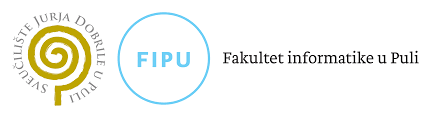
Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli



**DOKUMENTACIJA UZ PROJEKTNI ZADATAK**

**''BOLNICA''**

**TIM 1**

Neven Davidović

Marija Ilić

Darjan Juranović

Noel Modrušan

Tin Pritišanac

Smjer : Informatika

Kolegij : Baze podataka I

Mentor : doc. Dr. sc. Goran Oreški

Pula, svibanj, 2022. godina

Sadržaj

[1. UVOD 3](#_Toc104763146)

[2. OPIS POSLOVNOG PROCESA 3](#_Toc104763147)

[3. ENTITY RELATIONSHIP (ER) DIJAGRAM 4](#_Toc104763148)

[3.1. OPIS ER DIJAGRAMA 5](#_Toc104763149)

[4. RELACIJSKI MODEL (SHEME) 6](#_Toc104763150)

[5. EER DIJAGRAM (MYSQL) 7](#_Toc104763151)

[6. TABLICE 8](#_Toc104763152)

[6.1. TABLICA odjel 8](#_Toc104763153)

[6.2. TABLICA doktor 8](#_Toc104763154)

[6.3. TABLICA sos\_kontakt 9](#_Toc104763155)

[6.4. TABLICA pacijent 9](#_Toc104763156)

[6.5. TABLICA medicinske\_sestre 9](#_Toc104763157)

[6.6. TABLICA dijagnoza 9](#_Toc104763158)

[6.7. TABLICA soba 9](#_Toc104763159)

[6.8. TABLICA lijek 10](#_Toc104763160)

[6.9. TABLICA stanje\_lijekova 10](#_Toc104763161)

[6.10. TABLICA terapija 11](#_Toc104763162)

[6.11. TABLICA prijem 11](#_Toc104763163)

[6.12. TABLICA posjeta 11](#_Toc104763164)

[6.13. TABLICA oprema 12](#_Toc104763165)

[6.14. TABLICA stanje\_opreme 12](#_Toc104763166)

[7. UPITI 13](#_Toc104763167)

[7.1. UPIT 1 13](#_Toc104763168)

[7.2. UPIT 2 15](#_Toc104763169)

[7.3. UPIT 3 16](#_Toc104763170)

[7.4. UPIT 4 17](#_Toc104763171)

[7.5. UPIT 5 18](#_Toc104763172)

[7.6. UPIT 6 20](#_Toc104763173)

[7.7. UPIT 7 21](#_Toc104763174)

[7.8. UPIT 8 23](#_Toc104763175)

[7.9. UPIT 9 24](#_Toc104763176)

[7.10. UPIT 10 26](#_Toc104763177)

[7.11. UPIT 11 28](#_Toc104763178)

[7.12. UPIT 12 29](#_Toc104763179)

[7.13. UPIT 13 30](#_Toc104763180)

[7.14. UPIT 14 32](#_Toc104763181)

# UVOD

Ovaj projekt se od svoje početne verzije dosta promijenio, naravno na bolje. Kako su nam se znanje i vještine nadograđivale tokom semestra uvijek je postojala preinaka koja je poboljšala samu strukturu i funkciju pojedinih elemenata na izvedbu i rad same baze podataka. Sam proces izrade baze podataka je obavljen preko GitHuba zbog nemogućnosti okupljanja svih članova, što zbog posla što zbog drugih obaveza. Izgradili smo bazu podataka imena ''Bolnica'' te smo nastojali što detaljnije opisati sve poslovne procese i konceptualni model koji je prikazan preko ER dijagrama. Naravno, naša baza je samo djelić one prave, koju smo morali reducirati i izričito specificirati opis poslovanja za potrebe našeg projekta. Krajnji cilj je opisati što se događa kada pacijent dođe u bolnicu, od početnih koraka, prijema i pregleda, do terapije i smještaja u sobu. Tu su također dodatni skupovi entiteta koji su detaljno opisani u konceptualnom modelu. Također, svaka tablica, unosi podataka, upiti, ograničenja i ostalo su detaljno opisani u narednih nekoliko stranica. EER dijagram, odnosno logička shema baze podataka je generirana u MySQL Workbench-u preko Reverse Enginering opcije. Sva imena, prezimena, adrese, OIB-ovi, brojevi mobitel su generirani nasumično pomoću online alata, dok su nazivi opreme i odjela preneseni iz stvarne bolnice.

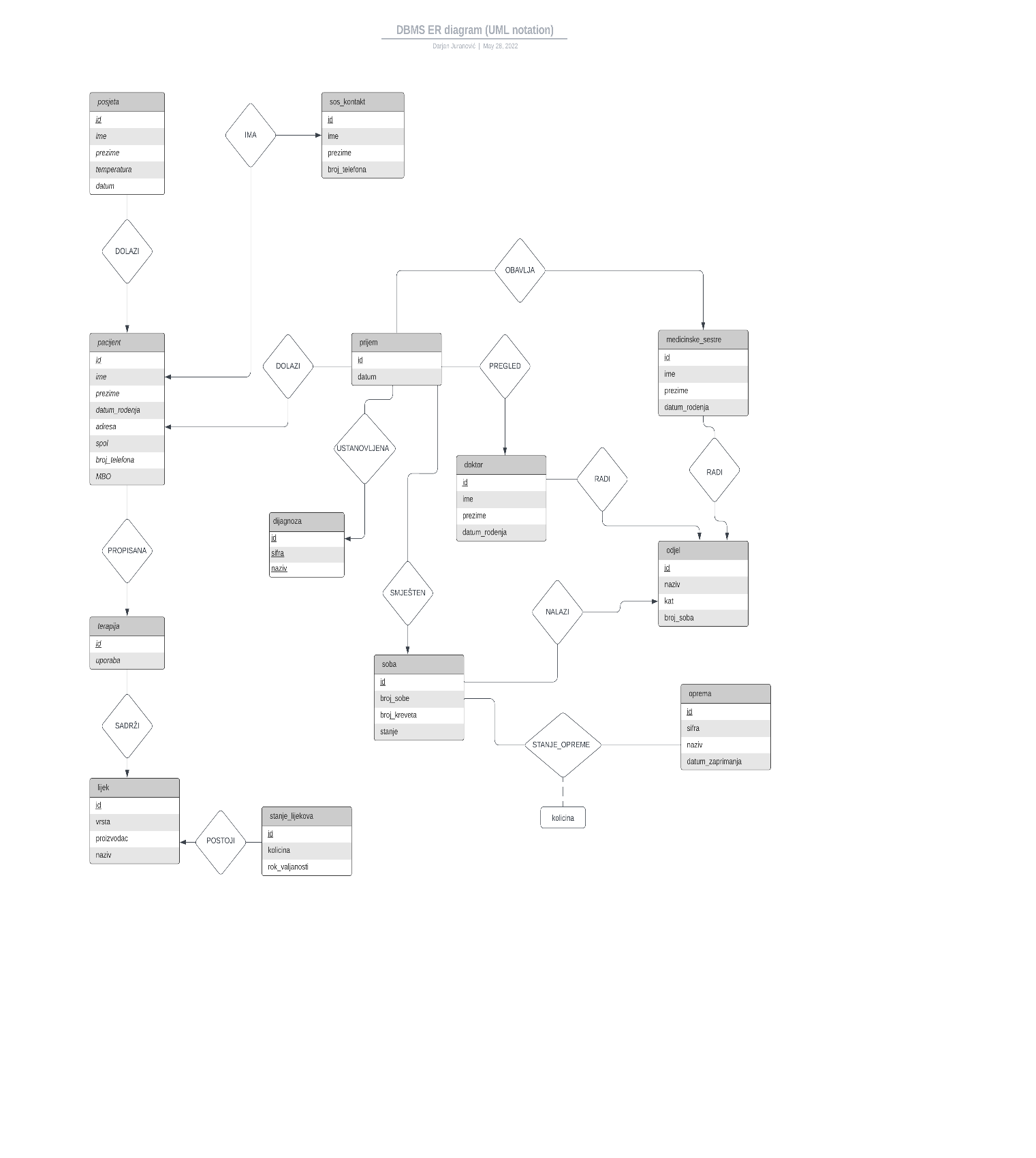
# OPIS POSLOVNOG PROCESA

U bolnici se prati prijem pacijenta. Medicinska sestra obavlja prijem pacijenta. Za svakog pacijenta se prati ime, prezime, datum\_rodenja, adresa, spol, broj\_telefona, MBO. Za svaki prijem pacijenta pratimo datum\_prijema.

Za pacijenta postoji SOS kontakt. Za svakog sos kontakta pratimo ime, prezime, broj\_telefona. Doktor obavlja pregled pacijenta, dok se za svakog doktora prati ime, prezime, datum\_rodenja. Nakon pregleda uspostavljena je dijagnoza za koju se prati naziv i sifra. Nakon toga je terapija propisana pacijentu, a za terapiju se prati uporaba. Za propisanu terapiju se izdaje lijek, te se za taj lijek prati vrsta, proizvodac i naziv. Kod stanja lijekova pratimo kolicinu i rok\_valjanosti. Pacijent je smješten u sobu na daljnje liječenje te se za sobu prati broj\_sobe, broj\_kreveta, i stanje. Posjeta dolazi pacijentu te se za nju prati ime, prezime, temperatura, datum, vrijeme\_dolaska, vrijeme\_odlaska. Sobe u kojima su smješteni pacijenti se nalaze po odjelima. Različita oprema je raspoređena po sobama, a za samu opremu se prati sifra, naziv, datum\_zaprimanja. Također postoji stanje opreme koje je povezano sa sobom i opremom te se za to stanje prati količina. Doktori i medicinske sestre rade na različitim odjelima u bolnici. Za medicinsku sestru se prati ime, prezime, datum\_rodenja.

# ENTITY RELATIONSHIP (ER) DIJAGRAM

Sljedeći ER dijagram detaljno i pregledno opisuje sve skupove - entiteta (njihove atribute) kao i skupove - veza između njih. Kardinalnost mapiranja (strelice) predstavljaju koliko drugih entiteta može biti povezano s entitetom preko određenog skupa – veza.



