

Primjena geoprostorne baze podataka u lociranju odgovarajuće zdravstvene ustanove

Bardić, Lucija

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, School of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:126:562069>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-27**



mathos

Repository / Repozitorij:

[Repository of School of Applied Mathematics and Informatics](#)



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJ



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE

Studij

Sveučilišni diplomski studij matematike

modul: Matematika i računarstvo

Primjena geoprostorne baze podataka u lociranju odgovarajuće zdravstvene ustanove

DIPLOMSKI RAD

Mentor:

doc. dr. sc. Mateja Đumić

Student:

Lucija Bardić

Osijek, 2024.

Sadržaj

1	Uvod	1
2	Baze podataka	2
2.1	Povijest baza podataka	3
2.2	Sustav za upravljanje bazom podataka	4
3	Geografski informacijski sustav	7
3.1	Povijest geografskih informacijskih sustava	8
3.2	Primjena geografskih informacijskih sustava	9
4	Geoprostorne baze podataka	10
4.1	Geoprostorni podaci	10
4.1.1	Vektorski podaci	10
4.1.2	Raster podaci	11
4.1.3	Usporedba vektorskih i rasterskih podataka	13
4.2	Geoprostorno indeksiranje	13
5	Alati	17
5.1	Docker	17
5.2	PostgreSQL	17
5.3	PostGIS	18
5.3.1	Povijest PostGISa	19
5.3.2	Geometrija u PostGISu	20
5.3.3	Geografija u PostGISu	23
5.3.4	Topologija u PostGISu	23
5.3.5	Funkcije u PostGISu	24
5.4	QGIS	27
6	Primjena geoprostorne baze podataka	28
6.1	Podaci	28
6.2	Baza podataka	29
6.3	Upiti na bazi	34
6.3.1	Pronalazak najbliže zdravstvene ustanove	34
6.3.2	Pronalazak zdravstvenih ustanova u danom radijusu	37
	Literatura	39

Sažetak	41
Summary	42
Životopis	43

1 | Uvod

Oduvijek je postojala potreba za spremanjem i upravljanjem podacima, a 1970-ih godina započinje razvoj i proučavanje baza podataka. Dodatnim razvojem računala i aplikacija, pojavila se potreba za radom sa sve većim brojem podataka, te se sve veći naglasak stavlja na brzinu pretraživanja tih podataka. Klasične relacijske baze podataka pokazale su se vrlo učinkovitima za strukturirane podatke, no skup upita koji se može izvršiti nad ovim bazama je ograničen i ne daje odgovore na pitanja o geoprostornim odnosima, kao što su udaljenost ili susjedstvo. Ova ograničenja dovela su do potrebe za razvojem geoprostornih baza podataka koje mogu pohranjivati i analizirati geoprostorne podatke.

Upravo su u tom kontekstu geografski informacijski sustavi (GIS) postali ključni jer omogućuju prikupljanje, pohranu, analizu i vizualizaciju geoprostornih podataka. Geoprostorni podaci odnose se na neku lokaciju na Zemlji ili na neko geografsko područje, a mogu biti predstavljeni točkama, linijama, poligonima ili nekim složenijim oblicima.

Korištenje geoprostornih baza podataka u GISu omogućava korisnicima da osim vizualizacije geoprostornih podataka provode i složene geoprostorne analize, kao što su procjena rizika od prirodnih katastrofa, optimizacija ruta i analiza tržišta čime se omogućava donošenje informiranih odluka.

Rad je organiziran na sljedeći način: u drugom poglavlju objašnjene su baze podataka, različiti tipovi baza podataka, povijest razvoja te što je to sustav za upravljanje bazom podataka, za što se koristi ovaj sustav i koje su mu prednosti i nedostaci. Treće poglavlje detaljnije objašnjava geografske informacijske sustave i podsustave te njihove komponente. Također, detaljno je opisana njihova povijest i primjena u stvarnome svijetu. U četvrtom poglavlju objašnjene su geoprostorne baze podataka, vrste geoprostornih podataka te je dana njihova usporedba. Dodatno, za bolje razumijevanje geoprostorne baze podataka, opisano je geoprostorno indeksiranje. U sklopu praktičnog dijela ovog diplomskog rada, napravljen je primjer u PostGISu koji pomaže pri pronalasku najbliže zdravstvene ustanove na području grada Osijeka s obzirom na hitnost pacijenta i danu lokaciju, te omogućuje pronalazak svih zdravstvenih ustanova u danom radijusu. U petom poglavlju opisani su svi alati korišteni za izradu praktičnog dijela rada, a u šestom je detaljno opisana implementacija i prikazani su rezultati.

2 | Baze podataka

Baza podataka je skup podataka koji je posebno organiziran za brzo pretraživanje i pronalazak informacija od strane računala. Baze podataka se mogu koristiti za spremanje i upravljanje velikom količinom strukturiranih i nestrukturiranih podataka te za aktivnosti poput analize i pretraživanja podataka.

Bazom podataka obično upravlja sustav za upravljanje bazom podataka (SUBP, engl. *database management system - DBMS*). SUBP je softver koji dopušta korisnicima da komuniciraju s jednom ili više baza podataka te omogućuje pristup podacima unutar istih. Baza podataka, SUBP i povezane aplikacije zajedno čine sustav baze podataka.

Postoje različite vrste baza podataka, a izbor koju koristiti ovisi o potrebama i problemima na koje se želi odgovoriti. Neke od najpopularnijih baza podataka su sljedeće:

- Relacijska baza podataka - Podaci se spremaju u više povezanih tablica u kojima su raspoređeni u stupce i retke. Relacijski sustav za upravljanje bazama podataka (RSUBP, engl. *relational database management system - RDBMS*), omogućuje kreiranje, ažuriranje i uređivanje relacijskih baza podataka. Najčešći jezik koji se koristi za čitanje, kreiranje, ažuriranje i brisanje podataka u ovoj bazi podataka je SQL, tj. strukturirani upitni jezik (engl. *Structured Query Language*). Relacijske baze su popularne jer im je razina pouzdanosti visoka te podržavaju ACID svojstva¹. Ova vrsta baze podataka radi dobro sa strukturiranim podacima pa organizacije koje imaju puno nestrukturiranih podataka ne bi trebale koristiti ovaj tip baze podataka.
- Objektno-orijentirana baza podataka - Temelji se na objektno-orijentiranom pristupu tako da su svi podaci prikazani u obliku objekata. Objektno-orijentiranom bazom podataka upravlja objektno-orijentirani sustav za upravljanje bazama podataka (OOSUBP, engl. *object-oriented database management systems - OODBMS*). Ova baza podataka radi dobro s objektno-orijentiranim programskim jezicima, poput C++ i Jave, te kao i relacijska baza podataka podržava ACID svojstva.

¹ACID je akronim koji se odnosi na skup od 4 ključna svojstva koja definiraju transakciju: atomarnost, dosljednost, izolacija i trajnost (engl. *atomicity, consistency, isolation, and durability*). Transakcija baze podataka predstavlja jedinicu rada koja se izvodi unutar SUBPa, ili sličnog sustava, prema bazi podataka i tretira se na koherentan i pouzdan način neovisno o drugim transakcijama, što znači da se ona ili cijela izvrši ili cijela poništi.

- NoSQL² baza podataka - Velika kategorija baza podataka koja obuhvaća svaku bazu podataka koja ne koristi SQL kao svoj primarni jezik za pristup podacima. Ovakva baza podataka ponekad se naziva i nerelacijska baza podataka, iako NoSQL baza podataka nije nužno nerelacijska.. Za razliku od relacijske baze podataka, ova vrsta je izvrstan izbor za organizacije čiji podaci nisu strukturirani. Također, velika prednost NoSQL baze podataka je ta što programeri mogu raditi promjene "u hodu", bez da te promjene utječu na aplikacije koje koriste tu bazu podataka.
- Baze podataka u oblaku - Odnosi se na bilo koju bazu podataka dizajniranu za rad u oblaku. Baza podataka u oblaku nudi fleksibilnost, skalabilnost i visoku razinu dostupnosti.
- Dokumentna baza podataka - Koristi dokumente u JSON formatu³ za modeliranje podataka umjesto korištenja redova i stupaca. Dokumentna baza podataka je dizajnirana da pohranjuje i upravlja informacijama orijentiranim na dokumente. Ova baza podataka je jednostavna i skalabilna, što ju čini korisnom za mobilne aplikacije kojima su potrebne brze iteracije.
- Geoprostorna baza podataka - Vrsta baze podataka koja sadrži geoprostorne podatke, koji predstavljaju objekte definirane u prostoru, te alate za upite i analizu takvih podataka. S obzirom da je glavni fokus u ovom radu upravo na ovom tipu baze podataka o njima će još biti dodatno rečeno u nastavku.

2.1 Povijest baza podataka

Ljudi su počeli pratiti i zapisivati podatke i informacije puno prije nego što su izumljena računala. Tada su podaci čuvani u dnevnicima, knjižnicama i kabinetima i, iako je takav način zahtijevao puno prostora i vremena za pretraživanje, to su bile preteče baza podataka.

Pojava računala u ranim 1960-im godinama označila je početak računalnih baza podataka. Charles Bachman, američki informatičar, dizajnirao je prvu bazu podataka poznatu kao *Integrated Data Store*, nakon čega je i IBM razvio svoju pod nazivom *Information Management System*. Ove dvije baze podataka bile su navigacijske, tj. zahtijevale su od korisnika da navigira kroz njih kako bi došao do željenih informacija. Dva najpopularnija modela navigacijskih baza podataka bila su hijerarhijski model IMS razvijen od strane IBMa te mrežni model CODASYL. Sustav baza podataka iz ovog vremena koji se pokazao kao komercijalni uspjeh bio je sustav SABRE koji je koristila aviokompanija American Airlines u upravljanju podacima o rezervacijama.

²NoSQL je skraćenica za "not only SQL" što se odnosi na baze podataka koje spremaju podatke u strukture koje se najčešće razlikuju od onih u relacijskim bazama podataka.

³JSON, odnosno JavaScript Object Notion, je otvoreni standardni format datoteke i format za razmjenu podataka koji koristi tekst čitljiv ljudima.

Tijekom 1970-ih E. F. Codd je izdao članak pod nazivom "*A Relational Model of Data for Large Shared Data Banks*" koji je označio početak relacijskih baza podataka. Ovo je potaklo razvoj relacijske baze podataka INGRES, na Sveučilištu u Kaliforniji, u Berkeleyju, koji je koristio QUEL upitni jezik. IBM je proizveo vlastitu relacijsku bazu podataka, *System R*, koja je prva koristila strukturirani upitni jezik, tj. SQL (engl. *Structured Query Language*).

Tijekom 1980-ih dolazi do rasta i standardizacije relacijskih baza podataka. Komercijalizacija relacijskih sustava dovela je do rasta u njihovoj upotrebi i popularnosti te je SQL postao standardni jezik za baze podataka. Također, ovo razdoblje obilježio je nastanak objektno-orijentiranih sustava za upravljanje bazom podataka (OOSUBP, engl. *object-oriented database management systems - OODBMS*) koji su na podatke gledali kao na objekte i radili s programskim jezicima koji podržavaju objektno-orijentirani pristup.

Tijekom 1990-ih dolazi do razvoja World Wide Weba čime se pojavljuje sve veća potreba za klijent-server sustavima baze podataka te dolazi do eksponencijalnog rasta industrije baza podataka. Relacijski sustav za upravljanje bazama podataka MySQL kreiran je 1995. godine čime je stvorena alternativa za sustave baza podataka velikih kompanija poput Microsofta i Oraclea. Također, 1998. godine se pojavljuje izraz NoSQL koji se odnosi na baze podataka koje nisu isključivo relacijske i ne koriste isključivo SQL kako bi spremale i vraćale podatke. NoSQL baze podataka su korisne za nestrukturirane podatke i svoj rast su postigle tijekom 2000.-ih godina, te je njihovom upotrebom omogućena brža obrada većih i raznolikijih skupova podataka.

