

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Danijel Filipović

IMPLEMENTACIJE PROSTORNIH TIPOVA
PODATAKA
ZAVRŠNI RAD

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Danijel Filipović

Matični broj: 39950/11-R

Studij: Informacijski sustavi

IMPLEMENTACIJE PROSTORNIH TIPOVA
PODATAKA
ZAVRŠNI RAD

Mentor

Prof. dr. sc. Vjeran Strahonja

Varaždin, srpanj 2015.

Sadržaj

1. Uvod.....	1
2. Prostorni tipovi podataka.....	2
2.1. Općenito.....	2
2.2. Objektni model za geometriju.....	2
2.3. <i>Geometry</i>	3
2.3.1. Opis.....	3
2.3.2. Prostorne funkcije.....	4
2.4. <i>Point</i>	8
2.4.1. Opis.....	8
2.4.2. Prostorne funkcije.....	9
2.5. <i>Curve</i>	9
2.5.1. Opis.....	9
2.5.2. Prostorne funkcije.....	10
2.6. <i>LineString</i>	11
2.6.1. Opis.....	11
2.6.2. Prostorne funkcije.....	11
2.7. <i>Surface</i>	12
2.7.1. Opis.....	12
2.7.2. Prostorne funkcije.....	12
2.8. <i>Polygon</i>	13
2.8.1. Opis.....	13
2.8.2. Prostorne funkcije.....	14
2.9. <i>GeometryCollection</i>	15
2.9.1. Opis.....	15
2.9.2. Prostorne funkcije.....	15
2.10. <i>MultiPoint</i>	16
2.10.1. Opis.....	16
2.10.2. Prostorne funkcije.....	16
2.11. <i>MultiCurve</i>	16
2.11.1. Opis.....	16
2.11.2. Prostorne funkcije.....	17
2.12. <i>MultiLineString</i>	17

2.12.1. Opis.....	17
2.12.2. Prostorne funkcije.....	18
2.13. <i>MultiSurface</i>	18
2.13.1. Opis.....	18
2.13.2. Prostorne funkcije.....	18
2.14. <i>MultiPolygon</i>	19
2.14.1. Opis.....	19
2.14.2. Prostorne funkcije.....	20
3. Praktičan primjer.....	21
3.1. Korištena tehnologija.....	21
3.2. Primjer.....	23
Literatura.....	30

1. Uvod

Ovaj rad se posvećuje analizi prostornih tipova podataka koji se pojavljuju u modernim sustavima za upravljanje bazom podataka. Rad se dijeli na dva dijela: analizi prostornih tipova podataka i praktični primjer.

U prvom dijelu će se općenito opisati i pokazati svojstva onih prostornih tipova podataka koji se pojavljuju u sustavima za upravljanje bazom podataka koji su uzeti kao uzorak. To jest, od svih prostornih tipova podataka opisat će se oni prostorni tipovi koji su najčešće implementirani u odabranim sustavima za upravljanje bazama podataka.

Drugi dio rada će pokazati praktičan primjer korištenja prostornih tipova podataka u specijaliziranom sustavom za upravljanje bazom podataka: PostgreSQL sa PostGIS dodatkom. Na početku će se opisati okruženje u kojem će se raditi primjer, a poslije će se pokazati općeniti primjer i objasniti pojedini dijelovi tog primjera.