Slika 1/ ER dijagram za našu bazu

## OPIS ER DIJAGRAMA

* **DOKTOR** radi na **ODJELU** (doktor radi na jednom odjelu, dok na tom odjelu može raditi više doktora) – kardinalnost **one** to **many**
* **PACIJENT** ima **SOS\_KONTAKT** (pacijent ima jedan sos\_kontakt, dok je taj sos\_kontakt vezan samo za tog pacijenta) **one** to **one.** (ovo je navedeno ovako radi jednostavnijeg daljnjeg rada)
* **MEDICINSKA\_SESTRA** radi na **ODJELU** (medicinska sestra radi na jednom odijelu, dok na tom odjelu može raditi više medicinskih sestara) **one** to **many**
* **SOBA** se nalazi u **ODJELU** (soba je u jednom odjelu, dok odjel može imati više različitih soba) **one** to **many**
* **TERAPIJA** je propisana **PACIJENTU** (pacijent prima jednu terapiju, dok se ta terapija može prepisati većem broju pacijenata) **one** to **many.** (pošto je pacijentu propisana jedna terapija, dok god je ona na snazi lijekovi se neće mijenjati za tu terapiju i zbog toga je kardinalnost mapiranja **one** to **many**)
* **LIJEK** se izdaje na propisanu **TERAPIJU** (lijek se izdaje za određenu terapiju, dok više terapija može imati isti lijek) **one** to **many**
* Za **LIJEK** postoji **STANJE\_LIJEKOVA** (za jedan lijek postoji više stanja, pošto se ovisno o terapiji mijenja količina lijekova) **one** to **many**
* **PACIJENT** primljen u **BOLNICU** (pacijent je primljen u bolnicu, dok u tu istu bolnicu može biti primljeno više pacijenata) **one** to **many**
* **MEDICINSKA\_SESTRA** obavlja **PRIJEM** (prijem je obavljen od strane medicinske sestre, dok ta ista sestra može obaviti više prijema) **one** to **many**
* **DOKTOR** nakon prijema pregledava **PACIJENTA** (pacijent je pregledan od jednog doktora, dok isti taj doktor može pregledati više pacijenata) **one** to **many**
* Doktor je uspostavio **DIJAGNOZU** nakon **PRIJEMA** (dijagnoza je uspostavljena od doktora, dok isti taj doktor može uspostaviti više dijagnoza) **one** to **many**
* Pacijent je smješten u **SOBU** nakon **PRIJEMA** (pacijent je smješten u sobu, dok u toj sobi može biti više pacijenata) **one** to **many**
* **POSJETA** dolazi **PACIJENTU** (jedan posjetitelj posjećuje pacijenta, dok taj pacijent može imati više posjetitelja) **one** to **many**
* **OPREMA** je raspoređena po **SOBI** (više soba može imati više različite opreme) **many** to **many**
* Dodatni entitet **STAVKA\_OPREME** gdje se primarni ključevi **SOBE** i **OPREME** referenciraju kao strani ključ za STAVKU\_OPREME

# RELACIJSKI MODEL (SHEME)

**odjel** (id, naziv, kat, broj\_soba)

**doktor** (id, ime, prezime, datum\_rodenja, id\_odjel)

**sos\_kontakt** (id, ime, prezime, broj\_telefona)

**pacijent** (id, ime, prezime, datum\_rodenja, adresa, spol, broj\_telefona, MBO, id\_sos\_kontakt)

**medicinske\_sestre** (id, ime, prezime, datum\_rodenja, id\_odjel)

**dijagnoza** (id, naziv, sifra)

**soba** (id, broj\_sobe, broj\_kreveta, stanje, id\_odjel)

**lijek** (id, vrsta, proizvodac, naziv)

**stanje\_lijekova** (id, id\_lijek, kolicina, rok\_valjanosti)

**terapija** (id, id\_pacijent, id\_lijek, uporaba)

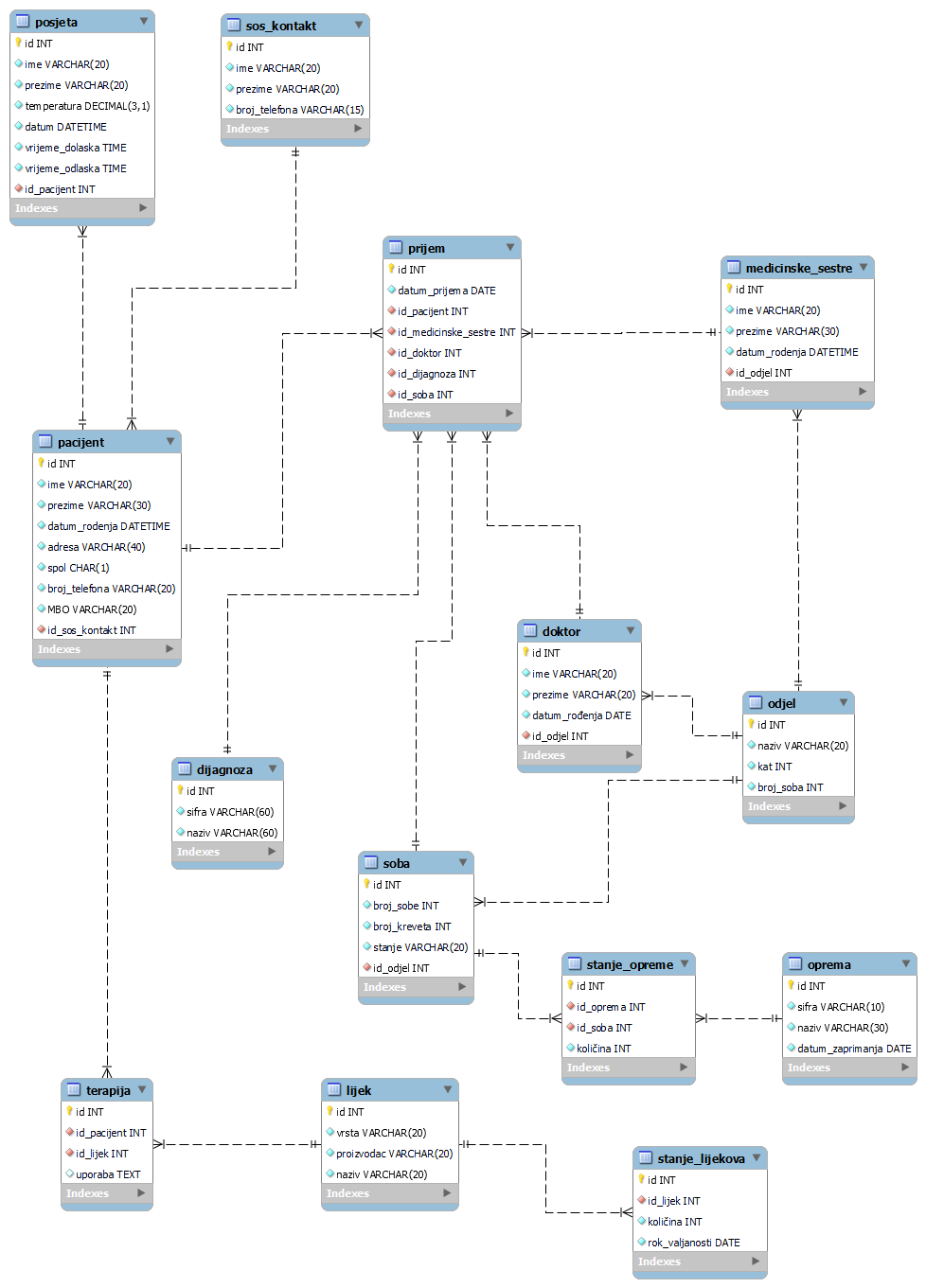
**prijem** (id, datum\_prijema, id\_pacijent, id\_medicinske\_sestre, id\_doktor, id\_dijagnoza, id\_soba)

**posjeta** (id, ime, prezime, temperatura, datum, vrijeme\_dolaska, vrijeme\_odlaska, id\_pacijent)

**oprema** (id, sifra, naziv, datum\_zaprimanja)

**stanje\_opreme** (id, id\_oprema, id\_soba, kolicina)

# EER DIJAGRAM (MYSQL)



Slika 2/EER dijagram, napravljen u MySQL Workbenchu

# TABLICE

## TABLICA odjel

Tablica **odjel** služi evidenciji postojećih odjela u bolnici. Sadrži atribute: **id**, **naziv**, **kat** i **broj\_soba**. Atribut **id** je PRIMARY KEY tablice tipa INTEGER jer se u njega unosi brojčana vrijednost. Služi kako bi smo mogli jedinstveno označiti svaki odjel u bolnici. Iako atribut **id** mora biti jedinstven kako bi opravdao svoju svrhu, nije potrebno staviti ograničenje UNIQUE jer je PRIMARY KEY jedinstven sam po sebi. Atribut **naziv** je tipa VARCHAR koji nam omogućuje dinamičku alokaciju memorije. Odabrala sam tip VARCHAR s obzirom da ne znam unaprijed koliko će znakova pojedini naziv odjela sadržavati. Maksimalna duljina naziva definirana je u zagradi odmah nakon tipa podatka, a iznosi 20 znakova. Unutar atributa **naziv** postavljeno je ograničenje NOT NULL koje ne dopušta unos podataka u bazu podataka bez ovog podatka. Također je postavljeno ograničenje UNIQUE koje osigurava da je svaki naziv odjela u tablici jedinstven. Atribut **kat** je tipa INTEGER, a označava na kojem se katu pojedini odjel u bolnici nalazi. Unutar **kat** atributa također je postavljeno ograničenje NOT NULL kako bi osigurali unos ovog podatka prilikom unošenja podataka unutar baze podataka. Posljednji atribut **broj\_soba** označava koliko se soba za smještanje pacijenata nalazi unutar pojedinog odjela, a tipa je INTEGER. Također je obavezan pri unošenju podataka u bazu podataka pa sadrži ograničenje NOT NULL.

CREATE TABLE odjel (

id INTEGER PRIMARY KEY,

naziv VARCHAR (20) NOT NULL UNIQUE,

kat INTEGER NOT NULL,

broj\_soba INTEGER NOT NULL

);

## TABLICA doktor

Tablica **doktor** služi evidenciji zaposlenih doktora u bolnici. Sadrži atribute: **id**, **ime**, **prezime**, **datum\_rodenja** i **id\_odjel**. Atribut **id** je tipa INTEGER jer u njega unosimo brojčanu vrijednost koja jedinstveno definira svakog pojedinog doktora, što ga ujedno čini i odličnim PRIMARY KEY atributom. Atributi **ime** i **prezime** označavaju ime i prezime svakog pojedinog doktora. Tip podatka VARCHAR koji sam odabrala za ova dva atributa služi dinamičkoj alokaciji memorije, a maksimalna duljina unosa iznosi 20 znakova i definirana je u zagradi odmah nakon tipa podatka. Ograničenje NOT NULL osigurava obavezan unos ova dva podatka prilikom unošenja podataka unutar baze podataka. Atribut **datum\_rodenja** je tipa DATE, a označava datum rođenja svakog pojedinog doktora. Ovaj tip podatka omogućuje nam lakši daljnji rad sa datumima prilikom pisanja upita. Također posjeduje ograničenje NOT NULL koje nam garantira obavezan unos ovog podatka u bazu podataka. Atribut **id\_odjel** je tipa INTEGER, a označava odjel na kojem je pojedini doktor zaposlen u bolnici. Atribut **id\_odjel** je FOREIGN KEY unutar tablice **doktor**, a povezuje tablicu **doktor** sa tablicom **odjel** preko atributa **id** nakon REFERENCES dijela unutar definicije stranog ključa.

