

JEGYZŐKÖNYV

Adatbázis rendszerek I.

Féléves Feladat

Egészségügyi rendszer

Készítette: **Tán Gergő**

Neptunkód: **BLCL20**

Dátum: **2025. 05. 13.**

Sárospatak, 2025

Tartalomjegyzék

1.	Feladat leírás	3
1.1	Az adatbázis ER modell	4
1.2	Relációs modell	5
1.3	Táblák létrehozása	6
1.4	Táblák feltöltése	7
1.5	Lekérdezések	9

1. Feladat leírás

Az „egészségügyi rendszer” adatbázisa egy olyan strukturált információs rendszer, amelynek célja az orvosi ellátással kapcsolatos adatok pontos, biztonságos és naprakész nyilvántartása. Ez a rendszer lehetővé teszi a páciensek, orvosok, időpontfoglalások, kezelések és receptek közötti kapcsolatok átlátható kezelését. Az adatbázis egyaránt támogatja a napi működést (pl. időpontkezelés) és a hosszabb távú nyilvántartási, statisztikai és adminisztratív feladatokat is.

A rendszer központi szereplője a „Páciens”, akinek adatai között szerepel a neve, társadalombiztosítási (TAJ) száma, születési ideje, lakcíme és elérhetőségei. Egy páciens többféle egészségügyi szolgáltatást is igénybe vehet, ezért az adatbázis lehetővé teszi, hogy egy páciens több „Időpontot” is foglalhasson különböző orvosokhoz. Az időpontokhoz tartozó adatok között megtalálható a dátum, az időpont, valamint az adott foglалás állapota (például megerősített, lemondott, vagy várakozó).

Az időpontok kapcsolódnak az Orvoshoz, aki a vizsgálatot vagy kezelést végzi. Az orvosok különböző szakterületeken dolgoznak, és rendelési idejük, valamint elérhetőségük is eltérő lehet. A kapcsolatokat úgy alakítottuk ki, hogy egy orvoshoz több időpont és így több páciens is tartozhat, ugyanakkor egy páciens több orvossal is kapcsolatba kerülhet. Ez az N\ : M kapcsolat az Időpont entitáson keresztül valósul meg.

Egy időponthoz konkrét Kezelések is kapcsolódhatnak, amelyek részletesen leírják a nyújtott szolgáltatást, annak típusát és költségét. Ez lehet például egy rutin vérvétel, fizioterápia vagy éppen sebészeti beavatkozás. A kezelések az időpontokhoz kapcsolódnak, mivel a vizsgálat vagy beavatkozás az adott időintervallumban történik meg.

Az orvos egy másik fontos feladata a Receptek kiállítása. Egy orvos több receptet is kiadhat, amelyek különböző páciensekhez tartoznak. A receptek adatai közé tartozik a kiállítás és lejárati dátuma, valamint a gyógyszerlista, amely lehet összetett és akár több értéket is tartalmazhat (pl. többféle gyógyszer kombinációja).

A rendszer kapcsolati modellje biztosítja az adatintegritást és lehetőséget ad a lekérdezésekre, például: "Milyen kezeléseken vett részt egy adott páciens az elmúlt évben?", "Melyik orvos írt ki legtöbbször antibiotikumot?" vagy "Hány páciens látogatott meg egy adott szakorvost egy hónap alatt?". A többféle kapcsolat (1:1, 1\ : N, N\ : M), kulcsok (elsődleges és idegen), összetett

és többértékű attribútumok révén az adatmodell képes rugalmasan kezelni a valós egészségügyi folyamatokat.

Ez a rendszer nemcsak a napi orvosi működést segíti, hanem az adminisztratív és döntéstámogatási folyamatokat is, hozzájárulva a gyorsabb ügyintézéshez, pontosabb diagnosztikához és jobb betegellátáshoz.