2. Prostorni tipovi podataka

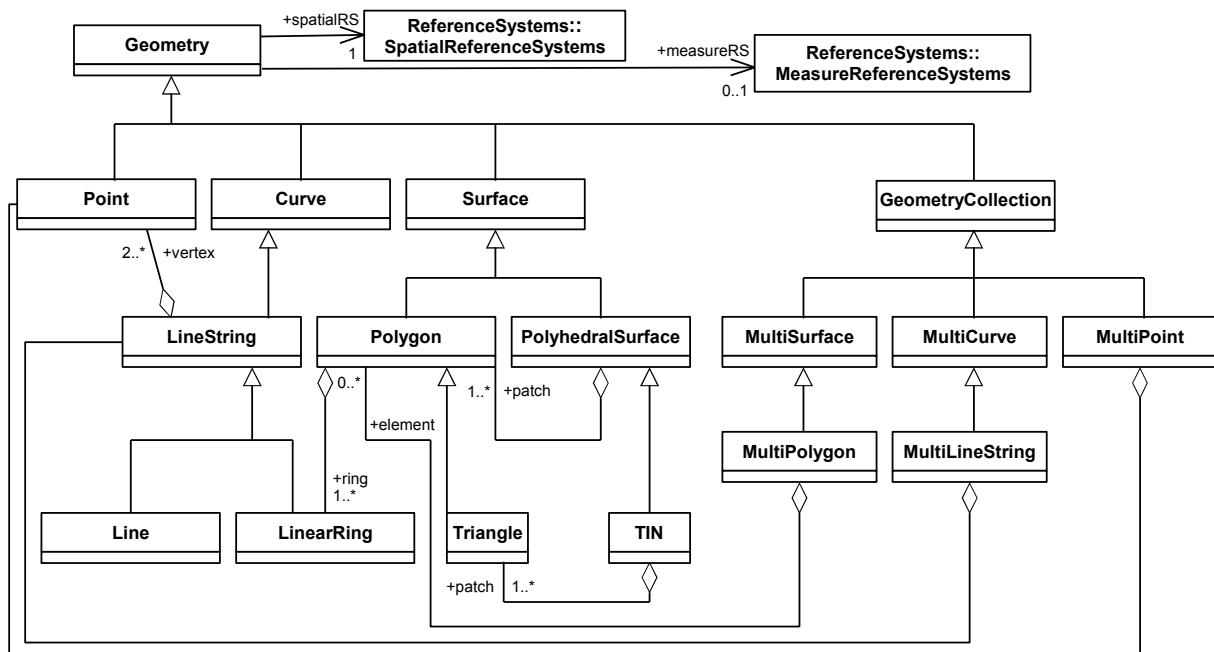
2.1. Općenito

Za analizu prostornih tipova podataka, uzeti su oni tipovi podataka koji se koriste u Microsoft SQL Server, MySQL i PostgreSQL sustavima za upravljanje bazom podataka. Dok Microsoft SQL Server i MySQL imaju implementirane prostorne tipove podataka u svojoj jezgri, PostgreSQL koristi PostGIS dodatak da bi omogućio podršku za takve tipove podataka.

Prostorne tipove podataka je osmislila i opisala organizacija zvana Open Geospatial Consortium (u daljem tekstu OGC) koja se bavi razvojem javno dostupnih standarda na području geografskih informacijskih sustava.

Glavni prostorni tipovi koji se koriste u Microsoft SQL Server, MySQL i PostgreSQL sustavima za upravljanje bazom podataka su sljedeći: *Geometry*, *Point*, *Curve*, *Surface*, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon* i *GeometryCollection*. OGC je definirao više prostornih tipova podataka i neki sustavi za upravljanje bazom podataka su definirali svoje vlastite tipove, ali navedeni tipovi se pojavljuju u sva tri navedena sustava.

2.2. Objektni model za geometriju



Slika 1: Hijerarhija klasa objektnog modela geometrije (Prema: Open Geospatial Consortium, 2011:14)

Kao što je već prethodno navedeno, prostorne tipove podataka je definirao OGC u svojem *Simple Feature Access* standardu. Ti tipovi podataka su opisani objektnim modelom za geometriju (eng. *geometry object model*) i za njega je napravljen dijagram klasa, odnosno hijerarhija klasa, prikazana na slici 1.

Klasa *Geometry* je temeljna klasa koja predstavlja neki geometrijski objekt i ona sadrži osnovne podklase: *Point*, *Curve*, *Surface* i *GeometryCollection*. Svaki geometrijski objekt je povezan sa prostornim referentnim sustavom (eng. *Spatial Reference System*) koji opisuje dimenziju koordinatnog prostora u kojem se nalazi geometrijski objekt (Open Geospatial Consortium, 2011:13). Nadalje, te osnovne podklase su superklase za ostale klase poput *LineString*, *Polygon*, *PolyhedralSurface*, itd.

Za rad s kolekcijom geometrijskih objekata, poput *Point*, *LineString*, *Polygon* i dr., objektni model za geometriju definira specijalizirane 0-, 1- i 2-dimenzionalne klase, kolekcije koje sadrže više geometrijskih objekata, kao što su to *MultiPoint*, *MultiLineString*, *MultiPolygon*, itd. (Open Geospatial Consortium, 2011:14). Takve klase nasljeđuju klasu *GeometryCollection*, a jedina iznimka su klase *MultiPolygon* i *MultiLineString* koje nasljeđuju klase *MultiSurface* i *MultiCurve*, respektivno.