CREATE TABLE doktor (

id INTEGER PRIMARY KEY,

ime VARCHAR (20) NOT NULL,

prezime VARCHAR (20) NOT NULL,

datum\_rodenja DATE NOT NULL,

id\_odjel INTEGER NOT NULL,

FOREIGN KEY (id\_odjel) REFERENCES odjel (id)

);

## TABLICA sos\_kontakt

Tablica **sos\_kontakt** služi evidenciji SOS kontakata pacijenata. Sadrži atribute **id**, **ime**, **prezime** i **broj\_telefona**. Atribut **id** je tipa podatka INTEGER, a ujedno je i PRIMARY KEY jer označava specifičnu brojčanu vrijednost svakog pojedinog SOS kontakta. Atributi **ime** i **prezime** su tipa podatka VARCHAR. Za ova dva atributa odabrala sam tip podatka koji koristi dinamičku alokaciju memorije s obzirom da ne znam unaprijed koja je duljina podataka koji će se unositi unutar njih. Maksimalna duljina unosa unutar atributa **ime** i **prezime** definirana je u zagradi odmah nakon tipa podatka. Ograničenje NOT NULL unutar ova dva atributa osigurava nam obavezan unos ovih podataka prilikom unosa podataka u bazu podataka. Atribut **broj\_telefona** tipa je VARCHAR, a označava broj telefona svakog pojedinog SOS kontakta. Odabrala sam VARCHAR tip podatka za ovaj atribut kako bi omogućila dinamičku alokaciju memorije, s obzirom da nisu svi brojevi telefona jednake duljine. Maksimalnu diljinu sam definirala u zagrodi nakon tipa, a iznosi 15 znakova. Drugi razlog iz kojeg sam odabrala VARCHAR tip podatka za ovaj atribut je taj što sam na ovaj način omogućila unos nule na prvo mjesto. Ograničenja koja su stavljena na atribut **broj\_telefona** su NOT NULL ograničenjne koje nam osigurava obavezan unos ovog podatka prilikom unošenja podataka u bazu podataka i UNIQUE ograničenje s obzirom da ne postoje dva jednaka broja telefona.

CREATE TABLE sos\_kontakt (

id INTEGER PRIMARY KEY,

ime VARCHAR (20) NOT NULL,

prezime VARCHAR (20) NOT NULL,

broj\_telefona VARCHAR (15) NOT NULL UNIQUE

);

## TABLICA pacijent

## TABLICA medicinske\_sestre

## TABLICA dijagnoza

## TABLICA soba

Tablica soba sadrži: integer id koji je primarni ključ u rasponu od 700-800, broj\_sobe i broj\_kreveta koji su integeri i ne smiju biti NULL, stanje koje je tipa varchar s ograničenjem do 20 znakova i opisuje trenutno stanje sobe (je li slobodna, popunjena ili se ne koristi itd.), te id\_odjel koji je integer i strani ključ koji povezuje sobu sa relacijeom odjel.

CREATE TABLE soba (

id INTEGER PRIMARY KEY,

broj\_sobe INTEGER NOT NULL,

broj\_kreveta INTEGER NOT NULL,

stanje VARCHAR (20) NOT NULL,

id\_odjel INTEGER NOT NULL,

FOREIGN KEY (id\_odjel) REFERENCES odjel (id)

);

## TABLICA lijek

Tablica lijek služi za pohranu informacija o svakom lijeku u bolnici.

Sadrži atribute: id tipa integer koji služi kao primarni ključ i u rasponu je od 800-900,

vrsta, proizvodac i naziv koji su tipa varchar s ograničenjem do 20 znakova i ne smiju imati NULL vrijednost.

CREATE TABLE lijek (

id INTEGER PRIMARY KEY,

vrsta VARCHAR (20) NOT NULL,

proizvodac VARCHAR (20) NOT NULL,

naziv VARCHAR (20) NOT NULL

);

## TABLICA stanje\_lijekova

Ova tablica služi kao evidencija stanja svih lijekova, pri čemu jedan lijek može biti upisan više puta, no sa različitim količinama i rokom valjanosti. Sadrži atribute: id tipa integer koji služi kao primarni ključ i u rasponu je od 900-1000, id\_lijek tipa integer koji je strani ključ koji povezuje ovu tablicu s tablicom lijek, količina tipa integer koji ne smije biti NULL, rok\_valjanosti tipa date koji ne smije biti NULL. Na kraju imamo ograničenje gdje rok valjanosti mora biti nakon datuma 15.05.2020.

CREATE TABLE stanje\_lijekova (

id INTEGER PRIMARY KEY,

id\_lijek INTEGER NOT NULL,

količina INTEGER NOT NULL,

rok\_valjanosti DATE NOT NULL,

FOREIGN KEY (id\_lijek) REFERENCES lijek (id),

CHECK (rok\_valjanosti > STR\_TO\_DATE('15.05.2020.', '%d.%m.%Y.'))

);

## TABLICA terapija

Tablica terapija povezuje pacijente sa lijekovima koji su im prepisani, uz kratak opis njihove upotrebe. Sadrži atribute: id tipa integer koji služi kao primarni ključ i u rasponu je od 1000-1100, id\_pacijent tipa integer koji je strani ključ koji povezuje ovu tablicu sa tablicom pacijent i ne smije biti NULL, id\_lijek tipa integer koji je strani ključ koji povezuje ovu tablicu sa tablicom lijek i ne smije biti NULL i uporaba tipa text pri čemu je omogućen unos dužeg teksta kao opisa (do 65535 znakova).

CREATE TABLE terapija (

id INTEGER PRIMARY KEY,

id\_pacijent INTEGER NOT NULL,

id\_lijek INTEGER NOT NULL,

uporaba TEXT,

FOREIGN KEY (id\_pacijent) REFERENCES pacijent (id),

FOREIGN KEY (id\_lijek) REFERENCES lijek (id)

);

## TABLICA prijem

Tablica prijem sastoji se od atributa id koji je tipa INTEGER te služi kao primarni ključ,datum\_prijema koje je tipa DATE. Ostali atributi su strani ključevi koji su tipa INTEGER. Strani ključ id\_medicinske\_sestre povezuje tablicu medicinske\_sestre, strani ključ id\_pacijent povezuje prijem sa tablicom pacijent, strani ključ id\_doktor povezuje tablicu prijem sa tablicom doctor, id\_dijagnoza sa tablicom dijagnoza i id\_soba sa tablicom soba.

CREATE TABLE prijem(

id INTEGER PRIMARY KEY,

datum\_prijema DATE NOT NULL,

id\_pacijent INTEGER NOT NULL,

id\_medicinske\_sestre INTEGER NOT NULL,

id\_doktor INTEGER NOT NULL,

id\_dijagnoza INTEGER NOT NULL,

id\_soba INTEGER NOT NULL,

FOREIGN KEY (id\_pacijent) REFERENCES pacijent(id),

FOREIGN KEY (id\_medicinske\_sestre) REFERENCES medicinske\_sestre(id),

FOREIGN KEY (id\_doktor) REFERENCES doktor (id),

FOREIGN KEY (id\_dijagnoza) REFERENCES dijagnoza(id),

FOREIGN KEY (id\_soba) REFERENCES soba(id)

);

## TABLICA posjeta

Tablica posjeta sadrži primarni ključ tipa INTEGER, atribute ime i prezime tipa VARCHAR s ograničenjem do 20 znakova, atribut temperature koji je tip NUMERIC gdje je odabrano da iza decimalnog zareza bude ukupno jedan broj. Imamo atribute vrijeme\_dolaska i vrijeme\_odlaska. Imamo i strani ključ id\_pacijent koji veže tablicu posjeta sa tablicom pacijent.

CREATE TABLE posjeta(

id INTEGER PRIMARY KEY,

ime VARCHAR(20) NOT NULL,

prezime VARCHAR(20) NOT NULL,

temperatura NUMERIC(3,1) NOT NULL,

datum DATETIME NOT NULL,

vrijeme\_dolaska TIME NOT NULL,

vrijeme\_odlaska TIME NOT NULL,

id\_pacijent INTEGER NOT NULL,

FOREIGN KEY (id\_pacijent) REFERENCES pacijent (id)

);

## TABLICA oprema

Tablica oprema ima primarni ključ tipa INTEGER te atribute sifra koji je tipa VARCHAR s ograničenjem od 10, naziv koji je isto VARCHAR s ograničenjem 30. Atribut datum\_zaprimanja je tipa DATE.

CREATE TABLE oprema(

id INTEGER PRIMARY KEY,

sifra VARCHAR(10) NOT NULL UNIQUE,

naziv VARCHAR(30) NOT NULL,

datum\_zaprimanja DATE NOT NUL

);

## TABLICA stanje\_opreme

Unutar tablice oprema imamo atribut id kao primarni ključ te strane ključeve id\_soba i id\_oprema. Atribut kolicina govori o kolicini opreme. Svi atributi, primarni i strani ključevi su tipa INTEGER.

CREATE TABLE stanje\_opreme(

id INTEGER PRIMARY KEY,

id\_oprema INTEGER NOT NULL,

id\_soba INTEGER NOT NULL,

kolicina INTEGER NOT NULL,

FOREIGN KEY (id\_oprema) REFERENCES oprema(id),

FOREIGN KEY (id\_soba) REFERENCES soba(id)

);

# UPITI

## UPIT 1

Količina opreme koju koristi pojedina medicinska sestra.

Medicinska sestra Ivana Ivić odlučila se požaliti ravnatelju bolnice kako joj fali opreme koju mora koristiti za kvalitetno obavljanje posla. Kako bi riješio problem medicinske sestre Ivane, ali i možebitne probleme ostalih medicinskih sestara u bolnici, ravnatelj bolnice je za početak zatražio popis svih medicinskih sestara i količine opreme koja im je dostupna za rad.

TRAŽENO RJEŠENJE:  
**medicinske\_sestre** id, **medicinske\_sestre** ime\_i\_prezime, kolicina\_opreme

KOD ZA UPIT:

CREATE VIEW sestra\_soba AS

SELECT medicinske\_sestre.id, CONCAT(medicinske\_sestre.ime, ' ', medicinske\_sestre.prezime) AS ime\_i\_prezime, soba.id AS id\_sobe

    FROM medicinske\_sestre, soba

    WHERE medicinske\_sestre.id\_odjel=soba.id\_odjel;

CREATE VIEW kol\_po\_sobi AS

SELECT sestra\_soba.id, sestra\_soba.ime\_i\_prezime, COALESCE(kolicina.kol\_u\_sobi, 0) AS kolicina

    FROM sestra\_soba

LEFT JOIN

    (SELECT \*, SUM(kolicina) AS kol\_u\_sobi

        FROM stanje\_opreme

        GROUP BY id\_soba) AS kolicina ON sestra\_soba.id\_sobe=kolicina.id\_soba;

SELECT kol\_po\_sobi.id, kol\_po\_sobi.ime\_i\_prezime, SUM(kolicina) AS kolicina\_opreme

    FROM kol\_po\_sobi

    GROUP BY id;

OPIS UPITA:

S obzirom da je oprema raspoređena po sobama moramo medicinske sestre povezati sa sobama u kojima obavljaju rad kako bi mogli prebrojati opremu koja je dostupna pojedinoj medicinskoj sestri.