2.2 Sustav za upravljanje bazom podataka

Sustav za upravljanje bazom podataka (SUBP) je softver koji je dizajniran za upravljanje i organiziranje podataka na strukturiran način. Korisnicima omogućuje stvaranje, izmjenu i postavljanje upita na bazi podataka, kao i upravljanje sigurnošću i kontrolama pristupa na toj bazi podataka. Neki od najčešće korištenih sustava za upravljanje bazama podataka su MySQL, Oracle, PostgreSQL, Microsoft SQL Server i MongoDB.

SUBP obavlja sljedeće funkcije:

- Administrativni poslovi - Podržani su tipični zadaci administracije baze podataka, uključujući upravljanje promjenama, praćenje performansi, sigurnost i sigurnosno kopiranje te oporavak. Većina sustava za upravljanje bazama podataka također je odgovorna za automatizirano vraćanje transakcija i ponovno pokretanje sustava, kao i za bilježenje i reviziju aktivnosti u bazama podataka i aplikacijama koje im pristupaju.
- Skladištenje - SUBP osigurava učinkovito pohranjivanje i dohvaćanje podataka osiguravajući da su podaci pohranjeni u tablicama, redovima i stupcima.

- Kontrola istodobnosti - U okruženjima u kojima više korisnika istovremeno pristupa i uređuje bazu podataka, SUBP jamči kontrolirano izvršenje transakcije kako bi se spriječila nedosljednost podataka.
- Centralizirani pogled - SUBP pruža centralizirani prikaz podataka kojima više korisnika može pristupiti s više lokacija na kontrolirani način. Krajnji korisnici i softverski programi ne moraju razumjeti gdje su podaci fizički smješteni ili na kojoj vrsti medija za pohranu se nalaze jer SUBP obrađuje sve zahtjeve.
- Manipulacija podacima - SUBP osigurava integritet i dosljednost podataka dopuštajući korisnicima umetanje, ažuriranje, brisanje i mijenjanje podataka unutar baze podataka.
- Neovisnost podataka - SUBP nudi i logičku i fizičku neovisnost podataka kako bi zaštitio korisnike i aplikacije od potrebe da znaju gdje su podaci pohranjeni. Sve dok programi koriste sučelje za programiranje aplikacija (API, engl. *the application programming interface*), za bazu podataka koju pruža SUBP, programeri neće morati mijenjati programe samo zato što su napravljene promjene u bazi podataka.
- Sigurnosno kopiranje i oporavak - Olakšana je mogućnost sigurnosnog kopiranja i oporavka stvaranjem sigurnosnih kopija tako da se podaci mogu vratiti u dosljedno stanje. To štiti od gubitka podataka zbog kvarova hardvera, softverskih grešaka ili drugih nepredviđenih događaja.

Postoji nekoliko prednosti korištenja SUBPa, neke od njih su sljedeće:

- Organizacija podataka - SUBP pruža alate za organiziranje podataka na strukturiran i logičan način, što može olakšati pretraživanje, sortiranje i dohvaćanje podataka.
- Integritet podataka - Provođenje pravila i postavljanje ograničenja na podatke kako bi se osiguralo da su podaci točni i dosljedni.
- Sigurnost podataka - Dostupni su alati za kontrolu pristupa podacima i njihovu zaštitu od neovlaštenog pristupa.
- Skalabilnost podataka - Podržano je pohranjivanje i upravljanje velikim količinama podataka, a prema potrebi količina se može povećati ili smanjiti.
- Interoperabilnost podataka - SUBP može podržati integraciju podataka iz raznih izvora i može im pristupiti više korisnika ili aplikacijskih programa.

Također, treba uzeti u obzir i potencijalne nedostatke korištenja SUBPa. Neki od nedostataka su:

- Trošak - Postavljanje i održavanje SUBPa može biti skupo, pogotovo za velike ili složene sustave.

- Složenost - Postavljanje i upravljanje SUBPom može biti složeno i može zahtijevati specijalizirano znanje ili stručnost.
- Ovisnost - Aplikacije koje koriste SUBP mogu biti previše ovisne o SUBPu, što može prouzročiti ograničenje njihove fleksibilnosti i prenosivosti.
- Izvedba - SUBP može imati opterećenje koje može utjecati na izvedbu aplikacije, posebno za aplikacije s velikom količinom transakcija ili složenih upita.

3 | Geografski informacijski sustav

Geografski informacijski sustav (GIS) je sustav za spremanje, upravljanje, analizu i vizualiziranje geografskih podataka. GIS povezuje podatke s kartom, integrirajući podatke o lokaciji, koja govori gdje se stvari nalaze, sa svim vrstama opisnih informacija, koje opisuju kakve stvari su tamo. To pruža temelj za mapiranje i analizu koji se koristi u znanosti i gotovo svakoj industriji. Četiri glavna podsustava GISa su:

- Podsustav za unos podataka - Omogućuje prikupljanje i obradu geoprostornih podataka iz tiskanih karata, fotografija iz zraka, izvješća ili istraživačkih dokumenata.
- Podsustav za pohranjivanje i dohvaćanje podataka - Organizira geoprostorne i atributne podatke u obliku koji omogućuje korisniku da ih brzo dohvati za naknadnu analizu. Također omogućava brza i točna ažuriranja i ispravke u geoprostornoj bazi podataka.
- Podsustav za rukovanje i analizu podataka - Omogućuje korisniku da definiira i izvrši geoprostorne i atributne procedure za generiranje željenih informacija te promjenu oblika podataka putem korisnički definiranih pravila. Ovaj se podsustav obično smatra srcem GISa.
- Podsustav za izvoz podataka - Dopušta prikaz cijele, ili dijela, baze podataka, kao i manipuliranje podacima i izvoz geoprostornih modela u obliku tablice ili karte. Stvaranje ovakvih kartografskih prikaza uključuje ono što se naziva digitalna kartografija.

Operativni GIS također ima niz komponenti koje se kombiniraju kako bi sustav funkcionirao i te su komponente ključne za uspješan GIS. Spomenute komponente su sljedeće:

- Hardver - Predstavlja računalo na kojem GIS radi. Danas GIS radi na širokom rasponu hardvera, od centraliziranih računala poslužitelja do stolnih računala koja se koriste na samostalan ili umreženi način.
- Softver - GIS softver pruža funkcije i alate potrebne za pohranu, analizu i prikaz geografskih informacija.
- Podaci - Podaci su vjerojatno najvažnija komponenta GISa. Geografski podaci mogu se prikupiti interno, sastaviti prema prilagođenim specifikacijama

i zahtjevima ili povremeno kupiti od komercijalnog dobavljača podataka. Integracija geoprostornih podataka i tabličnih podataka pohranjenih u SUBPu je ključna funkcionalnost koju pruža GIS.

- Ljudi - Komponenta koja obuhvaća sve ljude osposobljene za upravljanje sustavom i njegovom primjenom na stvarne probleme.
- Metode - Opisuju plan implementacije i poslovna pravila, te modele i prakse jedinstvene za svaku organizaciju.

3.1 Povijest geografskih informacijskih sustava

Korijeni geografskih informacijskih sustava sežu stotinama godina unatrag i pronalazimo ih na područjima kartografije i mapiranja. Rane karte koristile su se za istraživanje, strategiju i planiranje.

Jedan od prvih primjera korištenja geoprostorne analize došao je iz područja epidemiologije 1832. godine kada je francuski geograf i kartograf, Charles Picquet, izradio kartu koja ocrta okruge u Parizu, kako bi pružio vizualni prikaz broja prijavljenih smrti od kolere na svakih 1000 stanovnika.

Godine 1854., liječnik i psiholog, John Snow, uspio je odrediti izvor širenja zaraze kolerom u Londonu koristeći geoprostornu analizu. Do tada se vjerovalo da se kolera širi zrakom, što je on uspio opovrgnuti. Mapirao je sve izvore širenja zaraze, ceste, granice posjeda i vodene pumpe. Nakon što su sve te točke bile označene, uspio je identificirati uzorak koji je dokazao da se bolest ne širi zrakom već vodom, tj. putem zaražene pumpe za vodu. Ovo otkriće je označilo početak geoprostorne analize, ali i početak novog znanstvenog područja, epidemiologije.

Tijekom sljedećih sto godina, nije bilo velikih pomaka u razvoju GISa. Tijekom 1960-ih, računala su značajno napredovala u tehnologiji, brzini i dizajnu. Tada se počinju razvijati rani koncepti računalne geografije. Prvi pravi operativni GIS, razvio je Roger Tomlinson 1963. godine u Ottawi, u Kanadi, pod nazivom *Canada Geographic Information System* (CGIS). CGIS se koristio za spremanje, analizu i upravljanje podacima prikupljenima za Kanadski Inventar Zemljišta (engl. *Canada Land Inventory*) kako bi se odredio potencijal zemljišta mapiranjem informacija o tlu, poljoprivredi, divljim životinjama, pticama i šumama. CGIS je napravio veliki iskorak na ovom području jer je pružao mogućnosti za pohranu podataka, preklapanje slojeva, mjerenje i digitalizaciju. Kao rezultat toga, Tomlinson je postao poznat kao otac GISa.

Arhitekt Howard Fisher je 1965. godine osnovao *Harvard Laboratory for Computer Graphics and Spatial Analysis*, gdje su napravljeni neki od prvih GISova, poput ODYSSEY. Potaknuti idejama iz Harvard Laboratorija i knjige *Design with Nature*, koja je uvela slojevitou metodu za slaganje informacija i postala osnova za rad sa slojevima na kartama, Jack i Laura Dangermond osnovali su *Environmental*

Systems Research Institute, poznatiji kao Esri. Esri je danas najveća GIS softverska kompanija na svijetu koja je igrala važnu ulogu u povijesti GISa.

Tijekom sljedećih dvadeset godina dolazi do ubrzanog razvoja GISa. Kako računala postaju sve bolja, broj GIS softvera se povećava. Također, lansiranjem novih GPS¹ satelita dobiveno je mnogo novih informacije koje su pomogle u razvoju.

Tisuće organizacija u gotovo svim područjima koriste GIS za analizu, razmjenu informacija ili rješavanje kompleksnih problema zbog čega će se razvoj GISa u budućnosti nastaviti.

3.2 Primjena geografskih informacijskih sustava

GIS se može koristiti za rješavanje različitih problema i dobivanje niza informacija. Najčešće primjene GISa su sljedeće:

- Odabir odgovarajuće lokacije - GIS omogućava donošenje boljih poslovnih odluka ili širenje usluga. Moguće je saznati koje su lokacije s najboljim kupcima, što se nalazi u blizini lokacije, broj stanovnika ili lokalnu demografiju kako bi se donijela što bolja odluka.
- Pronalazak najbolje rute - GIS može odrediti najučinkovitije rute i logističke scenarije, uzimajući u obzir sve smetnje na koje je moguće naići na putu. Može koordinirati dnevne rute za veliki broj dostavnih vozila ili upravljati globalnim lancem opskrbe.
- Praćenje informacija - Moguće je steći bolje razumijevanje svime čime upravljamo što znači brže rješavanje problema i planiranje preventivnog održavanja. Na primjer. GIS može pomoći brinuti o cestama, vodovodnim cijevima ili uličnoj rasvjeti.
- Planiranje budućnosti - GIS omogućava predviđanje i donošenje informiranih odluka koje će koristiti i ljudima i planetu modeliranjem mogućih izazova. Primjerice, GIS analiza može odrediti gdje bi se trebale dodati zelene površine kako bi se smanjile ekstremne vrućine.
- Reagirane na hitne slučajeve - Od velike je važnosti gdje i na koji način usmjeriti potrebna sredstva prilikom hitnih slučajeva. GIS se uvelike koristi prije i nakon hitnih slučajeva, poput poplava, požara, uragana ili potresa, za planiranje oporavka.

¹Globalni položajni sustav (GPS, engl. *Global Positioning System*) je satelitski radionavigacijski sustav za određivanje položaja na Zemlji koji korisniku omogućuje određivanje svih triju koordinata njegova trenutnog položaja u jedinstvenom svjetskom koordinatnom sustavu.