2.3. Geometry

2.3.1. Opis

Geometry je temeljna klasa u objektnom modelu geometrije i ona je također apstraktna klasa (Open Geospatial Consortium, 2011:14). To znači da se klasa *Geometry* smije samo proširiti, ne instancirati. Drugim riječima, *Geometry* predstavlja predložak za geometrijske objekte koji će naslijediti ovu klasu.

Klasa *Geometry* prema OGC-ovoj *Simple Feature Access* specifikaciji (2011:15-17) sadrži sljedeće vrste funkcija: osnovne funkcije geometrijskih objekata, funkcije za testiranje prostornih odnosa između geometrijskih objekata i funkcije za prostornu analizu. Te funkcije se zajednički zovu prostorne funkcije i one se za *Geometry* detaljnije razrađuju u potpoglavlju 2.3.2. *Prostorne funkcije*.

OGC (2011:14) u svojoj specifikaciji navodi da su djeca klase *Geometry*, odnosno podklase te klase, ograničeni na 0, 1 i 2-dimenzionalne geometrijske objekte unutar 2, 3 ili 4-

dimenzionalnom koordinatnom prostoru (R^2 , R^3 ili R^4). 0, 1 i 2-dimenzionalni geometrijski objekti su točka, pravac (ili linija) i površina, respektivno.

Dimenzija koordinatnog prostora u kojoj se nalazi geometrijski objekt ovisi o danom prostornom referentnom sustavom i također ovisi o tome je li geometrijskom objektu dan referentni sustav za mjerenje (eng. *measure reference system*). Tu vezu možete vidjeti u slici 1 u potpoglavlju 2.1. *Objektni model za geometriju*. Ovisno o danim referentnim sustavima, točka (*Point*) može imati sljedeće koordinatne vrijednosti (Open Geospatial Consortium, 2011:14):

- u R^2 točka ima koordinatne vrijednosti x i y ,
- u R^3 točka ima koordinatne vrijednosti x , y i z ili x , y i m ,
- u R^4 točka ima koordinatne vrijednosti x , y , z i m .

Koordinatne vrijednosti x i y predstavljaju pozicije točke u prostoru, i ta osobina vrijedi za sve dimenzije koordinatnog prostora.

“Koordinatna vrijednost z po običaju predstavlja treću dimenziju (tj. 3D). U geografskim informacijskim sustavima (GIS) to se može odnositi na nadmorsku visinu ili dubinu” (Open Geospatial Consortium, 2011:18).