* Unutar prvog upita povezala sam tablicu **medicinske\_sestre** i tablicu **soba** pomoću kartezijevog produkta. Kartezijev produkt sam dobila na način da u FROM dijelu navedem obje tablice.
* U WHERE dio upita dodala sam uvjet povezivanja po jednakom **id\_odjel** atributu kako bi ispravna imena medicinskih sestra povezala sa ispravnim brojevima soba u kojima obavljaju rad.
* U SELECT dio navela sam atribute koji će mi koristiti za daljnje rješavanje, a to su **id** atribut iz tablice **medicinske\_sestre**, **ime\_i\_prezime** te **id\_sobe** iz tablice **soba**.
* Atribut **ime\_i\_prezime** sam spojila iz dva atributa naredbom CONCAT.
* Od prvog upita kreirala sam pogled pod nazivom **sestra\_soba** koristeći naredbu CREATE VIEW.
* Nakon što sam kreirala pogled koji nam govori u kojoj sobi radi koja medicinska sestra, pronašla sam količinu opreme koja se nalazi u svakoj od soba na način da sam tablicu **stanje\_opreme** grupirala prema atributu **id\_soba** naredbom GROUP BY.
* Unutar SELECT dijela drugog upita, osim cijele tablice **stanje\_opreme**, dodala sam i atribut **kol\_u\_sobi**.
* Atribut **kol\_u\_sobi** označava količinu opreme koja se nalazi u pojedinoj sobi, a s obzirom da sam već grupirala cijelu tablicu prema atributu **id\_soba**, lako sam izračunala i količinu koristeći naredbu SUM.
* Kako bi povezala medicinske sestre sa količinom opreme, drugi sam upit pomoću naredbe LEFT JOIN pridodala pogledu **sestra\_soba**. Pri povezivanju tablice preimenovala sam ju u **kolicina**, a povezivanje sam obavila prema uvjetu da su atributi **broj\_sobe** iz pogleda **sestra\_soba** i **id\_soba** iz tablice **kolicina** jednaki.
* U SELECT dio upita navela sam samo potrebne atribute, a to su: **id** i **ime\_i\_prezime** iz pogleda **sestra\_soba** i **kolicina**.
* Atribut **kolicina** rezultat je sumiranja količine opreme iz pridodane tablice **kolicina**, a naredbom COALESCE postigla sam upisivanje 0 gdje se nalazi NULL vrijednost unutar atributa.
* U konačnici sam pomoću naredbe CREATE VIEW kreirala pogled pod nazivom **kol\_po\_sobi** u kojem se nalaze sve medicinske sestre i količina opreme koju imaju na raspolaganju.
* Kako bi vidjeli točan popis medicinskih sestara i točnu količinu opreme koju imaju na raspolaganju ponaosob (kako se imena i prezimena ne bi ponavljala i ne bi se prepustilo nama da sami zbrajamo opremu po imenima), kreirala sam i treći upit.
* Unutar trećeg upita sam pogled **kol\_po\_sobi** grupirala prema atributu **id** koji se odnosi na **id** medicinskih sestara i u SELECT dijelu sam, osim atributa koji su potrebni za rješenje, konačno sumirala opremu pomoću funkcije SUM preko atributa **kolicina** i naredbom AS rezultat preimenovala u **kolicina\_opreme**.

REZULTAT:

|  |  |  |
| --- | --- | --- |
| id | ime\_i\_prezime | kolicina\_opreme |
| 500 | Ivana Ivić | 2 |
| 501 | Miliana Milić | 12 |
| 502 | Ivana Marić | 12 |
| 503 | Marko Marulić | 2 |
| 504 | Žarka Stanić | 12 |

## UPIT 2

Popis SOS kontakata svih pacijenata sa dijagnozom 'Infarctus myocardii acutus'.

Doktor Noris Grubor sa odjela Kardiologije zamolio je medicinsku sestru da obavijesti sve SOS kontakte pacijenata sa dijagnozom Infarctus myocardii acutus da im je stanje kritično zbog proživljenog stresa uzrokovanog potresom. Kako bi to učinila, medicinska sestra zatražila je popis svih SOS kontakata pacijenata s tom dijagnozom.

TRAŽENO RJEŠENJE:  
**pacijent** ime\_i\_prezime, **sos\_kontakt** sos\_ime\_i\_prezime , **sos\_kontakt** broj\_telefona

KOD ZA UPIT:

CREATE VIEW sa\_dijagnozom AS

SELECT prijem.id\_pacijent

    FROM prijem, dijagnoza

    WHERE prijem.id\_dijagnoza=dijagnoza.id AND dijagnoza.naziv='Infarctus myocardii acutus';

SELECT CONCAT(pacijent.ime, ' ', pacijent.prezime) AS ime\_i\_prezime,

CONCAT(sos\_pod.ime, ' ', sos\_pod.prezime) AS sos\_ime\_i\_prezime, sos\_pod.broj\_telefona

    FROM pacijent

RIGHT JOIN sa\_dijagnozom AS trazena\_dijag ON pacijent.id=trazena\_dijag.id\_pacijent

LEFT JOIN sos\_kontakt AS sos\_pod ON pacijent.id\_sos\_kontakt=sos\_pod.id;

OPIS UPITA:

Kako bi uspješno odgovorili na ovaj upit prvo moramo pronaći kojim je pacijentima dijagnosticiran 'Infarctus myocardii acutus', što ćemo učiniti preko tablice **prijem** unutar koje su nam pobrojane dijagnoze svakog pacijenta. Zatim povezujemo tablice **pacijent** (kako bi dobili potrebne podatke o pacijentu) i **sos\_kontakt** (kako bi dobili potrebne podatke o SOS kontaktu pacijenta).

* Unutar prvog upita u FROM dijelu napravila sam kartezijev produkt tablica **prijem** i **dijagnoza**.
* U WHERE dio upita navela sam dva uvjeta spajanja. Prvi uvjet spajanja odnosi se na atribut **id** dijagnoze kako bi unutar tablice **prijem** dobili ispravna imena postavljenih dijagnoza, a drugi uvjet spajanja odnosi se na konkretan **naziv** dijagnoze koju tražimo u rezultatu.
* U SELECT dijelu upita navela sam samo atribut **id\_pacijent** jer mi je on dovoljan za daljnji rad na upitu.
* Pomoću naredbe CREATE VIEW sam kreirala pogled imena **sa\_dijagnozom** od prvog upita.
* Kako bi na temelju **id\_pacijent** atributa iz pogleda **sa\_dijagnozom** dobila potrebne podatke vezane uz pacijente, pogled **sa\_dijagnozom** sam pod imenom **trazena\_dijag**, pomoću RIGHT JOIN naredbe povezala sa tablicom **pacijent**. Kao uvjet povezivanja navela sam jednakost atributa **id** iz tablice **pacijent** i atributa **id\_pacijent** iz tablice **trazena\_dijag**.
* Jedini podaci koji nam sad fale su podaci vezani uz SOS kontakt podjedinog pacijenta. Kako bi i taj dio upita uspješno riješila uključila sam tablicu **sos\_kontakt** pomoću naredbe LEFT JOIN pod nazivom **sos\_pod**, sa uvijetom da su atributi **id\_sos\_kontakt** iz tablice **pacijent** i **id** iz tablice **sos\_pod** jednaki.
* U SELECT dio upita navela sam atribute **ime\_i\_prezime** (odnosi se na ime i prezime pacijenta), **sos\_ime\_i\_prezime** (odnosi se na ime i prezime SOS kontakta), te **broj\_telefona** (odnosi se na broj telefona SOS kontakta).
* Atribute **ime\_i\_prezime** i **sos\_ime\_i\_prezime** sam spojila pomoću naredbe CONCAT od odvojenih atributa ime i prezime iz tablica **pacijent** i **sos\_pod**.

REZULTAT:

|  |  |  |
| --- | --- | --- |
| ime\_i\_prezime | sos\_ime\_i\_prezime | broj\_telefona |
| Sebastijan Milošević | Aden Kotolaš | 0997871285 |
| Lucas Perić | Vojmil Novaković | 0990409583 |

## UPIT 3

Doktor s najviše pacijenata.

Ravnatelj bolnice odlučio je dodijeliti nagradu doktoru koji je zadužen za najviše pacijenata.

TRAŽENO RJEŠENJE:  
**doktor** id, **doktor** ime\_i\_prezime, **doktor** id\_odjel, broj\_pacijenata

KOD ZA UPIT:

CREATE VIEW svi\_brojevi\_pac AS

SELECT doktor.id, CONCAT(doktor.ime, ' ', doktor.prezime) AS ime\_i\_prezime, doktor.id\_odjel,

COALESCE(broj\_pacijenata, 0) AS broj\_pacijenata

    FROM doktor

LEFT JOIN

    (SELECT id\_doktor, COUNT(id\_pacijent) AS broj\_pacijenata

        FROM prijem

        GROUP BY id\_doktor) AS izracun ON doktor.id=izracun.id\_doktor;

SELECT \*

    FROM svi\_brojevi\_pac

    ORDER BY broj\_pacijenata DESC

    LIMIT 1;

OPIS UPITA:

Za rješavanje ovoga upita ključna je tablica **prijem** unutar koje je evidentirana dodjela doktora pacijentu prilikom prijema pacijenta u bolnicu.

* Prvi korak koji sam poduzela u svrhu rješavanja ovoga upita je prebrojavanje pacijenata unutar tablice **prijem** za svakog doktora ponaosob.
* Naredbom GROUP BY grupirala sam prema atributu **id\_doktor** podatke iz tablice **prijem**.
* Kako bi mogla prebrojati pacijente upotrijebila sam naredbu COUNT na atributu **id\_pacijent** unutar SELECT dijela upita i naredbom AS imenovala dobiveni atrinut kao **broj\_pacijenata**.
* Također sam u SELECT dio upita navela atribut **id\_doktor** koji mi je potreban za povezivanje traženih podataka o doktoru.
* Kako bi dobila podatke o doktoru, naredbom LEFT JOIN dodala sam dobiveni rezultat upita unutar tablice **doktor** pod nazivom **izracun**. Kao uvijet spajanja iskoristila sam jednakost atributa **id** iz tablice **doktor** i atributa **id\_doktor** iz tablice **izracun**.
* U SELECT dio upita navela sam iz tablice **doktor** slijedeće atribute: **id**, **ime\_i\_prezime** i **id\_odjel**.
* Atribut **ime\_i\_prezime** dobila sam spajanjem atributa **ime** i atributa **prezime** iz tablice **doktor** koristeći naredbu CONCAT.
* Kreirala sam pogled s imenom **svi\_brojevi\_pac** pomoću naredbe CREATE VIEW.
* Na kraju sam kreirala upit u kojem sam u SELECT dijelu navela sve atribute pogleda **svi\_brojevi\_pac** koju sam navela u FROM dijelu upita.
* Naredbom ORDER BY sam sortirala podatke silazno prema atributu **broj\_pacijenata**.
* Naredbom LIMIT sa vrijednošću 1 sam u konačnici dobila doktora s najviše pacijenata.