4 | Geoprostorne baze podataka

Kao što je rečeno u prethodnim poglavljima, podaci se spremaju i njima se upravlja preko sustava za upravljanje bazom podataka, tj. SUBPom. Tradicionalna uloga SUBPa bila je praćenje podataka o skladištima i računovodstvu, poput informacija o zaposlenicima, dostavama, plaćama, kupcima i proizvodima. Skup upita koji se mogu provesti nad ovakvim bazama podataka je ograničen i primjerice, ovakva baza podataka ne može provjeriti upit: „Navedi sve kupce u krugu od 2 km od trgovine“. Iz ovog razloga, pojavila se potreba za sustavom koji može raditi s geoprostornim podacima.

Sustav za upravljanje geoprostornim bazama podataka (SUGBP, engl. *spatial databases management system - SDBMS*) je oblik sustava za upravljanje bazama podataka koji ima mogućnosti rada s geoprostornim podacima. Ovaj sustav sadrže komponente s pomoću kojih upravljaju geometrijskim podacima, projekcijama i strukturama geoprostornog indeksiranja kako bi se omogućili geoprostorni upiti i operacije. SUGBP proširuje funkcionalnosti standardnog SQLa dodavanjem geoprostornih funkcija i naredbi za rad s geoprostornim podacima.

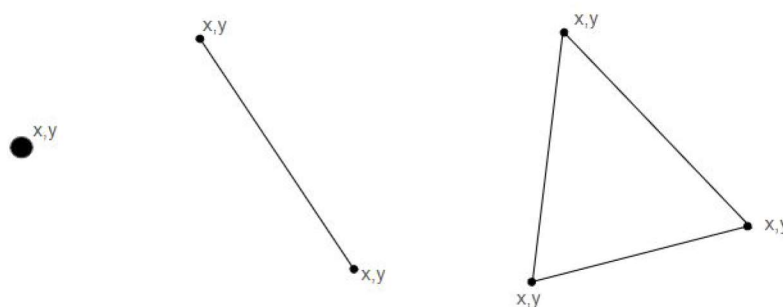
4.1 Geoprostorni podaci

Geoprostorni podaci su bilo koji podaci koji se odnose na određenu lokaciju na Zemlji ili neko geografsko područje. Ovi podaci prikazuju se s pomoću koordinata i spremaju se u tablicu kao atributi. Komponente geoprostornih podataka su geometrija, topologija i atributni podaci. Geometrija se odnosi na koordinate koje definiraju oblik objekta, a najčešće se za to koriste geografska dužina i geografska širina ili x i y u Kartezijevom koordinatnom sustavu. Topologija opisuje odnose između geoprostornih objekata koji se mogu odnositi na povezanost, preklapanje, susjedstvo i slično. Atributni podaci su dodatni podaci koji opisuju karakteristike objekata. Geoprostorni podaci mogu biti prikazani u vektorskom ili raster formatu.

4.1.1 Vektorski podaci

Vektorski podaci predstavljaju mjesto na Zemljinoj površini i dodjeljuju attribute tom mjestu. Vektori se sastoje od diskretnih geometrijskih lokacija koji predstavljaju vrhove koji definiraju oblik objekta u prostoru. Vrh opisuje položaj u prostoru

pomoću x , y i z koordinata. Vrijednosti x i y koordinate ovise o referentnom koordinatnom sustavu koji se koristi, ali najčešće su u upotrebi geometrijska dužina i geometrijska širina. Kada se geometrija objekta sastoji samo od jednog vrha, ona se naziva točka. Kada se sastoji od dva ili više vrha, koji nisu jednaki, kreira se linija. Tamo gdje su prisutna tri ili više vrha, a posljednji vrh je jednak početnom, formira se poligon. Točka, linija i poligon mogu se vidjeti na Slici 4.1.



Slika 4.1: Vektorski prikaz točke, linije i poligona.

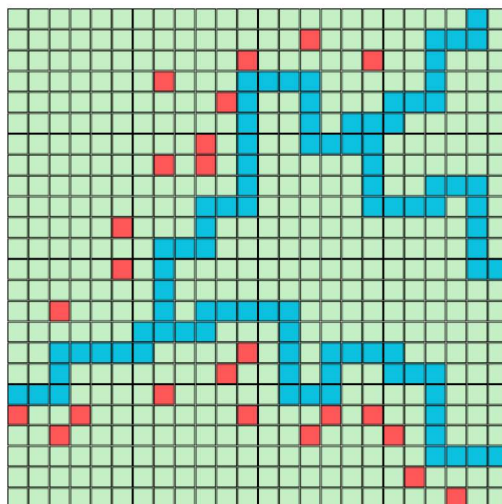
Korištenje točke, često ovisi o mjerilu koje se koristi. Primjerice, ako imamo kartu Hrvatske, točke mogu prikazivati gradove, ali ako se približimo nekom gradu više nema smisla da točka prikazuje taj grad. U tom slučaju, točka može prikazivati škole i vrtiće unutar grada. Linije ili polilinije (dvije ili više povezanih linija) prikazuju geometrije linearnih značajki poput cesta, željeznica ili rijeka. Polilinije ponekad trebaju pratiti određena pravila, primjerice, polilinije koje predstavljaju ceste trebaju biti povezane na raskrižjima. Atributi koji bi mogli opisivati poliliniju ceste su broj traka, je li cesta jednosmjerna ili dvosmjerna ili radi li se o autocesti. Poligoni se koriste za zatvorena područja poput otoka, granica država ili građevina. Poligoni često imaju dijeljenu geometriju, što znači da su im granice zajedničke sa susjednim poligonom. Korištenje vektorskih podataka ljudima dolazi prirodno jer je njihov prikaz vizualno vrlo jednostavan. Međutim, pošto se vektorski podaci spremaju kao niz točaka, njihova pohrana i obrada je kompliciranija, jer se svaka točka pohranjuje posebno.

4.1.2 Raster podaci

Raster podaci su predstavljeni kao matrica piksela raspoređenih u mrežu, tj. u retke i stupce, gdje svaki piksel predstavlja neku geografsku lokaciju. Pomoću ovakve strukture podataka prikazuju se sve digitalne fotografije. Geoprostorni raster razlikuje se od digitalne fotografije samo po tome što on sadrži informacije koje povezuju podatke s određenom lokacijom. To uključuje opseg rastera i veličinu ćelije, broj redaka i stupaca te njegov koordinatni referentni sustav (CRS, engl. *coordinate reference system*). Kao što je spomenuto, točka, linija i poligon dobro prikazuju ceste, građevine ili drveće, ali primjerice gustoću ili vrste drveća prikazuju se rasterom. Rasterski podaci mogu se koristiti i kao pozadina vektorskih slojeva kako bi se vektorskim informacijama dalo više značenja. Ljudi vrlo

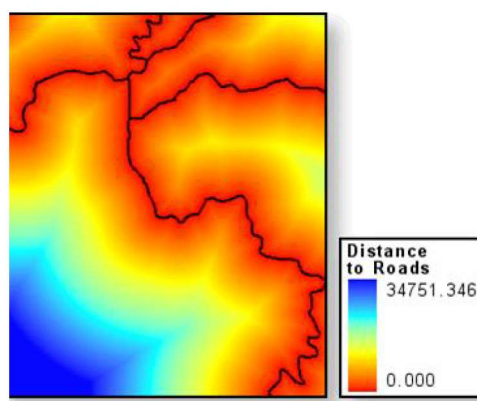
dobro tumače slike pa korištenje slike iza vektorskih slojeva daje karte s puno više detalja i informacija. Raster podaci mogu se podijeliti u dva tipa: diskretni ili kontinuirani.

Diskretne vrijednosti su različite i često su kategoričke. Pod diskretne vrijednosti ubrajaju se primjerice tipovi zemljišnih pokrova ili kategorije korištenja zemljišta. Diskretni podaci obično su predstavljeni cijelim brojevima, pri čemu svaka vrijednost predstavlja drugu kategoriju. Primjer diskretnih raster podataka može se vidjeti na Slici 4.2.



Slika 4.2: Prikaz diskretnih raster podataka gdje plava boja označava rijeku, zelena šumu, a crvena vikendicu.

Kontinuirane vrijednosti su numeričke i predstavljaju mjerenja koja mogu poprimiti bilo koju vrijednost unutar nekog raspona. Primjer kontinuiranih podataka su podaci o visini ili temperaturi. Kontinuirani raster podaci mogu se vidjeti na Slici 4.3.



Slika 4.3: Primjer kontinuiranih raster podataka koji prikazuju udaljenost od ceste, izvor: [5].

4.1.3 Usporedba vektorskih i rasterskih podataka

Usporedba vektorskih i rasterskih podataka s obzirom na njihove prednosti prikazana je u Tablici 4.1.

Vektorski podaci	Rasterski podaci
Jednostavna geometrija što znači i manje podataka	Rad i s kontinuiranim i s diskretnim podacima
Jednostavni za uređivanje	Dobri za matematičko modeliranje i kvantitativnu analizu
Logička struktura podataka	Dopuštaju brzu i jednostavnu analizu podataka
Podržava velik broj tehnika za analizu	Geografska lokacija svake ćelije ovisi o njezinom mjestu u mreži

Tablica 4.1: Prednosti vektorskih i rasterskih podataka.

Nedostaci vektorskih i rasterskih podataka prikazani su u Tablici 4.2.

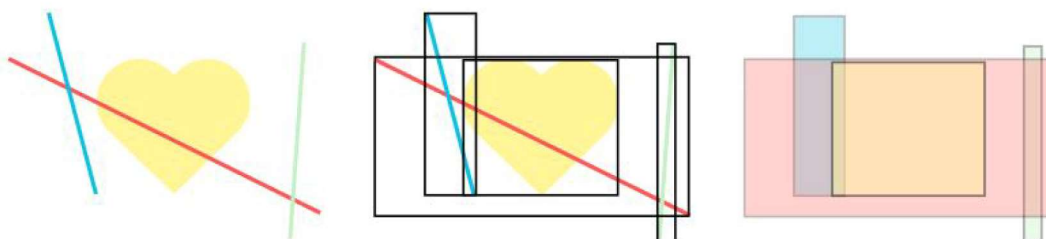
Vektorski podaci	Rasterski podaci
Nisu prikladni za prikaz kontinuiranih podataka	Manje efektivni za prikaz linearnih značajki
Učinkovita analiza podataka zahtjeva pretvaranje u topološke strukture	Dobivene karte općenito ne zadovoljavaju kartografske potrebe
Može zahtijevati puno ručnog uređivanja	Rezolucija je određena veličinom ćelija
Algoritmi za analitičke funkcije su složeni	Budući da je većina ulaznih podataka u vektorskom obliku, podaci moraju proći konverziju vektora u raster

Tablica 4.2: Nedostaci vektorskih i rasterskih podataka.

4.2 Geoprostorno indeksiranje

Geoprostorna baza podataka koristi geoprostorne indekse za optimizaciju geoprostornih upita, tj. za brži pristup nekom objektu u prostoru. Bez indeksa, za svaki upit bi se čitao svaki zapis u bazi zbog čega bi cijeli proces bio znatno sporiji.

Standardni indeksi baze podataka stvaraju hijerarhijsko stablo na temelju vrijednosti stupca koji se indeksira. Geoprostorni indeksi su malo drugačiji te oni ne mogu sami indeksirati geometrijske značajke pa umjesto toga indeksiraju granične okvire značajki.



Slika 4.4: Prikaz načina rada geoprostornih indeksa.