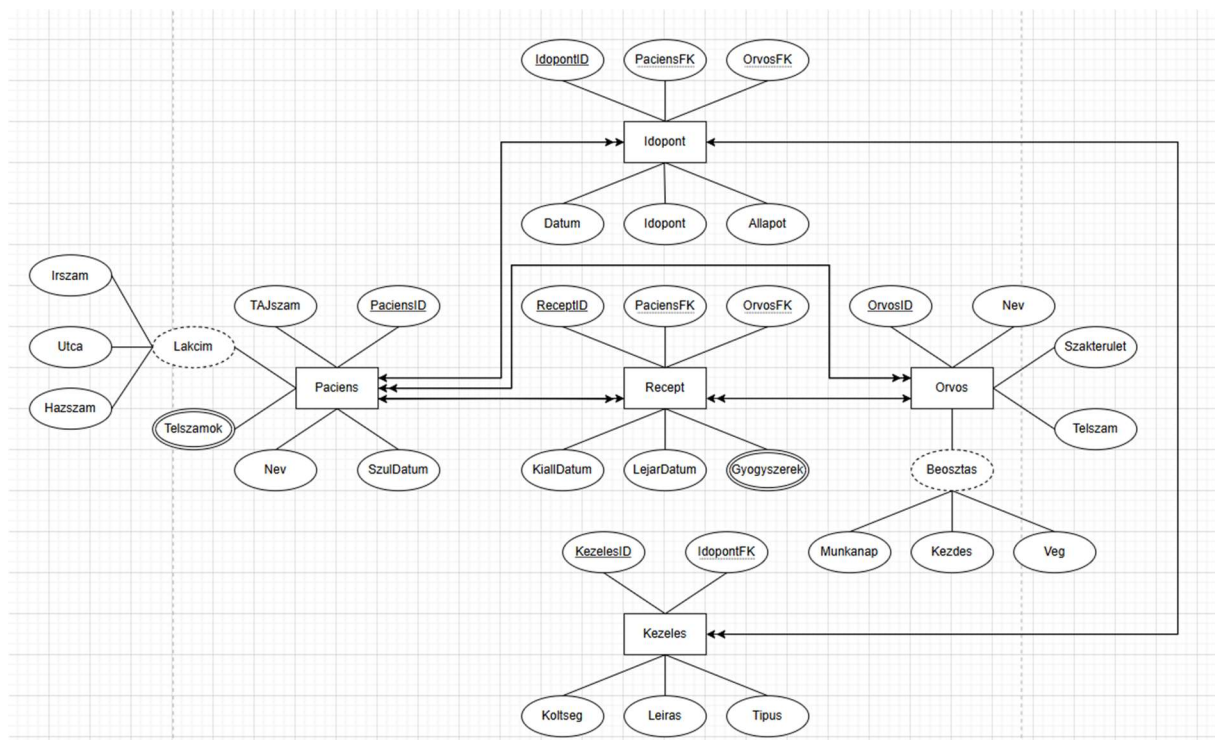
1.1 Az adatbázis ER modell

Az adatbázis tervezését a valós egészségügyi folyamatok logikai felépítése alapján végeztem, figyelembe véve a páciens-orvos kapcsolatokat, a kezelések és receptek adminisztrációját.

Célom az volt, hogy minden fontos szereplő és tevékenység önálló entitásként jelenjen meg, egyértelmű és hatékony kapcsolatokkal.

A rendszer kialakításánál ügyeltem arra is, hogy a követelményeknek megfelelően legyenek jelen kulcsok, összetett és többértékű attribútumok, valamint különböző kapcsolatfajták (1:1, 1:N, N:M).

A modell rugalmas, bővíthető és jól leképezhető valós alkalmazási környezetben is.