RJEŠENJE:

|  |  |  |  |
| --- | --- | --- | --- |
| id | ime\_i\_prezime | id\_odjel | broj\_pacijenata |
| 200 | Krešimira Paspalj | 102 | 9 |

## UPIT 4

Popis svih lijekova kojima je datum isteka prije 2023. godine, a proizvođač im je Bayer.

Potrebno je napisati narudžbu lijekova za proizvođača Bayer.

TRAŽENO RJEŠENJE:  
**lijek** id, **lijek** vrsta, **lijek** proizvodac, **lijek** naziv

KOD ZA UPIT:

CREATE VIEW prije\_zadane AS

SELECT \*

    FROM stanje\_lijekova

    WHERE YEAR (rok\_valjanosti) < 2023

    ORDER BY rok\_valjanosti ASC;

SELECT lijek\_pod.id, lijek\_pod.vrsta, lijek\_pod.proizvodac, lijek\_pod.naziv

    FROM prije\_zadane

LEFT JOIN lijek AS lijek\_pod ON prije\_zadane.id\_lijek=lijek\_pod.id

HAVING proizvodac='Bayer';

OPIS UPITA:

Datum isteka roka svakog lijeka vidljiv je unutar tablice **stanje\_lijekova**, a ostali podaci o lijeku koji su nam potrebni da bi izvršili narudžbu proizvođaču nalaze se unutar tablice **lijek**.

* Unutar prvog upita pomoću naredbe YEAR izvršene na atributu **rok\_valjanosti** u WHERE dijelu navela sam kako godina isteka roka mora biti manja od 2023. godine.
* Tablica navedena u FROM dijelu upita, na kojoj sam izvršila ovaj uvijet je **stanje\_lijekova**, a pomoću ORDER BY naredbe datume sam poredala silazno.
* Pomoću naredbe CREATE VIEW kreirala sam pogled imena **prije\_zadane**.
* Kako bi dobila podatke vezane uz lijekove koji su potrebni da bi se izvršila narudžba, povezala sam tablicu **lijek** pomoću LEFT JOIN naredbe pod imenom **lijek\_pod** sa pogledom **prije\_zadane**. Uvijet koji sam postavila za povezivanje je jednakost **id\_lijek** atributa iz pogleda **prije\_zanadne** sa **id** atributom iz tablice **lijek\_pod**.
* Naredbom HAVING sam zadala uvijet da se prikazuju samo oni lijekovi koji imaju pod atributom **proizvodac** naziv traženog proizvođača.
* U SELECT dijelu navela sam samo atribute potrebne za izvršavanje narudžbe, a to su **id**, **vrsta**, **proizvodac** i **naziv** iz tablice **lijek\_pod**.

RJEŠENJE:

|  |  |  |  |
| --- | --- | --- | --- |
| id | vrsta | proizvodac | naziv |
| 800 | Anelgetik | Bayer | Aspirin |
| 812 | Antibiotik | Bayer | Meropenem |

## UPIT 5

Popis pacijenata sa brojem posjeta (odvojeno posjete SOS kontakata i onih koji nisu SOS kontakti).

Na otpusnom listu pacijenta bolnica svakom pacijentu daje podatak koliko je ukupno imao SOS kontakat posjeta i ostalih posjeta. Zbog potresa bolnica se seli u zamjensku zgradu dok oštećena zgrada ne bude ponovno u funkciji. Ravnatelj bolnice naredio je da se svakom pacijentu prije preseljenja izda dokument sličan otpusnom listu koji između ostalih sadrži i te podatke.

TRAŽENO RJEŠENJE:  
**pacijent** id, **pacijent** ime\_i\_prezime, **pacijent** spol, sos\_broj\_posjeta, broj\_ostalih\_posjeta

KOD ZA UPIT:

SELECT pacijent.id, CONCAT(pacijent.ime, ' ', pacijent.prezime) AS ime\_i\_prezime,

pacijent.spol, COALESCE(sos.sos\_broj\_posjeta, 0) AS sos\_broj\_posjeta,

COALESCE(ukupno.ukupna\_posjeta, 0)-COALESCE(sos.sos\_broj\_posjeta, 0) AS broj\_ostalih\_posjeta

    FROM pacijent

LEFT JOIN

    (SELECT posjeta.id\_pacijent, COUNT(posjeta.id) AS sos\_broj\_posjeta

        FROM posjeta, sos\_kontakt

        WHERE posjeta.ime=sos\_kontakt.ime AND posjeta.prezime=sos\_kontakt.prezime

        GROUP BY id\_pacijent) AS sos ON pacijent.id=sos.id\_pacijent

LEFT JOIN

    (SELECT posjeta.id\_pacijent, COUNT(posjeta.id) AS ukupna\_posjeta

        FROM posjeta

        GROUP BY id\_pacijent) AS ukupno ON pacijent.id=ukupno.id\_pacijent;

OPIS UPITA:

Za rješavanje ovoga upita koristit ću tri tablice. Unutar tablice **posjeta** možemo provjeriti koliko je posjeta imao pojedini pacijent. U kombinaciji tablica **sos\_kontakt** i **posjeta** možemo provjeriti koliko je od tih posjeta bilo od strane SOS kontakata pacijenta, a unutar tablice **pacijent** možemo naći podatke o pojedinom pacijentu.

* Prvi korak je pronaći broj SOS posjeta, a to sam učinila na način da sam u FROM dijelu upita navela tablice **posjeta** i **sos\_kontakt** i time kreirala kartezijev produkt.
* U WHERE dijelu upita navela sam dva uvjeta. Prvi uvjet garantira jednakost atributa **ime** u objema tablicama, a drugi uvjet atributa **prezime**.
* Sve skupa sam grupirala prema atributu **id\_pacijent** naredbom GROUP BY.
* U SELECT dijelu upita navela sam atribut **id\_pacijent** (koji će mi biti potreban za daljnje povezivanje podataka o pacijentu) i atribut **sos\_broj\_posjeta** koji sam dobila prebrojavajući atribut **id** iz tablice **posjeta** naredbom COUNT.
* Drugi korak je pronaći ukupan broj posjeta. Tablicu **posjeta** grupirala sam naredbom GROUP BY prema atributu **id\_pacijent**.
* U SELECT dijelu upita navela sam atribut **id\_pacijent** (koji mi je potreban za daljnje povezivanje) i atribut **ukupna\_posjeta** koji sam dobila koristeći naredbu COUNT na atributu **id**.
* Kako bi povezala agregirane podatke sa podacima vezanim uz pojedinog pacijenta prvo sam pomoću naredbe LEFT JOIN pod imenom **sos** i uvjetom jednakosti **id** atributa iz tablice **pacijent** sa atributom **id\_pacijent** iz tablice **sos** dodala prvi rezultat unutar tablice **pacijent**.
* Zatim sam na isti način dodala i drugi rezultat samo pod nazivom **ukupno**.
* U SELECT dijelu navela sam atribute **id**, **ime\_i\_prezime** i **spol** iz tablice **pacijent**, te **sos\_broj\_posjeta** i **broj\_ostalih\_posjeta**.
* Atribut **ime\_i\_prezime** objedinila sam koristeći naredbu CONCAT od atributa **ime** i **prezime** iz tablice **pacijent**.
* Kako bi zamjenila NULL vrijendosti sa nulama, u svrhu računanja, koristila sam naredbu COALESCE na atributima **ukupna\_posjeta** iz tablice **ukupno** i **sos\_broj\_posjeta** iz tablice **sos**.
* Atribut **broj\_ostali\_posjeta** dobila sam oduzimanjem atributa **sos\_broj\_posjeta** od atributa **ukupna\_posjeta** u SELECT dijelu upita.

RJEŠENJE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | ime\_i\_prezime | spol | sos\_broj\_posjeta | broj\_ostalih\_posjeta |
| 400 | Alen Kolić | M | 1 | 0 |
| 401 | Ivan Rupčić | M | 1 | 0 |
| 402 | Lissa Ivić | Ž | 1 | 0 |
| 403 | Klara Zenzerović | Ž | 1 | 0 |
| 404 | Sebastijan Milošević | M | 1 | 0 |
| 405 | Lucas Perić | M | 1 | 0 |
| 406 | Paola Marić | Ž | 1 | 0 |
| 407 | Ema Knežević | Ž | 0 | 1 |
| 408 | Tomi Ivković | M | 0 | 1 |
| 409 | Stefan Markulinčić | M | 0 | 1 |
| 410 | Mia Stepančić | Ž | 0 | 1 |
| 411 | Bruno Marušić | M | 0 | 1 |
| 412 | Iris Orbanić | Ž | 0 | 1 |
| 413 | Nora Marjanović | Ž | 0 | 1 |
| 414 | Borna Karlović | M | 0 | 0 |
| 415 | Oliver Kinić | M | 0 | 0 |
| 416 | Toni Belić | M | 0 | 0 |
| 417 | Petar Dragojević | M | 0 | 0 |
| 418 | Boris Vlašić | M | 0 | 0 |
| 419 | Matija Mladenović | M | 0 | 0 |
| 420 | Filip Perišić | M | 0 | 0 |

## UPIT 6

Svi pacijenti koji su primljeni prije datuma X kod doktora Y a koji još nisu primili terapiju

Doktorica Debeljak (id:208) otišla je na godišnji odmor 30.08.2022, te je bilo potrebno zbrinuti njene pacijente koji su joj dodjeljeni prije odlaska na godišnji odmor, a kojima ona još nije prepisala terapiju.

KOD ZA UPIT:

SELECT pa.\*

FROM pacijent as pa, prijem as pr, doktor as d

WHERE pa.id = pr.id\_pacijent

    AND pr.id\_doktor = d.id

    AND d.id = 208

    AND pr.datum\_prijema < STR\_TO\_DATE('30.08.2022','%d.%m.%Y.')

HAVING pa.id NOT IN (SELECT id\_pacijent FROM terapija);

OPIS UPITA:

Kako bi ostvarili ovaj upit potrebno je povezati pacijente sa datumima kada su zaprimljeni u bolnicu, te sa doktorom koji ih je zaprimio. Tablica **prijem** povezuje ove tri relacije jer ona sadrži id svakog pacijenta, datum prijema i doktora koji je pacijenta primio.