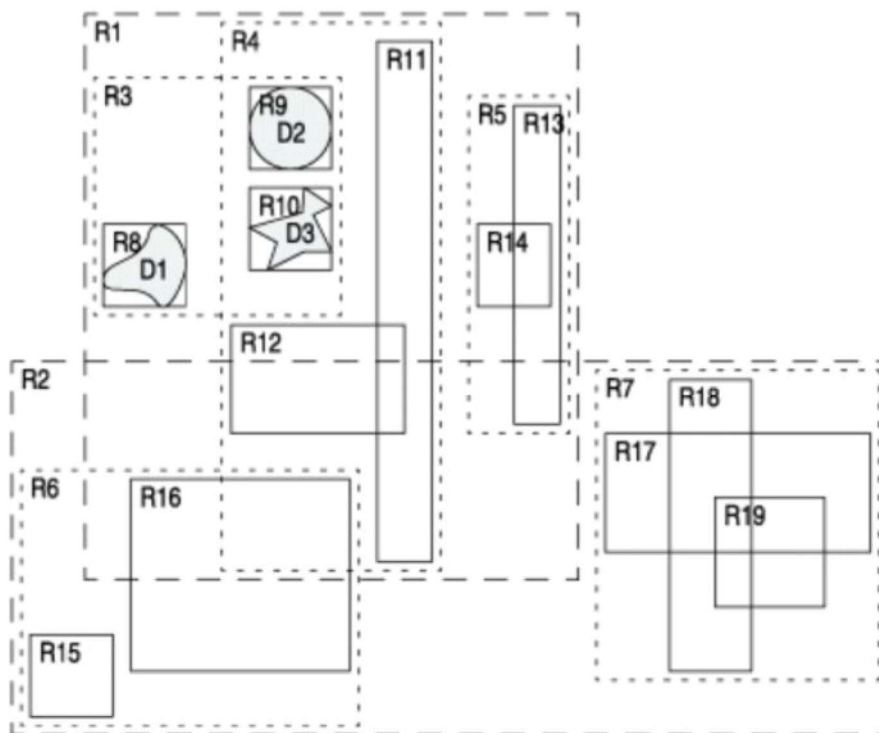
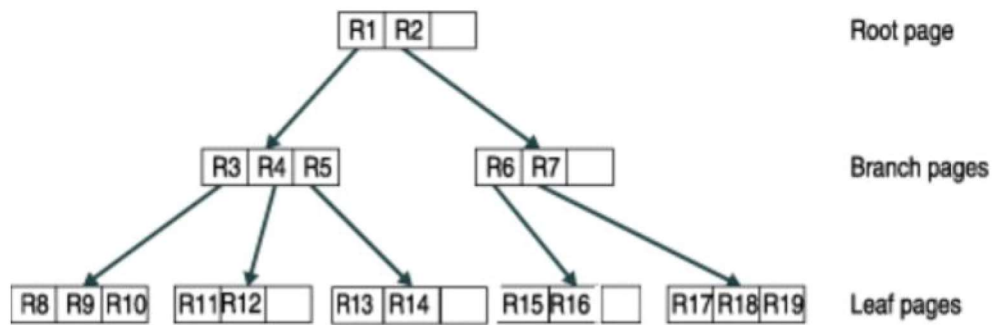
Rad geoprostornih indeksa bit će opisan s pomoću Slike 4.4. Broj linija koji presijecaju žuto srce na Slici 4.4 je jedan, tj. crvena linija. Ali ako oko svakog lika nacrtamo najmanji mogući pravokutnih koji sadrži taj lik, broj pravokutnika koji presijecaju žuti pravokutnik je dva, tj. crveni i plavi pravokutnik. Način na koji baza podataka učinkovito odredi koje linije sijeku žuto srce je da prvo odredi koji okviri ili pravokutnici sijeku žuti okvir što se određuje pomoću indeksa i vrlo je brzo. Na kraju se provjerava broj linija koje sijeku žuto srce samo za one značajke koje je vratio prvi upit.

Za velike tablice, ovakav način dvostrukog prolaza u kojem se prvo procjenjuje približni indeks, a zatim provodi točan upit, može značajno smanjiti količinu izračuna potrebnih za odgovor na upit.

PostGIS za geoprostorno indeksiranje koristi strukturu R-stabla (engl. *R-tree*). R-stablo dijeli podatke u pravokutnike koji se onda dijele na još manje pravokutnike. Ovu strukturu predložio je Antonin Guttman 1984. godine, a koristi se za pohranjivanje geoprostornih objekata poput određene lokacije ili brzo pronalaženje odgovora na upite kao što je: „Pronađi sve bolnice u krugu od 2 km od moje trenutne lokacije“.

Glavna ideja R-stabla je grupirati objekte koji su blizu i predstaviti ih njihovim minimalnim opisanim pravokutnikom na sljedećoj višoj razini stabla. Budući da svi objekti leže unutar opisanog pravokutnika, pravokutnik koji ne siječe opisani pravokutnik također ne može sjeći nijedan od sadržanih objekata. Ako gledamo listove stabla, svaki pravokutnik opisuje jedan objekt, a na višim razinama grupiranje uključuje sve veći broj objekata. Prikaz R-stabla može se vidjeti na Slici 4.5.

Kao i kod većine stabala, algoritmi pretraživanja su prilično jednostavni, a ključna ideja je koristiti granične okvire kako bi se odlučilo hoćemo li pretraživati unutar podstabla ili ne. Na taj način se većina čvorova nikada ne čita tijekom pretraživanja. R-stabla su prikladna za velike skupove podataka i velike baze podataka.



Slika 4.5: Primjer jednog R-stabla, izvor: [14].

Svojstva R-stabla su sljedeća:

- sastoji se od jednog korijena, unutarnjih čvorova i listova,
- korijen sadrži pokazivač na najveće područje u domeni,
- roditeljski čvorovi sadrže pokazivače na svoju djecu gdje se područje od djece u potpunosti preklapa s područjima roditelja,
- listovi sadrže podatke o najmanjim graničnim pravokutnicima objekata.

Primjerice, ako nas iz Slike 4.5 zanima sijeku li se pravokutnici R15 i R16, moramo pretražiti stablo preko R2 i R6. Pravokutnih R6 sadrži oba pravokutnika pa usporedbom njihovih geometrija zaključujemo da se pravokutnici R15 i R16 ne sijeku. Bez korištenja geoprostornih indeksa, pretraga za ovakav primjer trajala bi puno duže jer bi svaki pravokutnik prolazili posebno.

Iako R-stablo ima mnoge prednosti, neki od njegovih nedostataka, zbog kojih su se pojavile varijante poput R+-stabla ili R*-stabla, su sljedeći:

- Izvršavanje upita o lokaciji točke može dovesti do pretraživanja nekoliko puteva od korijena do lista zbog čega se performanse pogoršavaju.
- Nekoliko velikih pravokutnika može značajno povećati stupanj preklapanja, što dovodi do smanjenja performansi tijekom izvršavanja upita zbog praznog prostora.
- Nisu pogodna za rad s podacima koji imaju visok stupanj preklapanja jer to rezultira stvaranjem velikog broja preklapajućih čvorova u stablu.

5 | Alati

5.1 Docker

Docker je platforma za razvoj, isporuku i pokretanje aplikacija. To je alat koji se koristi za automatizaciju postavljanja aplikacije u spremnike, tj. kontejnere kako bi aplikacije mogle raditi u različitim izoliranim okruženjima. Izolacija i sigurnost omogućuju pokretanje više kontejnera istovremeno na određenom računalu. Kontejneri su lagani i sadrže sve što je potrebno za pokretanje aplikacije, tako da se korisnik ne mora oslanjati na ono što je instalirano na glavnom računalu. Također, kontejneri se mogu dijeliti čime se osigurava da svi s kojima je kontejner podijeljen dobivaju kontejner koji radi na isti način.

Docker koristi arhitekturu klijent-poslužitelj. Docker klijent razgovara s Docker daemonom¹, koji obavlja posao izgradnje, pokretanja i distribucije Docker kontejnera. Docker klijent i daemon mogu se izvoditi na istom sustavu ili se klijent može povezati s udaljenim daemonom.

Osim kontejnera, važni objekti su i slike, mreže, volumeni i dodaci. U okviru ovog rada korištene su samo slike pa će one biti detaljnije objašnjene, a o ostalim objektima može se više pročitati na [7]. Slika je predložak za čitanje s uputama za stvaranje Docker kontejnera. Često se slika temelji na drugoj slici, uz neke dodatne prilagodbe.

5.2 PostgreSQL

PostgreSQL je besplatni, open-source objektno-relacijski sustav za upravljanje bazama podataka koji koristi i proširuje SQL jezik. Razvoj PostgreSQLa započeo je 1986. godine kao dio POSTGRES projekta na Sveučilištu u Kaliforniji, u Berkeleyju, pod vodstvom profesora Michaela Stonebrakera. Godine 1996. POSTGRES mijenja ime u PostgreSQL kako bi se označilo da podržava SQL.

Ono zbog čega PostgreSQL postaje sve više popularan je činjenica da ne pripada jednoj tvrtki, već grupi *The PostgreSQL Global Development Group* koja se sastoji od volontera i tvrtki koje zajednički rade na poboljšanju sustava. Post-

¹Daemon je program čija je svrha raditi nešto u pozadini, bio korisnik prijavljen na računalo ili ne, dakle glavna svrha nije interakcija s korisnikom, nego obavljanje nekog zadatka.

greSQL podržava veliki dio SQL standarda i pruža puno modernih mogućnosti poput složenih upita, stranih ključeva, okidača (engl. *triggers*) ili prikaza koji se mogu ažurirati. Također, dopušta proširivanje od strane korisnika dodavanjem novih tipova podataka, funkcija, operatora, metoda indeksa ili proceduralnih jezika. PostgreSQL je izuzetno popularan u radu s geoprostornim podacima putem svog proširenja PostGISa.

5.3 PostGIS

PostGIS je proširenje PostgreSQLa, te u PostgreSQL dodaje podršku za geoprostorne tipove podataka, geoprostorne indekse i geoprostorne funkcije. PostGIS omogućava PostgreSQL serveru da se koristi kao *back-end* geoprostorna baza podataka za geografske informacijske sustave i kartografske aplikacije na isti način kao i Microsoft SQL Server Spatial i Oracle Spatial proširenja baza podataka. PostGIS implementira geometrijske, geografske, topološke i rasterske geoprostorne tipove podataka, tj. geometriju, geografiju, topologiju i raster. Rasterski tipovi podataka objašnjeni su detaljnije u poglavlju 4.1.2, a ostali tipovi bit će objašnjeni u nastavku.

Značajke PostGISa su sljedeće:

- Pohrana geoprostornih podataka - Moguća je pohrana različitih geoprostornih podataka, kao što su točke, linije, poligoni i multi-geometrije u 2D ili 3D obliku.
- Geoprostorno indeksiranje - Brzo pretraživanje i dohvaćanje geoprostornih podataka na temelju njihove lokacije.
- Geoprostorne funkcije - Dostupan je širok raspon geoprostornih funkcije koje omogućavaju filtriranje i analizu geoprostornih podataka, mjerenje udaljenosti i područja ili pronalazak geometrija koje se presijecaju.
- Obrada geometrije - Moguće je korištenje alata za obradu i manipuliranje geometrijskim podacima, kao što su pojednostavljivanje, konverzija ili generalizacija.
- Geokodiranje i obrnuto geokodiranje - Dostupne su funkcije za pretvaranje adresa u koordinate ili koordinata u adrese.
- Integracija - Pristup i rad s PostGISom koristeći alate trećih strana kao što su QGIS, GeoServer, MapServer ili ArcGIS.

Instalacija PostGISa moguća je preko službene PostGIS web stranice ili putem platforme Docker na način opisan u nastavku.

Prvo se instalira Docker putem službene web stranice te se na željenom mjestu na

računalu kreira posebna mapa. U mapi se kreira datoteka `docker-compose.yml`² s kodom sličnim Programskom kodu 5.3.1.

```
1 services:
2   postgis:
3     image: pgrouting/pgrouting
4     restart: always
5     environment:
6       - POSTGRES_DB=gis
7       - POSTGRES_USER=gis
8       - POSTGRES_PASSWORD=password
9     ports:
10      - 5433:5432
11     volumes:
12      - ./data/postgis:/var/lib/postgresql/data
```

Programski kod 5.3.1: Konfiguracijski kod za definiranje baze podataka i određivanje slike.

Kod 5.3.1 je konfiguracijski kod koji služi za određivanje slike koja će se koristiti (`pgrouting/pgrouting`, koja se temelji na slici `postgis/postgis` i obuhvaća sve njezine funkcionalnosti), definiranje baze (`POSTGRES_DB`), korisnika (`POSTGRES_USER`), lozinke (`POSTGRES_PASSWORD`), priključaka (`5433:5432`) i volumena. Nakon kreiranja ovakve datoteke, otvorimo Naredbeni redak i navigiramo do mape u kojoj se nalazi ta datoteka te koristimo naredbu `docker compose up` čime se izgrađuju i pokreću kontejneri. Nakon ovoga, moguće je na računalu koristiti PostGIS s podacima definiranim u Programskom kodu 5.3.1 za logiranje u bazu podataka.