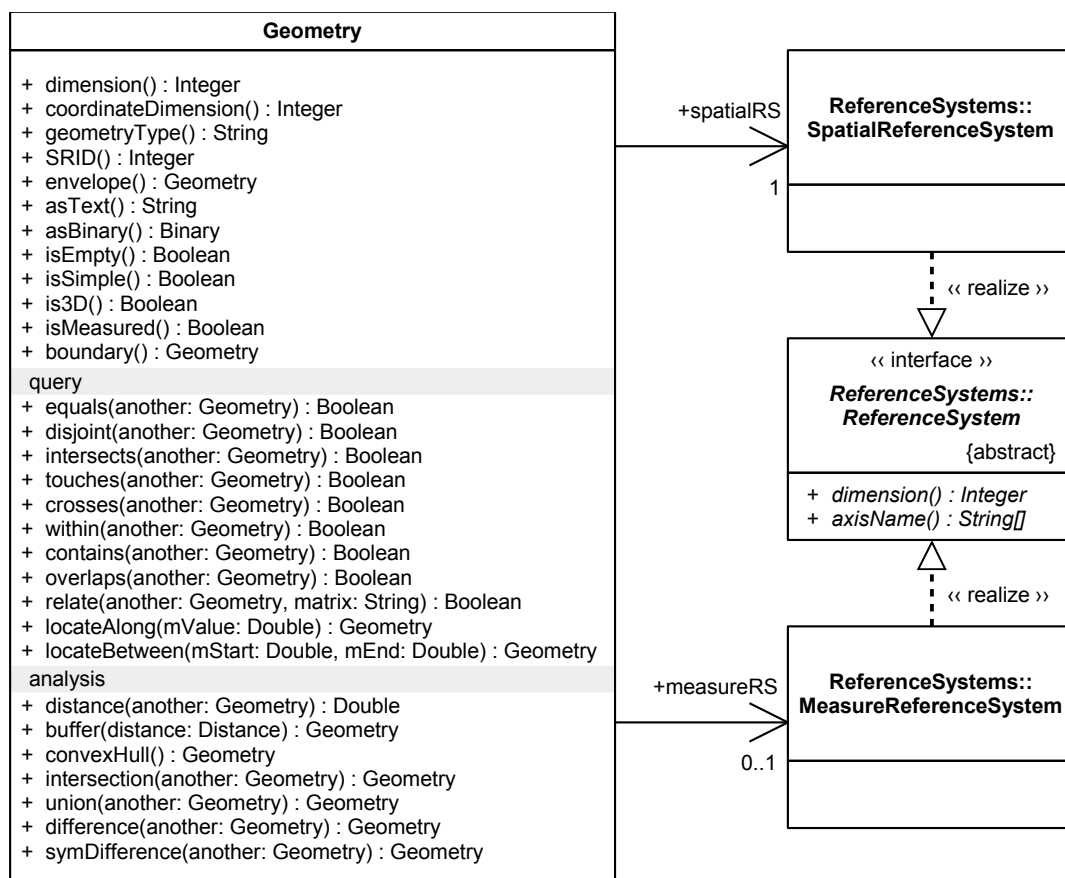
“Koordinatna vrijednost m omogućava programskom okruženju da asocira neku mjeru sa vrijednosti točke.” (Open Geospatial Consortium, 2011:18). Na primjer: rijeka se može prikazati kao linija (odnosno, kao prostorni tip podatka *LineString*) u kojoj koordinatna vrijednost m mjeri njenu duljinu od njenog ušća.

Većina informacija o klasi *Geometry* je preuzeta iz OGC-ove *Simple Feature Access* specifikacije, što ne znači da sve informacije, koje se nalaze unutar ovog potpoglavlja, vrijede za sve sustave u kojima su implementirani prostorni tipovi podataka.

2.3.2. Prostorne funkcije

Kao što je već navedeno, prostorne funkcije za prostorni tip podatka *Geometry* se dijele u 3 kategorije: osnovne funkcije geometrijskih objekata, funkcije za testiranje prostornih odnosa između geometrijskih objekata i funkcije za prostornu analizu. Kako se

Geometry nalazi na najvišoj razini generalizacije prema objektnom modelu za geometrije, sve razine podklasa (podklase od *Geometry*, podklase tih podklasa, itd.) će moći koristiti funkcije definirane za taj tip podatka. Slika 2 prikazuje dijagram klasa za *Geometry* kako ga je osmislio OGC (2011:15).



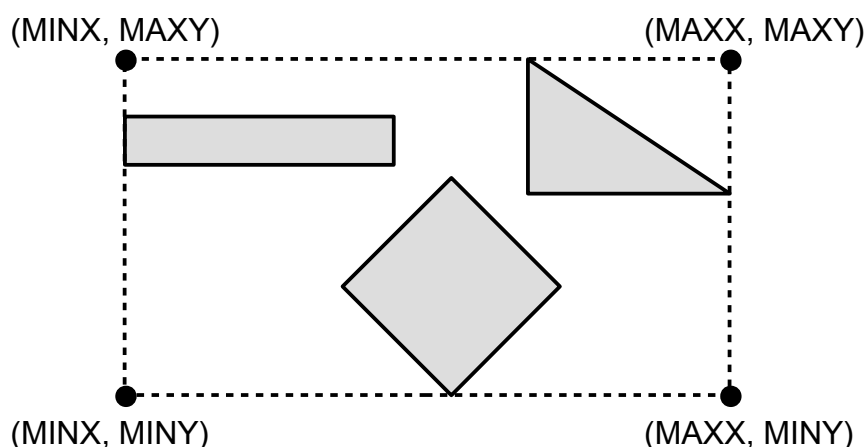
Slika 2: *Geometry* (Prema: Open Geospatial Consortium, 2011:15)

Osnovne funkcije geometrijskih objekata su (Open Geospatial Consortium, 2011:15-16):

- *dimension () : Integer* - vraća dimenziju geometrijskog objekta (vraća 0, 1 ili 2),
- *coordinateDimension () : Integer* - vraća koordinatnu dimenziju u kojoj se nalazi geometrijski objekt (vraća 2, 3 ili 4),
- *geometryType () : String* - vraća naziv tipa geometrijskog objekta, naziv je različit za svaku podklasu klase *Geometry* (npr. geometrijski objekt tipa *Point* će vratiti “*POINT*”, *Curve* će vratiti “*CURVE*”, itd.),
- *SRID () : Integer* - vraća identifikacijski broj prostornog referentnog sustava (eng.