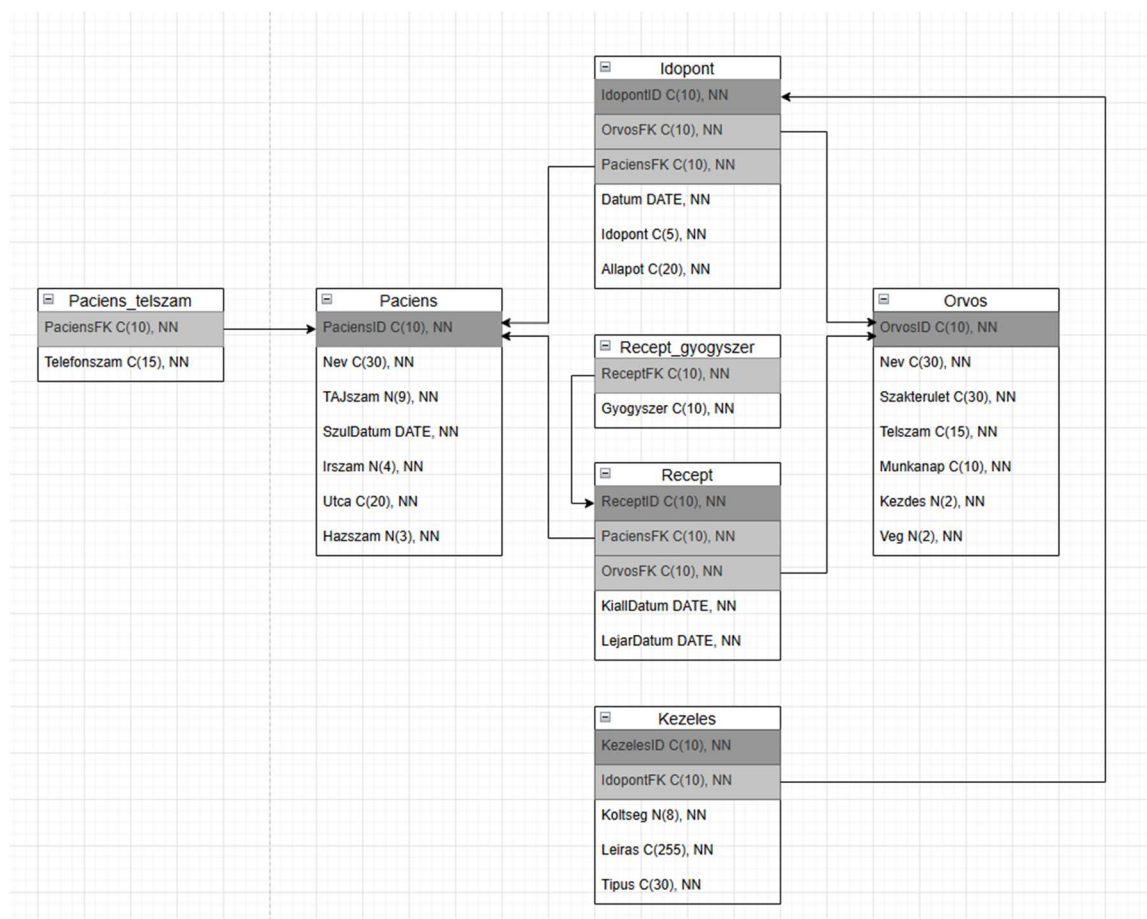
1.2 Relációs modell

Ez a relációs adatbázismodell egy olyan egészségügyi nyilvántartó rendszert ábrázol, amelyet orvosok, páciensek, időpontok, receptek és kezelések adatainak rendszerezésére hoztam létre. Igyekeztem minden fontos entitást külön táblában megjeleníteni, és ezek között logikus, jól átlátható kapcsolatokat kialakítani.

Az orvosokat és pácienseket egyedi azonosítókkal azonosítom, és az időpontokat ezekhez a szereplőkhöz rendeltem hozzá. Azt is fontosnak tartottam, hogy a receptekhez kapcsolódó gyógyszerek, valamint a páciensek telefonszámai külön táblákban szerepeljenek, ezzel biztosítva a normalizáltságot és az adatok redundanciamentes tárolását.

A kezelésekhez tartozó költségeket, leírásokat és típusokat szintén tárolom, és minden kezelés egy időponthoz kapcsolódik, ezáltal pedig közvetetten egy orvoshoz és egy pácienshez is. Az adattípusokat gondosan választottam ki, figyelembe véve az adatok jellegét – például dátumokat 'DATE' típussal, neveket és azonosítókat 'CHAR' mezőkkel tárolok.

Ezzel a modellel egy olyan átlátható rendszert szerettem volna létrehozni, amely könnyen kezelhető, bővíthető, és minden szükséges információt képes tárolni egy orvosi rendelő működéséhez.



1.3 Táblák létrehozása

Az előzetesen megtervezett struktúra szerint létrehoztam az adatbázist, majd az összes szükséges táblát, a megfelelő mezőnevekkel, adattípusokkal és integritási feltételekkel.

Mezőnevek és adattípusok

A mezők elnevezésénél törekedtem az egyértelmű, jól értelmezhető névadásra (pl. OrvosID, Nev, TAJszam, LejarDatum), az adattípusokat a tárolandó adatokhoz igazítottam:

CHAR(n): fix hosszúságú karakterlánc, pl. azonosítókhoz (OrvosID, PaciensID, ReceptID) és név elemekhez.

INT(n): számokhoz (pl. TAJszam, Koltseg, Irszam, Kezdes, Veg).

DATE: dátum mezőkhöz, mint pl. SzulDatum, Datum, KiallDatum

Integritási feltételek

A lényeges adatkapcsolatok, megszorítások biztosítása érdekében:

Elsődleges kulcsok (PRIMARY KEY): minden tábla első mezője az egyedi azonosítója. Ez alapján érhető el a különböző kapcsolatok során, illetve egyediséget biztosít, még egy ilyen nincs a táblában.

Idegen kulcsok (FOREIGN KEY): biztosítják a táblák közti kapcsolatokat, az elsődleges kulcsokra hivatkozik, pl. OrvosFK, PaciensFK, IdopontFK, stb.