5.3.1 Povijest PostGISa

U svibnju 2001. godine, objavljena je prva verzija PostGISa koja je imala objekte, indekse i mali broj funkcija. Rezultat je bila baza podataka pogodna za spremanje i vraćanje podataka, ali ne i analizu. Kako se broj funkcija povećavao, postala je jasna potreba za organizacijom. Specifikacija *Simple Features for SQL* (SFSQL) je pružila smjernice za imenovanje funkcija, a Mapserver je postao prva vanjska aplikacija koja je koristila PostGIS i koja je omogućavala vizualizaciju podataka iz baze podataka.

Tijekom sljedećih nekoliko godina, broj PostGIS funkcija je rastao, ali je njegov potencijal ostao ograničen jer je mnoge funkcije koje bi bile od velike koristi bilo vrlo teško kodirati. Tada se pojavio drugi projekt, *Geometry Engine, Open Source* (GEOS), koji je bio biblioteka koja pruža potrebne algoritme za implementaciju SFSQL specifikacija.

Prvobitna struktura PostGIS geometrije baze podataka kreirana je bez obra-

²YAML je jezik razumljiv od strane ljudi koji se često koristi za pisanje konfiguracijskih datoteka.

ćanja previše pažnje na količinu prostora koju zauzima. Struktura je dodijelila prostor za 3-dimenzionalne podatke (3D), čak i onda kada bi podaci bili 2-dimenzionalni (2D). Za male 2D objekte, poput 2D točke, spremanje kao 3D objekt zauzelo bi i do 200% memorije više nego što je bilo potrebno. Kako bi se PostGIS ubrzao, smanjene su strukture podataka korištenjem novog eksperimenta *light-weight geometry*, koji je koristio pojedinačne bitove za *boolove* te se smanjila veličina zahtijevanih dimenzija.

Nakon izdavanja prvih verzija, nastavljen je rad na poboljšanju PostGIS performansi, a danas se i dalje radi na njegovom razvoju.

5.3.2 Geometrija u PostGISu

Geometrija je osnovni tip geoprostornih podataka koji se koristi za prikaz značajki ili objekata u planarnom koordinatnom sustavu³. Često se za prikaz koriste geografska dužina i širina, a zakrivljenost Zemlje se ne uzima u obzir. Koordinate x i y su obavezne, a z i m su opcionalne. Z koordinata najčešće predstavlja visinu, a koordinata m može predstavljati vrijeme ili udaljenost. Ako su z ili m vrijednosti prisutne, moraju se definirati za svaku točku u geometriji. Ako geometrija ima z ili m koordinatu, dimenzija je 3D, a ako ima i z i m, onda je dimenzija 4D. Osnovni geometrijski tipovi su:

1. *Point* - točka,
2. *LineString* - linija,
3. *LinearRing* - prsten,
4. *Polygon* - poligon,

a koleksijski tipovi su:

1. *MultiPoint* - kolekcija točaka,
2. *MultiLineString* - kolekcija linija,
3. *MultyPoligon* - kolekcija poligona,
4. *GeometryCollection* - kolekcija bilo kojih geometrija.

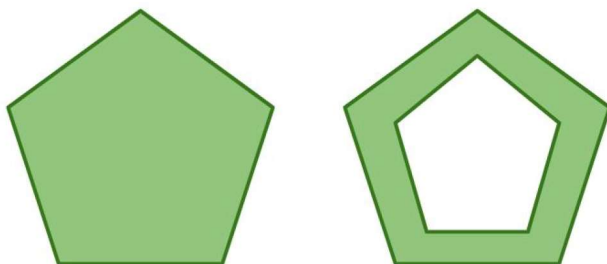
Točka predstavlja jednu lokaciju na Zemlji i predstavljena je uređenim parom koordinata (u dvije, tri ili četiri dimenzije). Točka se koristi za objekte kojima detalji, poput oblika ili veličine, nisu važni.

Linija predstavlja put između lokacija i ima oblik uređenog niza od dvije ili više točaka, tj. od jednog ili više segmenata. Svaki segment je definiran s dvije točke, tako da je završna točka jednog segmenta, početna točka sljedećeg

³Planarni koordinatni sustav je vrsta geoprostornog referentnog sustava koji predstavlja lokacije na Zemlji koristeći Kartezijeve koordinate x i y na ravnoj površini.

segmenta. Kaže se da je linija zatvorena ako počinje i završava u istoj točki, a jednostavna ako se ne križa i ne dodiruje sama sebe. Prsten je linija koja je u isto vrijeme i zatvorena i jednostavna.

Poligon predstavlja neko dvodimenzionalno područje, omeđeno vanjskom granicom i s nula ili više unutarnjih granica. Svaka granica poligona predstavljena je prstenom. Na Slici 5.1 tamno zelenom bojom prikazani su prsteni poligona.



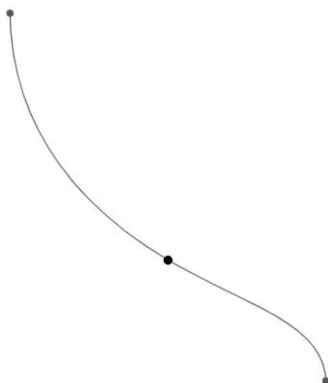
Slika 5.1: Poligoni sa svojim prstenima.

Kolekcijski tipovi grupiraju veći broj jednostavnih geometrija u skupove i korisni su za izravno modeliranje objekata stvarnog svijeta kao geoprostornih objekata.

ISO/IEC 13249-3 standard je proširio dosadašnje tipove u PostGISu i definirao sljedeće krivulje s kružnim lukovima:

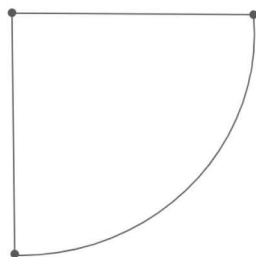
1. *CircularString*,
2. *CompoundCurve*,
3. *CurvePolygon*,
4. *MultiCurve*,
5. *MultiSurface*.

CircularString je osnovni tip krivulje, sličan tipu *LineString* u linearnom svijetu. Jedan segment luka određen je s tri točke: početnom, krajnjom i još jednom točkom na luku što se vidi na Slici 5.2. U nizu segmenata lukova, krajnja točka prethodnog segmenta jednaka je početnoj točki sljedećeg segmenta. *CircularString* mora imati neparan broj točaka veći od 1.



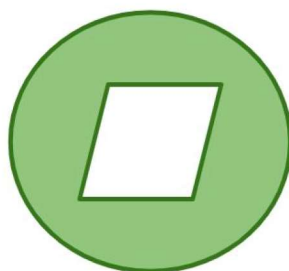
Slika 5.2: Tip krivulje CircularString.

CompoundCurve predstavlja kontinuiranu krivulju koja može sadržavati segmente kružnog luka i linearne segmente kao što se vidi na Slici 5.3. To znači da se krajnja točka svake komponente (osim posljednje) mora podudarati s početnom točkom sljedeće komponente.



Slika 5.3: Tip krivulje CompoundCurve.

CurvePolygon je poput poligona, s vanjskim prstenom i nula ili više unutarnjih prstena. Razlika je u tome što prsten može biti CircularString ili CompoundCurve kao što se vidi na Slici 5.4.



Slika 5.4: Tip krivulje CurvePolygon.

MultiCurve je kolekcija krivulji koja može sadržavati LineStringove, CircularStringove ili CompoundCurve.

MultiSurface je kolekcija površina, koje mogu biti linearni poligoni ili CurvePolygoni.

5.3.3 Geografija u PostGISu

Geografski tip podataka podržava geoprostorne objekte predstavljene geografskim koordinatama, tj. sfernim koordinatama izraženim u kutnim jedinicama (stupnjevima). U sfernim koordinatama točka je određena kutom rotacije od referentnog meridijana (geografska dužina) i kutom od ekvatora (geografska širina). Za razliku od geometrijskog tipa koji se temelji na ravnini, geografski tip temelji se na sfernom modelu. Tada je najkraći put između dvije točke na sferi kružni luk pa se tako geografske funkcije, poput površine, udaljenosti, duljine ili sjecišta, računaju pomoću lukova na sferi. Udaljenost između točaka postaje sve veća kako se približavamo problematičnim područjima poput polova ili međunarodne datumske granice.

Kao i geometrijski tip podataka, geografski podaci povezani su s prostornim referentnim sustavom putem identifikatora prostornog referentnog sustava, tj. SRID (engl. *spatial reference system identifier*)⁴. Može se koristiti bilo koji geodetski prostorni referentni sustav definiran u PostGISu.

Vrste podataka s kojima geografija radi su Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon i GeometryCollection. Geografski tipovi ne podržavaju krivulje.

5.3.4 Topologija u PostGISu

Topologija je generalni koncept u kojem su objekti definirani s obzirom na njihove odnose, a ne geometrije. Umjesto linija, koriste se vrhovi i bridovi, zbog čega topologija podsjeća na osnovne koncepte iz teorije grafova.

Topologija osigurava integritet podataka, što znači da pomaže utvrditi i održavati bolje prikaze značajki u geoprostornoj bazi podataka. Osim toga, koristi se za modeliranje geoprostornih odnosa između objekata koji omogućuju podršku za razne analitičke operacije, kao što je pronalaženje susjednih objekata, rad sa zajedničkim granicama između objekata i navigacija duž povezanih objekata.

Topologija je način na koji točke, linije i poligoni dijele geometriju, te se topologija može koristiti za sljedeće:

- Ograničavanje načina na koji objekti dijele geometriju. Na primjer, susjedni poligoni kao što su parcele imaju zajedničke rubove.
- Definiranje pravila za integritet podataka. Na primjer, ne smije biti razmaka između poligona ili se objekti ne smiju preklapati.
- Podržavanje upita o topološkim odnosima i navigaciji, kao što je identificiranje susjednih objekata i povezanosti.
- Podržavanje alata za uređivanje koji provode topološka ograničenja.

⁴Prostorni referentni sustav je jedinstveni identifikator povezan s određenim koordinatnim sustavom, tolerancijom i rezolucijom.

5.3.5 Funkcije u PostGISu

Funkcije u PostGISu nude širok raspon funkcionalnosti za analizu podataka te ih možemo podijeliti u sljedeće kategorije:

- apstraktne funkcije,
- konstruktorske funkcije,
- pristupne funkcije,
- ulazne funkcije,
- izlazne funkcije,
- funkcije odnosa,
- funkcije mjerenja.

Apstraktne funkcije pomažu pri definiranju tablica koje sadrže geometrijske stupce. Neki od primjera apstraktnih funkcija su:

- `AddGeometryColumn` - dodaje geometrijske stupce u postojeću tablicu,
- `DropGeometryColumn` - briše geometrijski stupac iz tablice,
- `DropGeometryTable` - briše cijelu tablicu i sve njezine reference.

Konstruktorske funkcije se koriste za kreiranje geometrijskih ili geografskih objekata u PostgreSQL bazi. Neke konstruktorske funkcije su sljedeće:

- `ST_MakePoint`, `ST_MakeLine`, `ST_MakePolygon` - kreiraju 2D, 3D ili 4D točku, liniju ili poligon,
- `ST_Point` - kreira točku s x , y i SRID vrijednostima,
- `ST_Polygon` - kreira poligon iz `LineStringa` sa specificiranom SRID vrijednosti.

Pristupne funkcije koriste se za pristup i dohvaćanje informacija o objektima unutar baze podataka.

- `ST_X`, `ST_Y`, `ST_Z` - vraćaju x , y ili z koordinatu točke,
- `ST_StartPoint` - vraća početnu točku od `LineStringa` ili `CircularLineStringa`,
- `ST_EndPoint` - vraća krajnju točku od `LineStringa` ili `CircularLineStringa`,
- `ST_IsClosed` - provjerava preklapaju li se početna i krajnja točka `LineStringa`,
- `ST_IsEmpty` - provjerava je li geometrija prazna,
- `ST_IsSimple` - provjerava presijeca li geometrija samu sebe,

- `ST_NumPoints` - vraća broj točaka na `LineStringu` ili `CircularStringu`.