* Najprije smo naveli sve relacije pod naredbu FROM odvajajući ih zarezom i dodjeljujući im novi alias. Time smo ih spojili krartezijevim produktom.
* Nakon toga smo povezali tablice preko id-jeva dodajući uvjete pod naredbu WHERE pri čemu smo naveli da id-jevi iz tablice pacijent moraju odgovarati id-jevima pacijenata iz tablice prijem, te da id-jevi doktora iz tablice prijem moraju odgovarati id-jevima doktora iz tablice doktor. Te uvjete smo lančano povezali naredbom AND
* Također dodali smo još i uvjet da id doktora mora biti 208, što odgovara doktorici Debeljak, te da datum prijema iz tablice prijem mora biti raniji tj. manji od datuma odlaska doktorice na godišnji, koji smo unijeli funkcijom STR\_TO\_DATE koja formatira podatak o datumu iz tipa podatka string u tip podatka date.
* Za kraj smo dodali još jedan uvjet naredbom HAVING gdje smo definirali da se id pacijenta ne smije nalaziti unutar skupa id-jeva iz relacije terapija. Za to smo koristili naredbu NOT IN iza koje slijedi drugi upit koji kao rezultat daje id-jeve svih pacijenata koji su primili terapiju.

REZULTAT:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | ime | prezime | datum\_rodenja | adresa | spol | broj\_telefona | MBO | id\_sos\_kontakt |
| 407 | Ema | Knežević | 1995-10-24 00:00:00 | Busoler 18 | Ž | 0957135987 | 24578946 | 307 |
| 418 | Boris | Vlašić | 1984-07-28 00:00:00 | Puljska cesta 19 | M | 092548796 | 815687459 | 318 |

## UPIT 7

Popis svih pacijenata i ljudi koji su bili u doticaju s njima (doktori, sestre, posjetitelji) na katu broj 2 na kojem je izbila zaraza nakon datuma X

Na katu broj 2 bolnice izbila je zaraza među pacijentima i potrebno je sastaviti popis svih ljudi (uključujući i pacijente) koji su se nakon određenog datuma našli na ovom katu.

KOD ZA UPIT:

CREATE VIEW zarazeni\_pacijenti AS

SELECT p.id,p.ime,p.prezime

FROM pacijent as p, prijem, soba, odjel

WHERE p.id = prijem.id\_pacijent

  AND prijem.id\_soba = soba.id

  AND soba.id\_odjel = odjel.id

  AND kat = 2

  AND datum\_prijema >= STR\_TO\_DATE('24.08.2022','%d.%m.%Y.');

-- popis zarazenih pacijenata

SELECT \*  FROM  zarazeni\_pacijenti

UNION

-- popis svih doktora na temelju zarazenih pacijenata

SELECT DISTINCT doc.id, doc.ime, doc.prezime

FROM doktor as doc, zarazeni\_pacijenti as za , prijem

WHERE za.id = prijem.id\_pacijent

AND doc.id = prijem.id\_doktor

UNION

-- popis svih medicinskih sestara na temelju zarazenih pacijenata

SELECT DISTINCT med.id, med.ime, med.prezime

FROM medicinske\_sestre as med, zarazeni\_pacijenti as za , prijem

WHERE za.id = prijem.id\_pacijent

AND med.id = prijem.id\_medicinske\_sestre

UNION

-- popis svih posjeta pacijentima drugog kata na temelju tablice zarazenih pacijenata

SELECT DISTINCT pos.id, pos.ime, pos.prezime

FROM posjeta as pos, zarazeni\_pacijenti as za

WHERE za.id = pos.id\_pacijent;

OPIS UPITA:

Za ovaj upit najprije je bilo potrebno pronaći sve zaražene pacijente sa drugog kata bolnice nakon zadaog datuma. Za ovaj korak bilo je potrebno povezati relacije **pacijent, prijem, soba i odjel**, budući da relacija prijem povezuje svakog pacijenta sa njegovom sobom, a relacija sobe povezuje svaku sobu sa pripadajućim odjelom, te u tablici odjel nalazimo informaciju na kojem katu bolnice se nalazi koji odjel. Ovime dakle dolazimo do informacije na kojem katu bolnice se nalaze koji pacijenti te kada su primljeni u bolnicu. U sljedećim koracima povezali smo te pacijente sa drugim osobama koje su im dodjeljene kroz tablicu prijem (sestre i doktori) te tablicu posjeta.

* U prvom upitu povezali smo sve relacije **pacijent, prijem, soba i odjel** odvajajući ih zarezom nakon naredbe FROM. Naredbom WHERE povezali smo ih preko pripadajućih id-jeva. Pacijenta i sobu sa prijemom, te sobu i odjel.
* Postavili smo uvjet da se pacijenti moraju nalaziti na 2. katu i da moraju biti primljeni nakon ili na datum 24.08.2022.
* U naredbi SELECT odabrali smo podatke o pacijentima
* Za lakše daljnje korištenje rezultate ovog upita spremili smo u novi pogled zvan zarazeni\_pacijenti postavljanjem naredbe CREATE VIEW zarazeni\_pacijenti AS ispred upita.
* Budući da su pacijenti povezani sa sestrama i doktorima preko relacije prijem, u sljedećem upitu povezali smo tablice doktor, zaraženi\_pacijenti i prijem, također preko id-jeva iza naredbe WHERE te smo iza naredbe SELECT odabrali podatke o doktorima. Također bilo je potrebno dodati i naredbu DISTINCT kako bi eliminirali duplikate, budući da više pacijenata može biti primljeo kod istog doktora.
* Prethodni korak ponovili smo još jednom za novi upit no ovaj puta umjesto tablice doktor povezali smo tablicu medicinske sestre.
* Za kraj povezali smo tablice zaraženi\_pacijenti i posjeta navodeći ih iza naredbe FROM i postavljajući uvjet da id zaraženog pacijenta mora odgovarati id-ju pacijenta iz relacije posjeta. Međusobno smo povezali sve ove upite naredbama UNION i tako došli do konačnog rezultata.

REZULTAT:

|  |  |  |
| --- | --- | --- |
| id | ime | prezime |
| 402 | Lissa | Ivić |
| 403 | Klara | Zenzerović |
| 205 | Jan | Nikolić |
| 207 | Božidarka | Obad |
| 503 | Marko | Marulić |
| 1203 | Issa | Biševac |
| 1204 | Elenora | Delfar |

## UPIT 8

Količine lijekova po nazivima kojima ističe rok trajanja za manje od godinu dana

Odjel nabave obavezan je unaprijed naručiti nove zalihe lijekova kojima ističe rok za manje od godinu dana, stoga je potrebno sastaviti popis i stanje takvih zaliha lijekova u bolnici.

Lijekovi kojima je već istekao rok trajanja su također u evidenciji sve do njihovog zbrinjavanja i odlaganja, no u ovom slučaju nas zanimaju samo lijekovi sa još uvijek valjanim rokom uporabe koji se trenutno koriste u bolnici.

KOD ZA UPIT:

SELECT li.id, li.naziv, st.količina, st.rok\_valjanosti,

(SELECT DATEDIFF(rok\_valjanosti,(SELECT NOW() FROM DUAL))) as dani\_do\_isteka

FROM lijek as li

INNER JOIN stanje\_lijekova as st ON li.id = st.id\_lijek

HAVING dani\_do\_isteka BETWEEN 0 AND 365

ORDER BY dani\_do\_isteka ASC;

OPIS UPITA:

Informacije koje su nam potrebne za ovaj upit su naziv, količina i rok valjanosti, a nalaze se u dvije tablice, lijek i stanje\_lijekova.

* Relaciju lijek navodimo iz naredbe FROM i dodjeljujemo joj alias li
* Spajamo ovu relaciju INNER JOIN-om sa relacijom stanje\_lijekova preko id-jeva lijeka
* Iza naredbe HAVING navodimo uvjet korištenjem riječi BETWEEN da dani do isteka moraju biti između 0 i 365 kako bi eliminirali sve lijekove kojima je rok već istekao i kojima će tek isteći ali za više od godinu dana.
* Listu lijekova poredali smo naredbom ORDER BY dani\_do\_isteka prema rastu naredbom ASC na kraju retka kako bi na početku liste dobili one lijekove kojima rok najranije istječe.
* Za kraj iza naredbe SELECT navodimo podatke id i naziv iz tablice lijek, količina i rok valjanosti iz tablice stanje\_lijekova, te dane do isteka valjanosti koje računamo koreliranim podupitom koristeći funkciju DATEDIFF koja prima dva datuma, a kao rezultat vraća broj dana između njih. Za prvi datum navodimo rok\_valjanosti, a za drugi trenutni datum za vrijeme upita koristeći funkciju NOW koja vraća trenutni datum. Ovaj podupit izvršiti će se za svaku n-torku relacije i rezultat će biti prikazan u stupcu dani\_do\_isteka.

REZULTAT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | naziv | količina | rok\_valjanosti | dani\_do\_isteka |
| 812 | Meropenem | 8756 | 2022-07-02 | 38 |
| 822 | Haloperidol | 356 | 2022-07-05 | 41 |
| 814 | Gentamycin | 890 | 2022-09-07 | 105 |
| 820 | Heparin | 650 | 2022-09-10 | 108 |
| 804 | Ketamine | 1201 | 2022-09-15 | 113 |
| 824 | Cyclizine | 630 | 2022-10-07 | 135 |
| 805 | Bisoprolol | 610 | 2023-02-16 | 267 |
| 818 | Rivaroxaban | 980 | 2023-03-22 | 301 |
| 813 | Vancomycin | 312 | 2023-04-12 | 322 |

## UPIT 9

Koliko je koji doktor propisao lijekova pacijentima u razdoblju između X i Y i tko su doktori koji su prepisali naviše lijekova pacijentima u tom razdoblju?

KOD ZA UPIT:

CREATE VIEW broj\_izdavanja\_lijeka AS

SELECT doc.id, doc.ime, doc.prezime, COUNT(doc.id) as ukupno\_izdano\_lijekova

FROM pacijent as pa

INNER JOIN prijem as pr ON pa.id = pr.id\_pacijent

INNER JOIN doktor as doc ON doc.id = pr.id\_doktor

INNER JOIN terapija as te ON pa.id = te.id\_pacijent

WHERE  datum\_prijema BETWEEN STR\_TO\_DATE('22.08.2022','%d.%m.%Y.') AND STR\_TO\_DATE('29.08.2022','%d.%m.%Y.')

GROUP BY doc.id

ORDER BY ukupno\_izdano\_lijekova DESC;

-- ukupno po doktoru

SELECT \*

FROM broj\_izdavanja\_lijeka;

-- doktori sa najviše izdanih lijekova

SELECT \*

FROM broj\_izdavanja\_lijeka

WHERE ukupno\_izdano\_lijekova = (SELECT MAX(ukupno\_izdano\_lijekova) FROM broj\_izdavanja\_lijeka);

OPIS UPITA:

Kako bi povezali doktora sa brojem izdanih terapija (svaka pojedina terapija prepisuje samo jedan lijek! ) tj. brojem izdanih lijekova potrebno je povezati tablice pacijent, prijem, doktor, i terapija. Tablica prijem povezuje pacijente i doktore, a tablica terapija povezuje pacijente i lijekove koji su im prepisani. Povezivanjem ovih tablica u konačnici ćemo povezati doktore sa lijekvima.