NOT NULL kikötés minden fontos mezőnél, ahol szükséges az adat megléte. Jelen esetben mindegyik mező rendelkezik ezzel a tulajdonsággal, ugyanis bármely adat hiányossága nem kívánt félreértéshez, hibához vezethet.

```

MariaDB [(none)]> use SemBLCL20;
Database changed
MariaDB [SemBLCL20]> create table Orvos (OrvosID char(10) not null primary key, Nev char(30) not null, Szakterulet char(30) not null, Telszam char(15) not null, Munkanap char(10) not null, Kezdes int(2) not null, Veg int(2) not null);
Query OK, 0 rows affected (0.007 sec)

MariaDB [SemBLCL20]> create table Paciens (PaciensID char(10) not null primary key, Nev char(30) not null, TAJszam int(9) not null, SzulDatum DATE not null, Irszam int(4) not null, Utca char(20) not null, Hazszam int(3) not null);
Query OK, 0 rows affected (0.007 sec)

MariaDB [SemBLCL20]> create table Idopont (IdopontID char(10) not null primary key, OrvosFK char(10) not null, PaciensFK char(10) not null, Datum DATE not null, Idopont char(5) not null, Allapot char(20) not null, Foreign key (OrvosFK) references Orvos(OrvosID), Foreign key (PaciensFK) references Paciens(PaciensID));
Query OK, 0 rows affected (0.021 sec)

MariaDB [SemBLCL20]> create table Recept (ReceptID char(10) not null primary key, OrvosFK char(10) not null, PaciensFK char(10) not null, Kialldatum DATE not null, LejarDatum DATE not null, Foreign key (OrvosFK) references Orvos(OrvosID), Foreign key (PaciensFK) references Paciens(PaciensID));
Query OK, 0 rows affected (0.019 sec)

MariaDB [SemBLCL20]> create table Kezeles (KezelesID char(10) not null primary key, IdopontFK char(10) not null, Koltsag int(8) not null, Leiras char(255) not null, Tipus char(30) not null, Foreign key (IdopontFK) references Idopont(IdopontID));
Query OK, 0 rows affected (0.028 sec)

MariaDB [SemBLCL20]> create table Recept_gyogyszer (ReceptFK char(10) not null, Gyogyszer char(10), Foreign key (ReceptFK) references Recept(ReceptID));
Query OK, 0 rows affected (0.021 sec)

MariaDB [SemBLCL20]> create table Paciens_telszam (PaciensFK char(10) not null, Telefonszam char(15), Foreign key (PaciensFK) references Paciens(PaciensID));
Query OK, 0 rows affected (0.020 sec)

```

1.4 Táblák feltöltése

```

MariaDB [SemBLCL20]> INSERT INTO Paciens (PaciensID, Nev, TAJszam, SzulDatum, Irszam, Utca, Hazszam) VALUES ('PA001', 'Kovács János', 123456789, '1985-05-12', 1051, 'Petőfi Sándor', 12), ('PA002', 'Nagy Éva', 987654321, '1990-08-23', 2030, 'Kossuth Lajos', 45), ('PA003', 'Tóth Gábor', 456789123, '1978-11-15', 4025, 'Deák Ferenc', 7), ('PA004', 'Szabó Anna', 321654987, '1995-03-30', 6722, 'Ady Endre', 23), ('PA005', 'Horváth Péter', 789123456, '1982-07-18', 7621, 'Arany János', 9);
Query OK, 5 rows affected (0.003 sec)
Records: 5 Duplicates: 0 Warnings: 0

```

A páciensek táblába bekerültek az alapadatok: minden pácienshez megadtam a nevét, TAJ-számát, születési dátumát, lakcímét (irányítószám, utca, házszám). Például Kovács János a Petőfi Sándor utca 12. szám alatt lakik, TAJ-száma „123456789”.

```

MariaDB [SemBLCL20]> INSERT INTO Orvos (OrvosID, Nev, Szakterulet, Telszam, Munkanap, Kezdes, Veg) VALUES ('ORV001', 'Dr. Kovács Péter', 'Belgyógyászat', '06201234567', 'Hétfő', 8, 16), ('ORV002', 'Dr. Kiss Anna', 'Bőrgyógyászat', '06209876543', 'Kedd', 9, 17), ('ORV003', 'Dr. Tóth Gábor', 'Sebészet', '06301239876', 'Szerda', 7, 15), ('ORV004', 'Dr. Nagy Zsuzsa', 'Nőgyógyászat', '06202345678', 'Csütörtök', 8, 14), ('ORV005', 'Dr. Szabó László', 'Fül-orr-gégészlet', '06203456789', 'Péntek', 10, 18);
Query OK, 5 rows affected (0.004 sec)
Records: 5 Duplicates: 0 Warnings: 0