Ulazne funkcije kreiraju geometrijske objekte iz različitih tekstualnih ili binarnih formata. Neke od njih su sljedeće:

- `ST_GeomFromText` - kreira geometrijski objekt iz tekstualnog formata,
- `ST_GeogFromText` - kreira geografski objekt iz tekstualnog formata,
- `ST_GeomFromGeoJSON` - kreira geometrijski objekt iz GeoJSON formata,
- `ST_GeogFromWKB` - kreira geografsku instancu iz Well-Known Binary⁵ (WKB) geometrijske reprezentacije.

Izlazne funkcije pretvaraju geometrijske objekte u različite tekstualne ili binarne formate. Neke od izlaznih funkcija su sljedeće:

- `ST_AsEWKT` - vraća Well-Known Text⁶ (WKT) reprezentaciju geometrije sa SRID podacima,
- `ST_AsText` - vraća Well-Known Text (WKT) reprezentaciju geometrije ili geografije bez SRID podataka,
- `ST_AsBinary` - vraća Well-Known Binary reprezentaciju geometrije ili geografije bez SRID podataka,
- `ST_AsEWKB` - vraća Extended Well-Known Binary (EWKB) reprezentaciju geometrije sa SRID podacima.

Funkcije odnosa određuju geoprostorne odnose između geometrija. Neke od njih su sljedeće:

- `ST_Contains(geometry A, geometry B)` - vraća TRUE ako geometrija A sadrži geometriju B, tj. ako i samo ako sve točke iz B leže u A,
- `ST_Crosses(geometry A, geometry B)` - vraća TRUE ako geometrije imaju neke zajedničke točke, ali ne sve,
- `ST_Disjoint(geometry A, geometry B)` - vraća TRUE ako A i B nemaju zajedničkih točaka,
- `ST_Intersects(geometry A, geometry B)` - vraća TRUE ako se geometrije presijecaju, tj. imaju jednu zajedničku točku,
- `ST_Equal(geometry A, geometry B)` - vraća TRUE ako su geometrije jednake,
- `ST_DWithin(geometry A, geometry B, float distance)` - vraća TRUE ako se geometrije nalaze unutar dane udaljenosti.

⁵Well-Known Binary je jezik za prikaz i pohranjivanje informacija u obliku pogodnom za računalnu obradu koji nije čitljiv ljudima.

⁶Well-Known Text je jezik za prikaz geometrijskih objekata u tekstualnom formatu.

Funkcije mjerenja računaju udaljenosti, površine i kutove te geometrijske vrijednosti određene mjerenjima. Neke od ovih funkcija su sljedeće:

- `ST_Area` - vraća površinu poligona,
- `ST_Angle` - računa kut između dvije linije,
- `ST_ClosestPoint(geometry/geography A, geometry/geography B)` - vraća 2D točku na A koja je najbliža B,
- `ST_Distance(geometry/geography A, geometry/geography B)` - vraća minimalnu udaljenost između geometrija ili geografija, s razlikom u mjernim jedinicama,
- `ST_Length(geometry/geography A)` - vraća duljinu geometrije ili geografije, s razlikom u mjernim jedinicama,
- `ST_Distance(geometry A, geometry B)` - vraća najdulju 2D liniju između točaka na dvije geometrije.

5.4 QGIS

QGIS je besplatni GIS softver koji služi za pregledavanje, uređivanje i analiziranje geoprostornih podataka poput vektorskih i rasterskih. Također, služi za pravljenje i izvoz grafičkih karata. QGIS omogućuje korisnicima da vizualiziraju svoje podatke pomoću karata i dijagrama i prilagode ih velikim izborom simbola. Mogućnosti geografske analize koje pruža QGIS uključuju geoprostorne upite i geoprocesiranje. Za složeniju geografsku analizu korisnici mogu koristiti dodatke i algoritme. QGIS također olakšava dijeljenje i objavljivanje geoprostornih podataka kao karata, mrežnih usluga ili ispis karata u različitim formatima.

Značajke QGISa su sljedeće:

1. Fleksibilno i snažno upravljanje geoprostornim podacima:
 - izrada geoprostornih podataka putem vizualne i numeričke digitalizacije,
 - georeferenciranje rasterskih i vektorskih podataka.
2. Detaljna kartografija:
 - veliki izbor 2D i 3D opcija renderiranja,
 - detaljna kontrola nad simbologijom, označavanjem, legendama i dodatnim grafičkim elementima za lijepo prikazane karte,
 - izrada i ažuriranje karte s određenim mjerilom, opsegom, stilom i ukrasima,
 - automatsko generiranje više karata.
3. Napredna geoprostorna analiza:
 - snažan okvir za obradu s preko 200 algoritama za obradu,
 - trenutna vizualizacija geoprostornih upita i rezultata procesiranja.
4. Rad s proširenjima:
 - potpuno prilagodljivo korisničko iskustvo, uključujući korisničko sučelje i postavke aplikacije,
 - širok raspon proširenja koji uključuje napredne alate za analizu i crtanje, prikupljanje podataka na terenu ili konverziju datoteka.

QGIS sadrži velik skup alata i funkcionalnosti o kojima se može više pročitati na [13].

6 | Primjena geoprostorne baze podataka

Primjena geoprostorne baze podataka pokazat će se na primjeru pronalaska najbliže zdravstvene ustanove na području grada Osijeka s obzirom na hitnost pacijenta i razinu hitnosti koju zdravstvena ustanova pruža te na primjeru pronalaska svih zdravstvenih ustanova u danom radijusu. Programski kod i korišteni podaci dostupni su na GitHubu na sljedećem linku https://github.com/bardiclucija/spatial_database.

6.1 Podaci

Podaci s informacijama o ustanovama preuzeti su s web stranice Hrvatskog zavoda za javno zdravstvo [vidjeti [9]] i spremljeni su u CSV¹ formatu. Datoteka ima sljedeće stupce: naziv, županija, naselje, ulica, poštanski broj i vrsta te je pomoću Python skripte kreiran novi stupac adresa gdje su spojeni ulica, poštanski broj te naselje. Također, dodana su dva nova stupca dužina i širina u koje se pomoću *geopy* biblioteke dodaju geografska dužina i geografska širina, tj. koordinate, za svaku adresu. Ažurirani CSV podaci na kraju se spremaju u novu CSV datoteku. Sve opisano i način na koji je to napravljeno može se vidjeti u Programskom kodu 6.1.

```
1 import csv
2 from geopy.geocoders import Nominatim
3
4 filename = "facilities.csv"
5 output_filename = "updated_facilities.csv"
6 fields = []
7 rows = []
8
9 with open(filename, 'r', newline='', encoding='utf-8') as csvfile:
10     csvreader = csv.reader(csvfile)
11     fields = next(csvreader)
12     for row in csvreader:
13         rows.append(row)
14
15 fields.append('Adresa')
```

¹CSV (vrijednosti razdvojene zarezom, engl. *Comma-separated values*) je tekstualna datoteka koja koristi zareze kako bi razdvojila vrijednosti i prikazala ih u tabličnom obliku.


```
16 fields.append('Sirina')
17 fields.append('Duzina')
18
19 for row in rows:
20     postanski_broj = row[4]
21     ulica = row[3]
22     naselje = row[2]
23     adresa = f"{ulica} {naselje} {postanski_broj}"
24     row.append(adresa)
25
26 geolocator = Nominatim(user_agent="diplomski")
27
28 for row in rows:
29     adresa = row[6]
30     print(f"Geocoding address: {adresa}")
31     location = geolocator.geocode(adresa, addressdetails=True)
32     if location:
33         row.append(location.latitude)
34         row.append(location.longitude)
35     else:
36         row.append(0)
37         row.append(0)
38
39 with open(output_filename, 'w', newline='', encoding='utf-8') as
    csvfile:
40     csvwriter = csv.writer(csvfile)
41     csvwriter.writerow(fields)
42     csvwriter.writerows(rows)
```

Programski kod 6.1: Priprema podataka.

Vrste zdravstvenih ustanova koje su uzete u obzir su sljedeće: ljekarna, ljekarnička jedinica (koja pripada centralnoj ljekarni), dom zdravlja, ambulanta doma zdravlja (koja pripada centralnom domu zdravlja), poliklinika, ustanova za zdravstvenu skrb, dislocirani odjel specijalističke bolnice, klinički bolnički centar te ustanova za hitnu medicinsku pomoć. Također, u tablici se nalaze informacije o zavodu za javno zdravstvo županije i dislociranoj jedinici zavoda za javno zdravstvo.

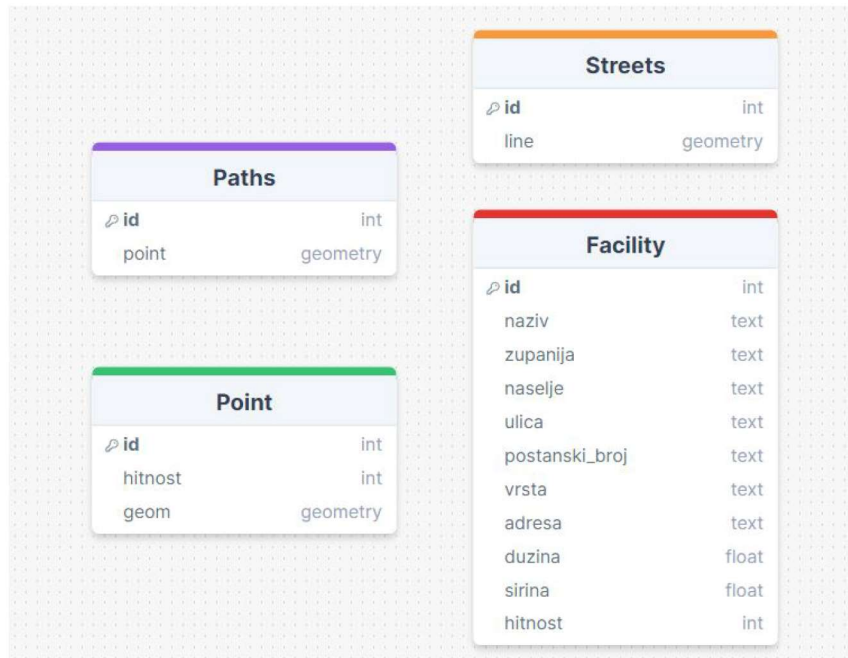
6.2 Baza podataka

Tablice sadržane u bazi podataka koja će se koristiti za rješavanje ovog problema su sljedeće:

- Ustanova (engl. *Facility*) - sprema podatke o zdravstvenim ustanovama,
- Točka (engl. *Point*) - sprema podatke o lokaciji na kojoj se korisnik nalazi te razinu hitnosti njegovog slučaja,
- Putevi (engl. *Paths*) - sprema točke pomoću kojih su ulice podijeljene u segmente,

- Ulice (engl. *Streets*) - sprema linije koje predstavljaju ulice u gradu Osijeku, nacrtane pomoću podataka iz tablice Paths.

Sve tablice sadržane u bazi zajedno sa svojim atributima mogu se vidjeti na Slici 6.1.



Slika 6.1: Tablice u bazi podataka.

Tablice se u QGISu kreiraju desnim klikom na bazu na lijevoj strani i klikom na *Execute SQL*. SQL naredbe za kreiranje tablica vidljive su u Programskom kodu 6.2.

```

1 CREATE TABLE streets (
2     id SERIAL PRIMARY KEY,
3     line GEOMETRY(LineString, 4326)
4 );
5
6 CREATE TABLE facility (
7     id SERIAL PRIMARY KEY,
8     naziv VARCHAR(255),
9     zupanija VARCHAR(255),
10    naselje VARCHAR(255),
11    ulica VARCHAR(255),
12    postanski_broj VARCHAR(255),
13    vrsta VARCHAR(255),
14    adresa VARCHAR(255),
15    longitude FLOAT,
16    latitude FLOAT
17 );
18
19 CREATE TABLE paths (
20     id SERIAL PRIMARY KEY,
21     point GEOMETRY(Point, 4326),

```

```

22 );
23
24 CREATE TABLE point (
25     id SERIAL PRIMARY KEY,
26     geom GEOMETRY(Point, 4326),
27     hitnost INT
28 );

```

Programski kod 6.2: SQL kod za kreiranje potrebnih tablica.

