC,
    "veny_koteles"    BIT,
    PRIMARY KEY("gyogyszer_ID" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "Vizsgalat" (
    "Vizsgalat_ID"    INTEGER NOT NULL UNIQUE,
    "Vizsgal_date"    DATE NOT NULL,
    "Vizsgal_time"    TIME NOT NULL,
    "Orvos_ID"        NUMERIC(1) NOT NULL,
    "gyogyszer_ID"    NUMERIC(3),
    "gyogyszer2"      NUMERIC(3),
    "Beteg_taj"       NUMERIC(9) NOT NULL,
    "tunet_ID"        NUMERIC(3),
    "tunet2"          NUMERIC(3),
    "tappenz_"        BIT,
    "cel_ID"          NUMERIC,
    PRIMARY KEY("Vizsgalat_ID" AUTOINCREMENT),
    FOREIGN KEY("Orvos_ID") REFERENCES "Orvos"("Orvos_ID"),
    FOREIGN KEY("gyogyszer2") REFERENCES "Gyogyszer"("gyogyszer_ID"),
    FOREIGN KEY("tunet_ID") REFERENCES "Tunet"("tunet_ID"),
    FOREIGN KEY("tunet2") REFERENCES "Tunet"("tunet_ID"),
    FOREIGN KEY("gyogyszer_ID") REFERENCES "Gyogyszer"("gyogyszer_ID"),
    FOREIGN KEY("Beteg_taj") REFERENCES "Beteg"("Beteg_taj")
)

```

A lekérdezésekhez szükséges két darab nézetet létrehozó kódok:

- CREATE VIEW tunetView
(tunet_ID, tunetnev2)
AS
SELECT tunet_ID, tunet_nev
FROM Tunet
- CREATE VIEW gyogyszerView
(gyogyszer_ID, gyogyszer_nev2, gyarto_ID, veny_koteles)
AS
SELECT gyogyszer_ID, gyogyszer_nev, gyarto_ID, veny_koteles
FROM Gyogyszer