```

Az orvosok táblájába felvettem az orvosok nevét, szakterületét, elérhetőségét és munkaidejüket. Dr. Kovács Péter például belgyógyász, hétfőnként 8-tól 16-ig rendel, telefonszáma „+36 28 123 4567”.

```

MariaDB [SemBLCL20]> INSERT INTO Idopont (IdopontID, OrvosFK, PaciensFK, Datum, Idopont, Allapot) VALUES ('ID001', 'ORV001', 'PA001', '2025-8-15', '08:30', 'Elfogadva'), ('ID002', 'ORV002', 'PA002', '2025-10-16', '09:15', 'Lemondva'), ('ID003', 'ORV003', 'PA003', '2025-09-01', '10:00', 'Befizetve'), ('ID004', 'ORV004', 'PA004', '2025-5-30', '13:45', 'Elfogadva'), ('ID005', 'ORV005', 'PA005', '2025-06-19', '15:30', 'Fizetésre vár');
Query OK, 5 rows affected (0.004 sec)
Records: 5 Duplicates: 0 Warnings: 0

```


Az időpontok táblájában rögzítettem a betegjegyzéseket, hogy melyik páciens mikor és melyik orvosnál volt időpontja. Például Kovács János 2025. május 15-én 8:30-kor Dr. Kovács Péternél volt vizsgálaton, az időpont állapota Elfogadva.

```
MariaDB [SemBLCL20]> INSERT INTO Kezeles (KezelesID, IdopontFK, Koltseg, Leiras, Tipus) VALUES ('BEL001', 'ID001', 20000, 'Teljes vérkép és biokémiai vizsgálat cukorbetegség szűrésére', 'Laboratóriumi vizsgálat'), ('BOR002', 'ID002', 35000, 'Anyajegy vizsgálat dermatoszkóppal és esetleges eltávolítás', 'Bőrrákszűrés'), ('SEB003', 'ID003', 180000, 'Laparoszkópos epehólyag-eltávolítás', 'Minimál invazív segítség'), ('NOG004', 'ID004', 25000, 'Éves szűrővizsgálat méhnyakrák megelőzésére', 'Női szűrővizsgálat'), ('FUL005', 'ID005', 40000, 'Endoszkópos orr- és arcüreg vizsgálat krónikus orrdugulás esetén', 'Felső légúti endoszkópia');
Query OK, 5 rows affected (0.003 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

A kezelések táblájába felvittem az elvégzett orvosi beavatkozásokat, költségeikkel együtt. Laparoszkópos epehólyag-eltávolítás például 180.000 Ft-ba került, és minimál invazív sebészeti kezelésként lett rögzítve.

```
MariaDB [SemBLCL20]> INSERT INTO Recept (ReceptID, OrvosFK, PAcienstFK, KiallDatum, LejarDatum) VALUES ('REC001', 'ORV001', 'PA001', '2025-04-22', '2025-05-22'), ('REC002', 'ORV002', 'PA002', '2025-04-01', '2025-05-01'), ('REC003', 'ORV003', 'PA003', '2025-05-09', '2025-06-09'), ('REC004', 'ORV004', 'PA004', '2025-05-01', '2025-06-01'), ('REC005', 'ORV005', 'PA005', '2025-04-30', '2025-05-30');
Query OK, 5 rows affected (0.004 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

A receptek táblájában látható, hogy az orvosok kiállítottak gyógyszereket a pácienseknek. Minden recepthez megadtam a kiállítás és lejárát dátumát. Például REC001 azonosítójú receptet Dr. Kovács Péter írta ki 2025. április 22-én, és 2025. május 22-én jár le.

```
MariaDB [SemBLCL20]> INSERT INTO Paciens_telszam (PacienstFK, Telefonszam) VALUES ('PA001', '+36301234567'), ('PA001', '+3619876543'), ('PA002', '+36209876543'), ('PA003', '+36701234567'), ('PA003', '+36203456789'), ('PA004', '+36307654321'), ('PA005', '+36201112233');
Query OK, 7 rows affected (0.003 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

A páciens telefonszámok táblájában minden pácienshez felvettem elérhetőségeket. Kovács Jánosnak két telefonszáma van: egy mobil („+36 30 123 4567”) és egy vezetékes („+36 1 987 6543”).