Tablica Points popunjena je nasumičnim točkama u gradu Osijeku koristeći QGIS alat *Toggle Editing* koji omogućava crtanje točaka pritiskom na kartu i spremanjem njihovih koordinata. Na isti takav način dobivena je tablica Paths koja služi za podjelu ulica u gradu na manje segmente kako bi se dobila veća točnost. Tablica Streets popunjena je crtanjem linija između točaka iz tablice Paths koristeći *Toggle Editing* alat i *Snapping* koji služi za spajanje izabranih točaka. Podaci iz CSV datoteke uneseni su u tablicu Facility naredbom koja je dana u Programskom kodu 6.3.

```

1 COPY facility(naziv, zupanija, naselje, ulica, postanski_broj,
2     vrsta, adresa, latitude, longitude)
3 FROM 'C:/Users/38595/Desktop/Diplomski rad/podaci/updated_ustanove.
4     csv'
5 DELIMITER ','
6 CSV HEADER;

```

Programski kod 6.3: SQL kod za učitavanje podataka iz CSV datoteke u tablicu Facility.

Naknadno je u tablicu Facility dodan stupac Hitnost kako bi se odredila razina hitnosti koju ta ustanova pruža pacijentima. SQL naredba kojom se određuje razina hitnosti ovisno o ustanovi dana je u Programskom kodu 6.4.

```

1 UPDATE facility
2 SET urgency = CASE
3     WHEN vrsta IN ('Zavod Za Javno Zdravstvo Zupanije', '
4     Dislocirana Jedinica Zavoda Za Javno Zdravstvo') THEN 0
5     WHEN vrsta IN ('Ljekarna', 'Ljekarnicka Jedinica') THEN 1
6     WHEN vrsta IN ('Ambulanta Doma Zdravlja Dislocirana Jedinica',
7     'Dom Zdravlja', 'Poliklinika', 'Ustanova Za Zdravstvenu Skrb', '
8     Dislocirani Odjel Spec. Bolnice') THEN 2
9     WHEN vrsta IN ('Klinicki Bolnicki Centar', 'Ustanova Za Hitnu
10    Medicinsku Pomoc') THEN 3
11 ELSE urgency
12 END;

```

Programski kod 6.4: SQL kod za dodavanje stupca za razinu hitnosti u tablicu Facility.

Kako bi se pronašao najkraći put između dvije točke na karti, kreira se tablica Edges (pogledati Programski kod 6.5) koja sprema podatke o svakom segmentu

ulice, tj. njegov početak, kraj, duljinu i geometrijske informacije.

```

1 CREATE TABLE edges (
2     id SERIAL PRIMARY KEY,
3     source INTEGER,
4     target INTEGER,
5     cost DOUBLE PRECISION,
6     geom GEOMETRY(LineString, 4326)
7 );

```

Programski kod 6.5: SQL kod za kreiranje tablice Edges.

Tablicu Edges popunjavamo na način koji je prikazan u Programskom kodu 6.6.

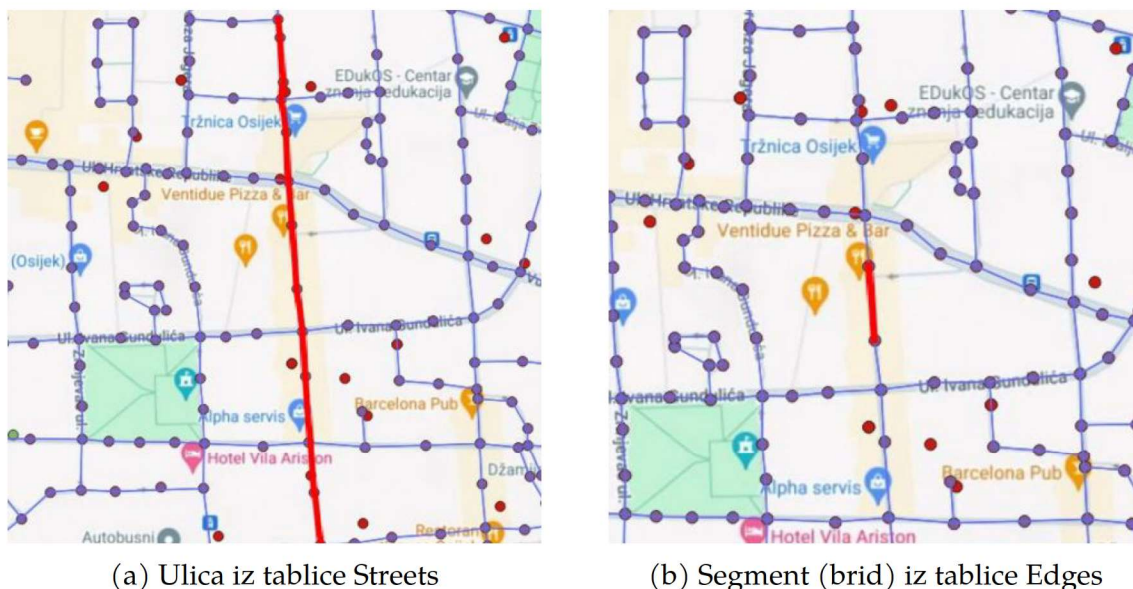
```

1 WITH line_segments AS (
2     SELECT
3         (ST_DumpSegments(line)).geom AS segment_geom
4     FROM
5         streets
6 ),
7 segment_points AS (
8     SELECT
9         segment_geom,
10        ST_StartPoint(segment_geom) AS start_point,
11        ST_EndPoint(segment_geom) AS end_point
12    FROM
13        line_segments
14 ),
15 closest_vertices AS (
16    SELECT
17        sp.segment_geom,
18        p1.id AS source,
19        p2.id AS target,
20        ST_Length(sp.segment_geom) AS cost
21    FROM
22        segment_points sp,
23        paths p1,
24        paths p2
25    WHERE
26        ST_DWithin(sp.start_point, p1.point, 0.0001)
27        AND ST_DWithin(sp.end_point, p2.point, 0.0001)
28 )
29 INSERT INTO edges (source, target, cost, geom)
30 SELECT
31     source,
32     target,
33     cost,
34     segment_geom
35 FROM
36     closest_vertices;

```

Programski kod 6.6: SQL kod za popunjavanje tablice Edges.

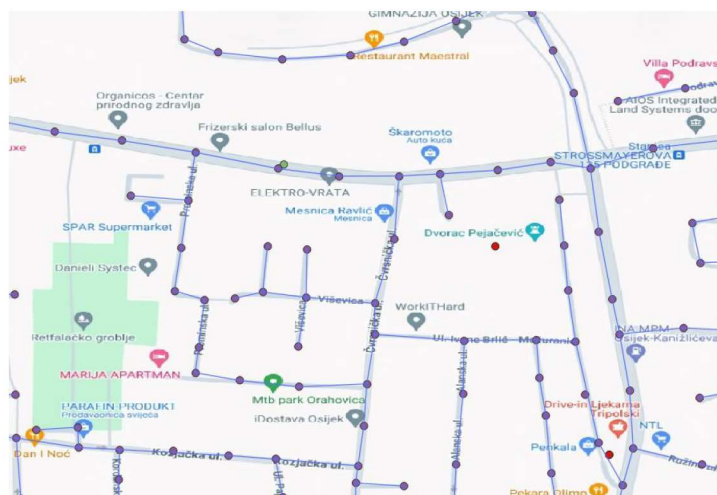
Funkcija `ST_DumpSegments` izdvaja segmente iz geometrijskih informacija tablice `Streets` u `line_segments`. U `segment_points` se pomoću funkcija `ST_StartPoint` i `ST_EndPoint` sprema početna, tj. krajnja točka svakog segmenta. Varijabla `closest_vertices` pronalazi točke iz tablice `Paths` najbliže početnoj i krajnjoj točki segmenta i računa duljinu segmenta. Pomoću funkcije `ST_DWithin` provjerava se je li početna točka segmenta unutar 0,0001 jedinica točaka iz `Paths`, tj. osigurava se da se početna (krajnja) točka segmenta poklapa s točkom iz `Paths`. Na Slici 6.2 prikazana je razlika između ulice i linije od jednog segmenta.



Slika 6.2: Razlika između ulice i segmenta (brida)

Nakon što su se tablice popunile, u QGISu je instalirano proširenje `QuickMapServices` koji omogućava dodavanje `Google Terrain`², preuzetog s `Google Karata`, kako bi sve ljepše izgledalo. Ovaj, ili neki drugi sloj iz `QuickMapServices` proširenja može se dodati klikom na `Web` → `QuickMapServices` → `Google` → `Google Terrain`. Desnim klikom na neku od tablica na lijevoj strani i odabirom `Add Layer to Project` podaci iz tablice dodaju se na našu kartu.

²`Google Terrain` je jedan od slojeva dostupnih u QGIS-u koji omogućava prikaz kao na `Google Kartama`, s ulicama, restoranima, znamenitostima, trgovinama i hotelima.



Slika 6.3: QGIS prikaz jednog dijela Osijeka.

Na Slici 6.3 plavom bojom su prikazane linije koje predstavljaju ulice, ljubičastim točkama prikazane su točke iz tablice Paths pomoću kojih je ulica podijeljena na segmente, crvenim točkama prikazane su zdravstvene ustanove, a zelenim točkama nasumične lokacije u gradu od kojih ćemo gledati udaljenost i put do najbliže ustanove. Iako se već ovim načinom iscrtavanja postiže dovoljno dobra točnost i prikaz, povećanjem gustoće točaka u tablici Paths i smanjivanjem segmenata ulica točnost bi se još dodatno povećala.

6.3 Upiti na bazi

6.3.1 Pronalazak najbliže zdravstvene ustanove

Ovako definirana geoprostorna baza podataka omogućava upit za pronalazak najkraćeg puta do najbliže zdravstvene ustanove s obzirom na hitnost koristeći gotove funkcije iz PostGISa i pgRoutinga.

SQL kod korišten za ovaj upit bit će objašnjen u nekoliko koraka, a cijeli kod dostupan je na GitHubu.

```

1 WITH point_of_interest AS (
2     SELECT geom, hitnost
3     FROM point
4     WHERE id = 19
5 )

```

Programski kod 6.7: Odabir početne točke interesa.

U Programskom kodu 6.7 point_of_interest predstavlja točku od koje krećemo s računanjima, tj. točku interesa na karti.

```

1 closest_facility AS (
2     SELECT

```

```

3         f.id AS facility_id,
4         f.geom AS facility_geom,
5         ST_Distance(f.geom, poi.geom) AS distance_to_point
6     FROM
7         facility f,
8         point_of_interest poi
9     WHERE (poi.hitnost IS NULL OR f.hitnost = poi.hitnost)
10    ORDER BY
11        ST_Distance(f.geom, poi.geom)
12    LIMIT 1
13 )

```

Programski kod 6.8: Pronalazak najbliže zdravstvene ustanove od točke interesa s obzirom na razinu hitnosti.

Varijablu `closest_facility`, koju vidimo u Programskom kodu 6.8, pronalazi najbližu zdravstvenu ustanovu s obzirom na `point_of_interest`. Klauzula `WHERE` provjerava je li hitnost točke interesa `NULL` ili slaže li se s hitnosti ustanove. Na kraju se rezultati slažu s obzirom na `ST_Distance` i uzima se samo prva ustanova na popisu, tj. osigurava se da je odabrana samo najbliža ustanova.

```

1 closest_point_to_given_point AS (
2     SELECT
3         p.id AS path_id,
4         p.point AS path_geom,
5         ST_Distance(p.point, poi.geom) AS distance_to_point
6     FROM
7         paths p,
8         point_of_interest poi
9     ORDER BY
10        ST_Distance(p.point, poi.geom)
11    LIMIT 1
12 ),
13 closest_point_to_facility AS (
14     SELECT
15         p.id AS path_id,
16         p.point AS path_geom,
17         ST_Distance(p.point, cf.facility_geom) AS
18         distance_to_facility
19     FROM
20         paths p,
21         closest_facility cf
22     ORDER BY
23        ST_Distance(p.point, cf.facility_geom)
24    LIMIT 1
25 )

```

Programski kod 6.9: Pronalazak točke iz tablice `Paths` najbliže danoj točki interesa i točke iz tablice `Paths` najbliže najbližoj ustanovi.