* Upit krećemo od tablice pacijent koju navodimo nakon naredbe FROM i dodjeljujemo joj kraći alias
* Nakon toga radimo INNER JOIN pacijenta sa prijemom, te doktora sa prijemom preko pripadajućih id-jeva te na kraju povezujemo terapiju sa pacijentom na također isti način.
* Nakon naredbe WHERE naredbom BETWEEN postavljamo uvjet između dva datuma koja unosimo funkcijom STR\_TO\_DATE koja datum tipa string pretvara u tip podatka date nad kojim se mogu vršiti razne operacije.
* Narebom GROUP BY grupiramo tablicu prema id-ju doktora kako bi se mogla izvršiti funkcija prebrojavanja COUNT.
* Konačni rezultat sortiramo naredbom ORDER BY prema ukupnom broju izdanih lijekova u padajućem smjeru ključnom riječi DESC.
* Za kraj iza naredbe SELECT specificiramo podatke koje želimo u rezultatu, a to su id, ime, prezime doktora iz tablice doktor (doc) i ukupno\_izdano\_lijekova dobiveno korištenjem funkcije COUNT kojom smo prebrojali prethodno grupirane id-jeve doktora.
* Cjelokupni upit pohranili smo u pogled broj\_izdavanja\_lijeka naredbom CREATE VIEW broj\_izdavanja\_lijeka AS prije naredbe SELECT.
* Selektiranjem svih atributa iz pogleda broj\_izdavanja\_lijeka dolazimo do broja izdanih lijekova po doktoru
* Kako bi dobili doktore koji su izdali najviše lijekova iza naredbe WHERE postavljamo uvjet da ukupno\_izdano\_lijekova mora biti jednako najvišoj vrijednosti iz tog upita. Tu vrijednost dobivamo korištenjem funkcije MAX u podupitu koja vraća najveću vrijednost stupca ukupno\_izdano\_lijekova.

REZULTATI:

|  |  |  |  |
| --- | --- | --- | --- |
| id | ime | prezime | ukupno\_izdano\_lijekova |
| 200 | Krešimira | Paspalj | 5 |
| 208 | Jelena | Debeljak | 4 |
| 201 | Noris | Grubor | 2 |
| 207 | Božidarka | Obad | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| id | ime | prezime | ukupno\_izdano\_lijekova |
| 200 | Krešimira | Paspalj | 5 |

## UPIT 10

Kako bi ustvrdili koje kategorije lijekova se u bolnici najviše koriste napravili smo upit koji će prebrojati koliko je pacijenata primilo koju vrstu lijeka uključujući i lijekove koji nisu prepisani niti jednom pacijentu.

KOD ZA UPIT:

CREATE VIEW pacijenti\_po\_lijeku AS

SELECT vrsta as vrsta\_lijeka, COUNT(vrsta) as broj\_pacijenata

FROM pacijent as pa

INNER JOIN terapija as te ON pa.id = te.id\_pacijent

INNER JOIN lijek as li ON li.id = te.id\_lijek

GROUP BY vrsta\_lijeka;

SELECT DISTINCT li.vrsta as vrsta\_lijeka, COALESCE(broj\_pacijenata, 0) as broj\_pacijenata

FROM lijek as li

LEFT JOIN pacijenti\_po\_lijeku ON li.vrsta = pacijenti\_po\_lijeku.vrsta\_lijeka

ORDER BY broj\_pacijenata DESC, vrsta\_lijeka ASC;

OPIS UPITA:

Tablica terapija povezuje svakog pacijenta sa lijekom koji mu je prepisan. Jedan pacijent može imati više terapija, tj. prepisanih lijekova. Budući da tablica terapija sadrži samo id lijeka, a ne i podatke o njemu, na to moramo povezati još i tablicu lijek koja sadrži podatak o vrsti lijeka koji je prepisan.

Potom ćemo vršiti grupiranje i prebrojavanje prema vrsti lijeka te ćemo to spojiti sa relacijom koja sadrži sve vrste lijekova u bolnici kako bi u rezultat uključili i lijekove koji nisu prepisani niti jednom pacijentu.

* Upit započinjemo navođenjem relacije pacijent iza naredbe FROM te mu dodjeljujemo kraći alias pa.
* Naredbama INNER JOIN povezujemo relacije terapija i lijek sa pacijentom prema pripadajućim id-jevima relacija.
* Grupiramo rezultat prema vrsti lijeka koristeći naredbu GROUP BY
* U naredbi SELECT odabiremo stupce prikaza, a to su vrsta lijeka i broj pacijenata dobiven prebrojavanjem stupca vrsta funkcijom COUNT.
* Ovim upitom dobili smo sve vrste lijekova koji su prepisani nekim pacijentima i broj pacijenata. Rezultat smo pohranili u pogled pacijenti\_po\_lijeku naredbom CREATE VIEW pacijenti\_po\_lijeku AS ispred naredbe SELECT
* Kako bi u rezultat uključili i lijekove koji nisu prepisani niti jednom pacijentu postavljamo novi upit.
* Tablica lijek sadrži popis svih lijekova u bolnici, stoga ćemo iz nje selektirati stupac vrsta, no budući da bolnica ima mnogo lijekova različitog naziva no iste vrste, potrebno je koristiti naredbu DISTINCT kako bi se eliminirali duplikati.
* U naredbi FROM navodimo tablicu lijek
* Sada ćemo ovu tablicu koja sadrži vrste svih lijekova, spojiti sa prethodnom tablicom pohranjenom u pogledu pacijenti\_po\_lijeku koristeći naredbu LEFT JOIN, a spajanje ćemo vršiti prema stupcu vrsta
* Za kraj rezultat ćemo sortirati prema broju pacijenata padajuće naredbom DESC, te potom prema vrsti rastuće naredbom ASC.
* Potrebno je u naredbu SELECT još uključiti stupac koji sadrži broj pacijenata, no kako bi izbjegli NULL vrijednosti u rezultatu koristiti ćemo funkciju COALESCE koja će iz zadanog seta odabrati prvi rezultat koji nije jednak NULL. Time će vrste lijekova koje nemaju niti jednog pacijenta biti evidentirane kao 0 umjesto NULL.

REZULTAT:

|  |  |
| --- | --- |
| vrsta\_lijeka | broj\_pacijenata |
| Antibiotik | 7 |
| Anelgetik | 4 |
| Antiaritmik | 2 |
| Antikoagulans | 1 |
| Antiemetik | 0 |
| Sedativ | 0 |

## UPIT 11

Prikaz svih medicinskih sestara i pacijenta za koje su one zadužene, poredane po atributu id\_pacijent. Cilj je bio prikazati sve pacijente i sve medicinske sestre koje su zadužene za njih te sobe i uz sobe id odjela na kojemu se i sobe i pacijenti, a i sestre nalaze.

SELECT m.id as ID ,CONCAT(m.ime ,' ', m.prezime)as Ime\_i\_prezime,pac.id, pac.ime as ime\_pacijenta,pac.prezime as prezime\_pacijenta, s.broj\_sobe,s.id\_odjel

  FROM medicinske\_sestre as m, prijem as p,pacijent as pac, soba as s

  WHERE m.id=p.id\_medicinske\_sestre AND p.id\_pacijent=pac. id AND p.id\_soba=s.id

  ORDER BY pac.id DESC;

**OPIS UPITA:**

Ovaj upit smo napravili tako da smo u SELECT dijelu odabrali attribute id, ime i prezime iz tablice medicinske\_sestre. Atribute ime i prezime smo spojili naredbom CONCAT u jedan atribut te smo ga preimenovali u Ime\_i\_prezime, te smo iz tablice pacijent uzeli attribute ime, prezime. Iz sobe smo uzeli attribute broj\_sobe i id\_odjel. Te smo te 4 tablice spojili u FROM dijelu što bi odgovaralo njihovom kartezijevom produktu.

Da bi pridružili pacijente medicinskim sestrama i sobe te odjele pacijentima i medicinskim sestrama morali smo to sve napraviti preko tablice prijem.

Tako smo u WHERE dijelu upita odabrali da se tablica medicinske\_sestre i prijem spoje po primarnom ključu iz tablice medicinske sestre, a sve ostale kombinacije se ne prikazuju. To smo isto napravili sa pacijentima i sobama koje smo također preko primarnog ključa povezali sa prijemom.

Na kraju smo to poredali prema atributu pacijent.id silazno.

REZULTAT:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Ime\_i\_prezime | id | ime\_pacijenta | prezime\_pacijenta | broj\_sobe | id\_odjel |
| 504 | Žarka Stanić | 420 | Filip | Perišić | 3 | 102 |
| 504 | Žarka Stanić | 419 | Matija | Mladenović | 3 | 102 |
| 504 | Žarka Stanić | 418 | Boris | Vlašić | 3 | 102 |
| 504 | Žarka Stanić | 417 | Petar | Dragojević | 5 | 102 |
| 504 | Žarka Stanić | 416 | Toni | Belić | 5 | 102 |
| 504 | Žarka Stanić | 415 | Oliver | Kinić | 5 | 102 |
| 504 | Žarka Stanić | 414 | Borna | Karlović | 5 | 102 |
| 502 | Ivana Marić | 413 | Nora | Marjanović | 4 | 102 |
| 502 | Ivana Marić | 412 | Iris | Orbanić | 4 | 102 |
| 502 | Ivana Marić | 411 | Bruno | Marušić | 4 | 102 |
| 502 | Ivana Marić | 410 | Mia | Stepančić | 4 | 102 |
| 504 | Žarka Stanić | 409 | Stefan | Markulinčić | 3 | 102 |
| 504 | Žarka Stanić | 408 | Tomi | Ivković | 3 | 102 |
| 502 | Ivana Marić | 407 | Ema | Knežević | 1 | 102 |
| 502 | Ivana Marić | 406 | Paola | Marić | 1 | 102 |
| 501 | Miliana Milić | 405 | Lucas | Perić | 9 | 101 |
| 501 | Miliana Milić | 404 | Sebastijan | Milošević | 9 | 101 |
| 503 | Marko Marulić | 403 | Klara | Zenzerović | 6 | 100 |
| 503 | Marko Marulić | 402 | Lissa | Ivić | 6 | 100 |
| 500 | Ivana Ivić | 401 | Ivan | Rupčić | 4 | 100 |
| 500 | Ivana Ivić | 400 | Alen | Kolić | 4 | 100 |

## UPIT 12

Prikaz broja pacijenata dodijeljenih svakoj medicinskoj sestri gdje se može vidjeti opseg posla svake sestre te njeno opterećenje na odjelu. Prikazuju se medicinske sestre koje imaju broj pacijenata veći od 5 .