```
MariaDB [SemBLCL20]> INSERT INTO Recept_gyogyszer (ReceptFK, Gyogyszer) VALUES ('REC001', 'IBUPROFEN'), ('REC001', 'PARACET'), ('REC002', 'AMLODIPIN'), ('REC003', 'ATROVENT'), ('REC003', 'SALBUTAM'), ('REC003', 'PREDNISON'), ('REC004', 'METFORMIN'), ('REC004', 'GLIBENKL'), ('REC005', 'CIPROFLOX');
Query OK, 9 rows affected (0.005 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

Végül a recept-gyógyszer kapcsolótáblában rögzítettem, hogy melyik recepthez milyen gyógyszert írtak fel. Például a REC001 azonosítójú recepthez Ibuprofent és Paracetamolt rendeltek, míg a REC003-hoz Atroventet, Salbutamolt és Prednizont.

1.5 Lekérdezések

I. Kérdezze le az összes páciens nevét, TAJ számát, listázza név szerint ABC sorrendben

```
MariaDB [SemBLCL20]> SELECT Nev, TAJszam FROM Paciens ORDER BY Nev ASC;
+-----+-----+
| Nev          | TAJszam |
+-----+-----+
| Horváth Péter | 789123456 |
| Kovács János  | 123456789 |
| Nagy Éva     | 987654321 |
| Szabó Anna   | 321654987 |
| Tóth Gábor    | 456789123 |
+-----+-----+
5 rows in set (0.000 sec)
```

$\Pi_{\text{Nev, TAJszam}}(\delta_{\text{Nev}})(\text{Paciens})$

II. Kérdezze le az időpontokat dátum szerint rendezve.

```
MariaDB [SemBLCL20]> SELECT IdopontID, Datum, Idopont, Allapot FROM Idopont ORDER BY Datum;
+-----+-----+-----+-----+
| IdopontID | Datum      | Idopont | Allapot |
+-----+-----+-----+-----+
| ID004     | 2025-05-30 | 13:45   | Elfogadva |
| ID005     | 2025-06-19 | 15:30   | Fizetésre vár |
| ID001     | 2025-08-15 | 08:30   | Elfogadva |
| ID003     | 2025-09-01 | 10:00   | Befizetve |
| ID002     | 2025-10-16 | 09:15   | Lemondva |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

$\Pi_{\text{IdopontID, Datum, Idopont, Allapot}}(\delta_{\text{Datum}})(\text{Idopont})$

III. Kérdezze le az orvosok nevét, szakterületét szakterület szerint csoportosítva, ismétlődés nélkül.

```
MariaDB [SemBLCL20]> SELECT DISTINCT Nev, Szakterulet, COUNT(*) AS Orvosok_szama FROM Orvos GROUP BY Szakterulet;
+-----+-----+-----+
| Nev          | Szakterulet | Orvosok_szama |
+-----+-----+-----+
| Dr. Kovács Péter | Belgyógyászat | 1 |
| Dr. Kiss Anna   | Bőrgyógyászat | 1 |
| Dr. Szabó László | Fül-orr-gégészet | 1 |
| Dr. Nagy Zsuzsa | Nőgyógyászat | 1 |
| Dr. Tóth Gábor  | Sebészet | 1 |
+-----+-----+-----+
5 rows in set (0.000 sec)
```

$\Gamma_{\text{Nev, Szakterulet, COUNT(*)} \rightarrow \text{Orvosok_szama}}(\text{Orvos})$

IV. Kérdezze le és listázza a kezeléseket a legdrágábbal kezdve.

```
MariaDB [SemBLCL20]> SELECT KezelesID, Leiras, Koltseg FROM Kezeles ORDER BY Koltseg DESC;
```

KezelesID	Leiras	Koltseg
SEB003	Laparoszkópos epehólyag-eltávolítás	180000
FUL005	Endoszkópos orr- és arcüreg vizsgálat krónikus orrdugulás esetén	40000
BOR002	Anyajegy vizsgálat dermatoszkóppal és esetleges eltávolítás	35000
NOG004	Éves szűrővizsgálat méhnyakrák megelőzésére	25000
BEL001	Teljes vérkép és biokémiai vizsgálat cukorbetegség szűrésére	20000

5 rows in set (0.001 sec)

$\tau_{\text{Koltseg desc}}(\pi_{\text{KezelesID, Leiras, Koltseg}})(\text{Kezeles})$

V. Kérdezze le az adott páciens kezelési költségeit.