Kao i kod pronalaska najbliže ustanove u Programskom kodu 6.8, u Programskom kodu 6.9 se pronalazi točka iz tablice `Paths` najbliža danoj točki interesa i točka iz tablice `Paths` najbliža najbližoj ustanovi.

```

1 SELECT
2     dij.seq,
3     dij.path_seq,
4     dij.node,
5     dij.edge,
6     dij.cost,
7     dij.agg_cost,
8     e.geom
9 FROM pgr_dijkstra(
10     'SELECT id, source, target, cost FROM edges',
11     (SELECT path_id FROM closest_point_to_given_point),
12     (SELECT path_id FROM closest_point_to_facility), false
13 ) AS dij
14 JOIN edges e ON dij.edge = e.id;

```

Programski kod 6.10: Pronalazak najkraćeg puta između dvije točke na grafu.

Zadnji dio upita, tj. Programski kod 6.10 koristi funkciju `pgr_dijkstra` iz ekstenzije `pgRouting` koja koristi Dijkstrin algoritam za pronalazak najkraćeg puta na grafu. [vidjeti [2]] Funkcija `pgr_dijkstra` pronalazi najkraći put s obzirom na SQL upit u Programskom kodu 6.10 na liniji 10 koji uzima osnovne informacije brida, s početkom u najbližoj točki iz tablice `Paths` dane točke (`closest_point_to_given_point`) i završetkom u najbližoj točki tablice `Paths` od najbliže ustanove (`closest_point_to_facility`). `False` na kraju funkciju govori da se ovdje radi o neusmjerenom grafu. Rezultat `dij` se pomoću `JOIN` operatora pridružuje tablici `Edges` tako da je `dij.edge=e.id`, čime se uzimaju geometrijske informacije svakog brida u najkraćem putu.

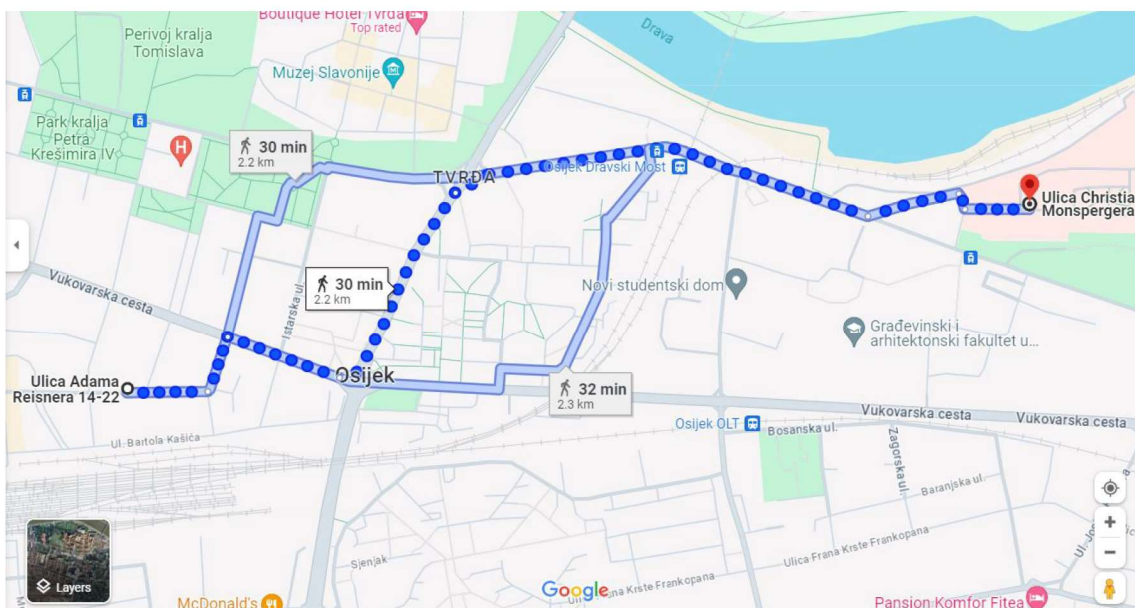
	seq	path_seq	node	edge	cost	agg_cost	geom
1	1	1	2483	4578	0,...	0	0102000020E61...
2	2	2	2350	4550	0,...	0,...	0102000020E61...
3	3	3	2349	4549	0,...	0,...	0102000020E61...
4	4	4	2348	4548	0,...	0,...	0102000020E61...
5	5	5	2347	4547	0,...	0,...	0102000020E61...
6	6	6	2346	4529	0,...	0,...	0102000020E61...
7	7	7	2504	4528	0,...	0,...	0102000020E61...
8	8	8	2505	4527	0,...	0,...	0102000020E61...
9	9	9	2508	4531	0,...	0,...	0102000020E61...

Slika 6.4: Rezultat upita koji vraća put do najbliže ustanove.

Na Slici 6.4 prikazan je rezultat prethodnog upita. Ispisani su svi čvorovi, tj. točke, i svi bridovi koji se posjete na putu od dane točke do najbliže ustanove. Rezultate upita moguće je spremirati u novu tablicu i kao sloj dodati u projekt, nakon čega se jasno vidi put koji je algoritam dao.



Slika 6.5: Prikaz puta (narančasta linija) od dane točke do zdravstvene ustanove.



Slika 6.6: Prikaz puta na Google Karti od iste točke do iste zdravstvene ustanove kao u primjeru sa Slike 6.5.

Na ovom primjeru (Slika 6.5 i Slika 6.6) se vidi da upiti na bazi daju isto rješenje, tj isti najkraći put do ustanove kao i Google Karte. Također, kao i geoprostorna baza u ovom radu, Google Karte koriste točke kako bi podijelile ulice na segmente, ali su njihove točke puno gušće, a segmenti manji, zbog čega im je točnost puno veća.

6.3.2 Pronalazak zdravstvenih ustanova u danom radijusu

Još jedan primjer upita koji ova baza podataka dopušta je pronalazak zdravstvenih ustanova u nekom danom radijusu.

```

1 WITH point_of_interest AS (
2     SELECT geom
3     FROM point
4     WHERE id = 20
5 )

```

```

6 SELECT
7     f.id,
8     f.naziv,
9     ST_Distance(f.geom, poi.geom) AS distance_meters
10 FROM
11     facility f
12 CROSS JOIN
13     point_of_interest poi
14 WHERE
15     ST_DWithin(f.geom, poi.geom, 400)
16 ORDER BY
17     distance_meters;

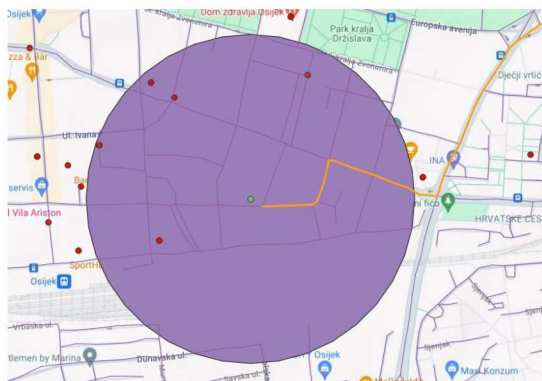
```

Programski kod 6.11: Upit za pronalazak svih zdravstvenih ustanova u danom radijusu.

Upit dan u Programskom kodu 6.11 uzima točku interesa i pomoću funkcije `ST_Distance` i `CROSS JOIN` operator računa udaljenost između naše točke interesa i svake zdravstvene ustanove. Klauzula `WHERE` osigurava da se uzimaju u obzir samo ustanove udaljene najviše 400 metara od točke interesa.

	id	naziv	distance_meters
1	3024	Poliklinika Za ...	248,86762692
2	2710	Odjel Osijek-...	314,52427978
3	3378	Ustanova Za ...	336,49741247
4	2956	Poliklinika Otos...	378,83881427
5	1152	Centar Za Ocne...	399,15481641
6	3581	Ustanova Za ...	399,15481641

Slika 6.7: Rezultat upita koji vraća sve zdravstvene ustanove u radijusu od 400m oko točke interesa.



Slika 6.8: Radijus od 400 metara oko točke interesa.

Točnost upita može se provjeriti crtanjem kružnice oko točke interesa kao što je prikazano na Slici 6.8.

Literatura

- [1] D.J. PEUQUET, D.F. MARBLE, *Introductory readings in Geographic Information Systems*, Department of Geography, The Pennsylvania State University, Department of Geography The Ohio State University, 1990.
- [2] M. SNIEDOVICH, *Dijkstra's algorithm revisited: the dynamic programming connexion*, Department of Mathematics and Statistics, The University of Melbourne, Australia, 2006.
- [3] R.H. GÜTING, *An Introduction to Spatial Database Systems*, Praktische Informatik IV, FernUniversität Hagen, 1994.
- [4] S. CHAWLA, S. SHEKHAR, *Spatial Databases: A Tour*, Prentice Hall, 2003.
- [5] *Department of geography and environmental science*, dostupno na <http://www.geography.hunter.cuny.edu/>.
- [6] *Database management system*, dostupno na <https://www.techtarget.com/searchdatamanagement/definition/database-management-system>.
- [7] *Docker dokumentacija*, dostupno na <https://docs.docker.com/>.
- [8] *GIS (Geographic Information System)*, dostupno na <https://education.nationalgeographic.org/resource/geographic-information-system-gis/>.
- [9] *Hrvatski zavod za javno zdravstvo*, dostupno na <https://www.hzjz.hr/>.
- [10] *PostGIS dokumentacija*, dostupno na <https://www.postgis.net/>.
- [11] *PostGIS/PostgreSQL*, dostupno na <https://mapserver.org/input/vector/postgis.html#postgis-postgresql>.
- [12] *PostgreSQL dokumentacija*, dostupno na <https://www.postgresql.org/>.

-
- [13] *QGIS dokumentacija*, dostupno na
<https://docs.qgis.org/2.18/en/docs/index.html>.
- [14] *R-Stablo*, dostupno na
<https://www.oninit.com/manual/informix/100/rtree/rtree29.htm>.
- [15] *Setup PostgreSQL with PostGIS on Docker*, dostupno na
<https://trevorstanley.medium.com/setup-postgresql-with-postgis-on-docker-880>.
- [16] *The Types of Databases*, dostupno na
<https://www.matillion.com/blog/the-types-of-databases-with-examples>.
- [17] *What is a Database?*, dostupno na
<https://www.simplilearn.com/tutorials/dbms-tutorial/what-is-a-database>.

Sažetak

U ovom radu objašnjeno je što su geoprostorne baze podataka, opisana su njihova svojstva i navedena je njihova upotreba. Također detaljno je opisano što su geoprostorni podaci i geoprostorno indeksiranje. Kako bi se bolje razumjele geoprostorne baze podataka, objašnjene su baze podataka i geografski informacijski sustavi te njihova primjena. Kao najčešće korišten alat za rad s geoprostornim bazama podataka, detaljno je objašnjen PostGIS, tj. proširenje PostgreSQL baze podataka. U zadnjem dijelu rada prikazana je primjena geoprostorne baze podataka i opisano je kako pronaći najbližu zdravstvenu ustanovu od dane lokacije te kako pronaći sve zdravstvene ustanove unutar danog radijusa.

Ključne riječi

baza podataka, geografski informacijski sustav, geoprostorni podaci, geoprostorno indeksiranje, PostGIS

Spatial databases

Summary

This work explains what spatial databases are, describes their features, and lists their uses. It also describes in detail what spatial data and spatial indexing are. Also, to understand spatial databases better, it explains what databases and geographic information systems are. As the most used tool for working with spatial databases, this work explains PostGIS, an extension for PostgreSQL databases. The last part of this paper explains spatial database application and describes how to find the closest health institution from the given location and how to find all health institutions in a given radius.

Keywords

database, geographic information system, spatial data, spatial indexing, PostGIS

Životopis

Rođena sam 1999. godine u Osijeku gdje završavam osnovnu i srednju školu. 2018. godine upisujem preddiplomski studij matematike i računarstva na Odjelu za matematiku u Osijeku. 2021. odlazim na Erasmus+ studentsku razmjenu u Krakov, gdje provodim zimski semestar. Preddiplomski studij završavam 2022. godine i upisujem diplomski studij matematike, smjer matematika i računarstvo na istom fakultetu.