  CREATE VIEW Medicinari AS

SELECT  p.id\_medicinske\_sestre,m.ime,m.prezime, COUNT(\*) as broj\_pacijenata

FROM prijem as p, medicinske\_sestre as m

     WHERE m.id=p.id\_medicinske\_sestre

     group by p.id\_medicinske\_sestre

     ORDER BY broj\_pacijenata DESC;

-- prikaz broja pacijenata po medicinskoj sestri

SELECT \* from Medicinari;

--prikaz medicinskih sestara koje imaju pridružen broj pacijenata koji je veći od 5

SELECT \* from Medicinari WHERE broj\_pacijenata>5;

**OPIS UPITA:**

Da znali id medicinske sestre u select dijelu odabiremo atribut id, te atribute ime i prezime, te unutar FROM dijela spajamo dvije tablice prijem i medicinske\_sestre i dobivamo kartezijev produkt. Da bi dobiveni rezultat imao smisla te se medicinske sestre povezale sa tablicom prijem unutar odabiremo uvjet WHERE, gdje id medicinske sestre odgovara id medicinske sestre unutar tablice prijem. Uz pomoć agregacijske funkcije COUNT zbrajamo sve medicinske sestre koje su grupirane po atributu id medicinske sestre.Poredali smo dobiveni rezultat uz pomoć naredbe ORDER BY silazno prema broju pacijenata.

Tako dobivenu tablicu smo sporemili kao pogled te je sada unutar nje moguće doći do podatka koliko je medicinska sestra obavila prijema i koji su pacijenti o kojima ona brine.

Sve sestre koje brinu o više od 5 pacijenata trebalo bi rateretiti.

PRIKAZ REZULTATA:

* Prikaz pogleda Medicinari

|  |  |  |  |
| --- | --- | --- | --- |
| id\_medicinske\_sestre | ime | prezime | broj\_pacijenata |
| 504 | Žarka | Stanić | 9 |
| 502 | Ivana | Marić | 6 |
| 500 | Ivana | Ivić | 2 |
| 501 | Miliana | Milić | 2 |
| 503 | Marko | Marulić | 2 |

* Prikaz medicinskih sestara koje su upisale i brinu se za više od 5 pacijenata

|  |  |  |  |
| --- | --- | --- | --- |
| id\_medicinske\_sestre | ime | prezime | broj\_pacijenata |
| 504 | Žarka | Stanić | 9 |
| 502 | Ivana | Marić | 6 |

## UPIT 13

Cilj nam je ovim upitom bio prikazati broj zaposlenih po godinama starosti, te izračunati prosječnu starost osoblja, najstariju i najstarije osobe, prikaz zaposlenika koji imaju više godina od prosječne starosti zaposlenih.

CREATE VIEW godine AS

SELECT \*, CURDATE() as sadasnji\_datum, TIMESTAMPDIFF(YEAR, datum\_rodenja, CURDATE())AS age FROM doktor

UNION

SELECT\*, CURDATE() as sadasnji\_datum,  TIMESTAMPDIFF(YEAR, datum\_rodenja, CURDATE())AS age FROM medicinske\_sestre

ORDER BY datum\_rodenja DESC;

-- koliko g ima najstariji zaposlenik

SELECT  MAX(age)

  FROM godine;

-- broj zaposlenika

SELECT  COUNT(\*) as broj\_zaposlenika

  FROM godine;

-- godine najmlađeg zaposlenika

SELECT MIN(age) FROM godine;

--prosječan broj godina svih zaposlenih

SELECT AVG(age) FROM godine;

-- PRVA tri najstarija zaposlenika unutar bolnice

SELECT \* FROM GODINE ORDER BY age DESC LIMIT 3 ;

-- saznaj koji zaposlenici imaju više godina od prosječne dobi

SELECT \*

  FROM godine

  WHERE age >(SELECT AVG(age) FROM godine);

**OPIS UPITA**

Unutar SELECT dijela smo odabrali sve iz tablica doktor i medicinske\_sestre te smo još dodali funkciju CURDATE() koja vraća sadašnji datum i nju smo usporedili sa datumom rođenja sa funkcijom TIMESTAMPDIF. To smo napravili za obije tablice te smoih tada povezali naredbom UNION.

Tu tablicu smo spremili kao pogled te nad njom izvršili upite MAX da saznamo broj godina najstarijeg zaposlenika, uz pomoć COUNT funkcije smo pobrojali broj zaposlenika, uz pomoć MIN funkcije saznali smo broj godina najmlađeg zaposlenika, uz pomoć funkcije SUM saznali smo prosječan broj godina unutar tablice godine. Da bi saznali koji zaposlenici imaju više od prosječne dobi izvršili smo podupit gdje smo prvo našli prosječnu dob pa odabrali uvjet gdje se prikažu zaposlenici sa većim brojem godina.

REZULTATI PRETRAŽIVANJA:

* Prikaz zaposlenika prema broju godina

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | ime | prezime | datum\_rođenja | id\_odjel | age |
| 503 | Marko | Marulić | 2000-03-13 00:00:00 | 100 | 22 |
| 501 | Miliana | Milić | 1999-02-11 00:00:00 | 101 | 23 |
| 502 | Ivana | Marić | 1998-02-01 00:00:00 | 102 | 24 |
| 500 | Ivana | Ivić | 1992-02-11 00:00:00 | 100 | 30 |
| 504 | Žarka | Stanić | 1992-02-05 00:00:00 | 102 | 30 |
| 207 | Božidarka | Obad | 1990-06-03 00:00:00 | 100 | 31 |
| 208 | Jelena | Debeljak | 1988-03-28 00:00:00 | 102 | 34 |
| 203 | Jasmina | Mejak | 1985-06-03 00:00:00 | 101 | 36 |
| 201 | Noris | Grubor | 1982-05-03 00:00:00 | 100 | 40 |
| 200 | Krešimira | Paspalj | 1978-01-11 00:00:00 | 102 | 44 |
| 206 | Gabrijel | Popović | 1976-05-01 00:00:00 | 101 | 46 |
| 202 | Šime | Ljubić | 1973-12-09 00:00:00 | 101 | 48 |
| 204 | Mauro | Tomasov | 1970-07-24 00:00:00 | 100 | 51 |
| 209 | Rudolf | Kolar | 1967-09-16 00:00:00 | 101 | 54 |
| 205 | Jan | Nikolić | 1962-08-22 00:00:00 | 100 | 59 |

* Prikaz broja godina najstarijeg zaposlenika

|  |
| --- |
| MAX(age) |
| 59 |

* Prikaz broja zaposlenika

|  |
| --- |
| broj\_zaposlenika |
| 15 |

* Godine najmlađeg zaposlenika

|  |
| --- |
| MIN(age) |
| 20 |

* Prosječan broj godina

|  |
| --- |
| AVG(age) |
| 38.1333 |

* Prva tri najstarija zaposlenika unutar bolnice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | ime | prezime | datum\_rođenja | id\_odjel | age |
| 205 | Jan | Nikolić | 1962-08-22 00:00:00 | 100 | 59 |
| 209 | Rudolf | Kolar | 1967-09-16 00:00:00 | 101 | 54 |
| 204 | Mauro | Tomasov | 1970-07-24 00:00:00 | 100 | 51 |

* Zaposlenici koji imaju više godina od prosječne dobi zaposlenika

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | ime | prezime | datum\_rođenja | id\_odjel | sadasnji\_datum | age |
| 201 | Noris | Grubor | 1982-05-03 00:00:00 | 100 | 2022-05-29 | 40 |
| 200 | Krešimira | Paspalj | 1978-01-11 00:00:00 | 102 | 2022-05-29 | 44 |
| 206 | Gabrijel | Popović | 1976-05-01 00:00:00 | 101 | 2022-05-29 | 46 |
| 202 | Šime | Ljubić | 1973-12-09 00:00:00 | 101 | 2022-05-29 | 48 |
| 204 | Mauro | Tomasov | 1970-07-24 00:00:00 | 100 | 2022-05-29 | 51 |
| 209 | Rudolf | Kolar | 1967-09-16 00:00:00 | 101 | 2022-05-29 | 54 |
| 205 | Jan | Nikolić | 1962-08-22 00:00:00 | 100 | 2022-05-29 | 59 |

## UPIT 14

Stvorili smo pogled svih zaposlenih medicinskih sestara i doktora unutar bolnice. Unutar pogleda moguće je izabrati koji radnici rade na kojim odjelima, koliko je ukupno radnika na svakom odjelu.

 CREATE VIEW zaposleni\_na\_odjelu AS

SELECT m.id, m.ime, m.prezime, o.naziv FROM medicinske\_sestre as m

INNER JOIN odjel as o ON o.id=m.id\_odjel

UNION

SELECT d.id, d.ime,d.prezime, o.naziv FROM doktor as d

INNER JOIN odjel as o ON o.id= d.id\_odjel

ORDER BY naziv DESC;

-- odaberi sve koji su zaposleni na odjelu kardiologije

SELECT \*

FROM zaposleni\_na\_odjelu

WHERE naziv='Kardiologija';

OPIS UPITA:

Unutar SELECT dijela odabrali smo iz tablice medicinske\_sestre atribute id, ime, prezime, a iz tablice odjel naziv odjela, te smo spojili tablicu odjel sa tablicom medicinske\_sestre pomoću funkcije INNER JOIN koja je povezala atribut id iz tablice odjel i atribut id\_odjel iz tablice medicinske sestre koji služi kao strani ključ. Isto smo napravili i za liječnike te smo ih povezali sa naredbom UNION. Dobivene retke smo dobili ispod medicinskih sestara. To smo spremili kao pogled te smo nad tim pogledom izvršili upit kako bi saznali zaposlenike odjela Kardiologije.

REZULTATI:

* Tablica zaposlenih po odjelima poredanih po odjelima

|  |  |  |  |
| --- | --- | --- | --- |
| id | ime | prezime | naziv |
| 502 | Ivana | Marić | Ortopedija |
| 504 | Žarka | Stanić | Ortopedija |
| 200 | Krešimira | Paspalj | Ortopedija |
| 208 | Jelena | Debeljak | Ortopedija |
| 500 | Ivana | Ivić | Kardiologija |
| 503 | Marko | Marulić | Kardiologija |
| 201 | Noris | Grubor | Kardiologija |
| 204 | Mauro | Tomasov | Kardiologija |
| 205 | Jan | Nikolić | Kardiologija |
| 207 | Božidarka | Obad | Kardiologija |
| 501 | Miliana | Milić | Intenzivno liječenje |
| 202 | Šime | Ljubić | Intenzivno liječenje |
| 203 | Jasmina | Mejak | Intenzivno liječenje |
| 206 | Gabrijel | Popović | Intenzivno liječenje |
| 209 | Rudolf | Kolar | Intenzivno liječenje |

* Zaposleni na kardiologiji

|  |  |  |  |
| --- | --- | --- | --- |
| id | ime | prezime | naziv |
| 500 | Ivana | Ivić | Kardiologija |
| 503 | Marko | Marulić | Kardiologija |
| 201 | Noris | Grubor | Kardiologija |
| 204 | Mauro | Tomasov | Kardiologija |
| 205 | Jan | Nikolić | Kardiologija |
| 207 | Božidarka | Obad | Kardiologija |