```
MariaDB [SemBLCL20]> SELECT p.Nev, SUM(k.Koltseg) AS Osszkoltseg FROM Paciens p JOIN Idopont i ON p.PaciensID = i.PaciensFK JOIN Kezeles k ON i.IdopontID = k.IdopontFK GROUP BY p.PaciensID, p.Nev ORDER BY Osszkoltseg DESC;
```

Nev	Osszkoltseg
Tóth Gábor	180000
Horváth Péter	40000
Nagy Éva	35000
Szabó Anna	25000
Kovács János	20000

5 rows in set (0.002 sec)

$\tau_{\text{Osszkoltseg desc}}(\gamma_{\text{Nev}}^{\sum(\text{Koltseg}) \rightarrow \text{Osszkoltseg}}(\text{Paciens} \blacktriangleright \blacktriangleleft \text{PaciensID=PaciensFK,}$

$\text{Idopont} \blacktriangleright \blacktriangleleft \text{IdopontID=IdopontFK}))(Kezeles)$

VI. Kérdezze le az orvosonkénti átlagos kezelési költséget és kezelések száma.

```
MariaDB [SemBLCL20]> SELECT o.Nev, o.Szakterulet, AVG(k.Koltseg) AS Atlag_koltseg, COUNT(*) AS Kezelesek_szama FROM Orvos o JOIN Idopont i ON o.OrvosID = i.OrvosFK JOIN Kezeles k ON i.IdopontID = k.IdopontFK GROUP BY o.OrvosID, o.Nev, o.Szakterulet ORDER BY Atlag_koltseg DESC;
```

Nev	Szakterulet	Atlag_koltseg	Kezelesek_szama
Dr. Tóth Gábor	Sebészet	180000.0000	1
Dr. Szabó László	Fül-orr-gégész	40000.0000	1
Dr. Kiss Anna	Bőrgyógyászat	35000.0000	1
Dr. Nagy Zsuzsa	Nőgyógyászat	25000.0000	1
Dr. Kovács Péter	Belgyógyászat	20000.0000	1

5 rows in set (0.003 sec)

$\tau_{\text{Atlag_koltseg desc}}(\gamma_{\text{Nev, Szakterulet}}^{\text{AVG(Koltseg)} \rightarrow \text{Atlag_koltseg, COUNT(*)} \rightarrow \text{Kezelesek_szama}}(\text{Orvos} \blacktriangleright \blacktriangleleft \text{OrvosID=OrvosFK,}$

$\text{Idopont} \blacktriangleright \blacktriangleleft \text{IdopontID=IdopontFK}))(Kezeles)$

VII. Kérdezze le a recepteket gyógyszerenként.

```
MariaDB [SemBLCL20]> SELECT rg.Gyogyszer, COUNT(*) AS Receptek_szama, p.Nev AS Gyakori_paciens, o.Nev AS Gyakori_orvos FROM Recept_gyogyszer rg JOIN Recept r ON rg.ReceptFK = r.ReceptID JOIN Paciens p ON r.PaciensFK = p.PaciensID JOIN Orvos o ON r.OrvosFK = o.OrvosID GROUP BY rg.Gyogyszer ORDER BY Receptek_szama DESC;
```

Gyogyszer	Receptek_szama	Gyakori_paciens	Gyakori_orvos
METFORMIN	1	Szabó Anna	Dr. Nagy Zsuzsa
IBUPROFEN	1	Kovács János	Dr. Kovács Péter
GLIBENKL	1	Szabó Anna	Dr. Nagy Zsuzsa
PARACET	1	Kovács János	Dr. Kovács Péter
CIPROFLOX	1	Horváth Péter	Dr. Szabó László
AMLODIPIN	1	Nagy Éva	Dr. Kiss Anna
ATROVENT	1	Tóth Gábor	Dr. Tóth Gábor
SALBUTAM	1	Tóth Gábor	Dr. Tóth Gábor
PREDNISON	1	Tóth Gábor	Dr. Tóth Gábor

9 rows in set (0.000 sec)

$\gamma_{\text{Gyogyszer}} \text{COUNT}(\ast) \rightarrow \text{Receptek_szama} (\pi_{\text{Gyogyszer, Nev (Paciens), Nev (Orvos)}}$

$(\text{Recept_gyogyszer} \blacktriangleright \blacktriangleleft \text{ReceptFK=ReceptID, Recept} \blacktriangleright \blacktriangleleft \text{PaciensFK=PaciensID}$
 $\text{Paciens} \blacktriangleright \blacktriangleleft \text{OrvosFK=OrvosID}))(Orvos)$

VIII. Kérdezze le a páciensek életkorát, számát életkor szerinti csoportosítással és a kezelési statisztikáit.

```
MariaDB [SemBLCL20]> SELECT FLOOR(DATEDIFF(CURDATE(), p.SzulDatum)/365) AS Eletkor, COUNT(DISTINCT p.PaciensID) AS Paciens_szama, AVG(k.Koltseg) AS Atlag_koltseg, COUNT(*) AS Kezelesek_szama FROM Paciens p JOIN Idopont i ON p.PaciensID = i.PaciensFK JOIN Kezeles k ON i.IdopontID = k.IdopontFK GROUP BY Eletkor ORDER BY Eletkor;
```

Eletkor	Paciensek_szama	Atlag_koltseg	Kezelesek_szama
30	1	25000.0000	1
34	1	35000.0000	1
40	1	20000.0000	1
42	1	40000.0000	1
46	1	180000.0000	1

5 rows in set (0.001 sec)

$\gamma_{\text{Eletkor}} \text{COUNT}(\text{DISTINCT PaciensID}) \rightarrow \text{Paciensek_szama}, \text{AVG}(\text{Koltseg}) \rightarrow \text{Atlag_koltseg}, \text{COUNT}(\ast) \rightarrow \text{Kezelesek_szama}$

$(\pi_{\text{Eletkor, PaciensID, Koltseg}}((\text{Paciens} \blacktriangleright \blacktriangleleft \text{PaciensID=PaciensFK, Idopont}) \blacktriangleright \blacktriangleleft \text{IdopontID=IdopontFK}))(Kezeles)$

IX. Kérdezze le a kezelések számát, bevételét havi bontásban, időrendben.

```
MariaDB [SemBLCL20]> SELECT DATE_FORMAT(i.Datum, '%Y-%m') AS Honap, COUNT(*) AS Kezelesek_szama, SUM(k.Koltseg) AS Osszbevetel, AVG(k.Koltseg) AS Atlag_koltseg FROM Idopont i JOIN Kezeles k ON i.IdopontID = k.IdopontFK WHERE i.Datum >= DATE_SUB(CURDATE(), INTERVAL 12 MONTH) GROUP BY Honap ORDER BY Honap;
```

Honap	Kezelesek_szama	Osszbevetel	Atlag_koltseg
2025-05	1	25000	25000.0000
2025-06	1	40000	40000.0000
2025-08	1	20000	20000.0000
2025-09	1	180000	180000.0000
2025-10	1	35000	35000.0000

5 rows in set (0.000 sec)

$\gamma_{\text{honap}} \text{COUNT}(\ast) \rightarrow \text{Kezelesek_szama}, \text{SUM}(\text{Koltseg}) \rightarrow \text{Osszbevetel}, \text{AVG}(\text{Koltseg}) \rightarrow \text{Atlag_koltseg}$

$(\pi_{\text{Honap, Koltseg, Datum}}(\sigma_{\text{Datum} \geq \text{DATE_SUB}(\text{CURDATE}(), \text{INTERVAL } 12 \text{ MONTH})}(\text{Idopont} \blacktriangleright \blacktriangleleft \text{IdopontID=IdopontFK}))(Kezeles)$

X. Kérdezze le az 5 leggyakrabban felírt gyógyszereket, csoportosítsa szakterületek szerint.

```

MariaDB [SemBLCL20]> SELECT rg.Gyogyszer, COUNT(*) AS Elofordulas, GROUP_CONCAT(DISTINCT o.Szakterulet SEPARATOR ', ') AS Szakteruletek FROM Recept_gyogyszer rg JOIN Recept r ON rg.ReceptFK = r.ReceptID JOIN Orvos o ON r.OrvosFK = o.OrvosID GROUP BY rg.Gyogyszer ORDER BY Elofordulas DESC LIMIT 5;
+-----+-----+-----+
| Gyogyszer | Elofordulas | Szakteruletek |
+-----+-----+-----+
| AMLODIPIN | 1 | Bőrgyógyászat |
| ATROVENT | 1 | Sebészet |
| CIPROFLOX | 1 | Fül-orr-gégészet |
| GLIBENKL | 1 | Nőgyógyászat |
| IBUPROFEN | 1 | Belgyógyászat |
+-----+-----+-----+
5 rows in set (0.011 sec)

```

$\gamma_{\text{Gyogyszer}}^{\text{COUNT(*)} \rightarrow \text{Elofordulas}, \text{GROUP_CONCAT(DISTINCT Szakterulet)} \rightarrow \text{Szakteruletek}}$

$(\pi_{\text{Gyogyszer}, \text{Szakterulet}}(\text{Recept_gyogyszer} \bowtie \text{ReceptFK=ReceptID, Recept} \bowtie \text{OrvosFK=OrvosID}))(\text{Orvos})$