Spatial Reference System ID), u slučaju sustava za upravljanje bazom podataka to će biti vrijednost vanjskog ključa na tablicu prostornih referentnih sustava,

- *envelope () : Geometry* - vraća geometrijski objekt koji predstavlja minimalni okvir u kojem se nalazi trenutni geometrijski objekt. Sastoji se od skupa od 4 točaka $[(MINX, MINY), (MAXX, MINY), (MAXX, MAXY), (MINX, MAXY)]$, sam okvir izgleda kao pravokutnik. Za bolji prikaz pogledati sliku 3.



Slika 3: Primjer funkcije *envelope()*, isprekidana linija je geometrijski objekt koji će funkcija vratiti na temelju dane kolekcije geometrijskih objekata

- *asText () : String* - izvozi geometrijski objekt u tekstualni oblik,
- *asBinary () : Binary* - izvozi geometrijski objekt u binarni oblik,
- *isEmpty () : Boolean* - vraća *TRUE* ako se radi o praznom geometrijskom objektu, tj. ako je skup točaka koji čine geometrijski objekt prazan,
- *isSimple () : Boolean* - vraća *TRUE* ako geometrijski objekt ne presijeca i ne dodiruje samoga sebe,
- *is3D () : Boolean* - vraća *TRUE* ako geometrijski objekt ima koordinatu *z*,
- *isMeasured () : Boolean* - vraća *TRUE* ako geometrijski objekt ima koordinatu *m*,
- *boundary () : Geometry* - vraća granicu geometrijskog objekta kao sami geometrijski

objekt.

Funkcije za testiranje prostornih odnosa između geometrijskih objekata su sljedeće (Open Geospatial Consortium, 2011:16-17):

- *equals (another: Geometry) : Boolean* - provjerava je li geometrijski objekt jednak geometrijskom objektu koji je dan kao parametar *another*,
- *disjoint (another: Geometry) : Boolean* - provjerava je li geometrijski objekt odvojen od geometrijskog objekta *another*, tj. vraća *TRUE* ako dva geometrijska objekta nemaju zajedničkih točaka,
- *intersects (another: Geometry) : Boolean* - provjerava sječe li trenutni geometrijski objekt drugi geometrijski objekt dan kao parametar *another*,
- *touches (another: Geometry) : Boolean* - provjerava dodiruje li trenutni geometrijski objekt drugi geometrijski objekt,
- *crosses (another: Geometry) : Boolean* - provjerava imaju li dva geometrijska objekta neke zajedničke točke unutar svojih granica, te je li njihov presjek neprazan skup točaka i je li dimenzija presjeka manja od maksimalne dimenzije između dva geometrijska objekta (PostGIS, str. 250),
- *within (another: Geometry) : Boolean* - provjerava nalazi li se geometrijski objekt unutar geometrijskog objekta *another*,
- *contains (another: Geometry) : Boolean* - provjerava sadrži li geometrijski objekt unutar sebe geometrijski objekt *another*,
- *overlaps (another: Geometry) : Boolean* - provjerava je li presjek dvaju geometrijskih objekata različit, ali da se nalazi u istoj dimenziji,
- *relate (another: Geometry, matrix: String) : Boolean* - provjerava prostorni odnos trenutnog geometrijskog objekta s drugim tako da provjerava presjeka između njih na temelju dane matrice,
- *locateAlong (mValue: Double) : Geometry* - vraća deriviranu kolekciju geometrijskih

objekata na temelju parametra *mValue* koji predstavlja koordinatu *m*,

- *locateBetween (mStart: Double, mEnd: Double) : Geometry* - vraća deriviranu kolekciju geometrijskih objekata unutar danog raspona koji je specificiran parametrima *mStart* i *mEnd*, a ti parametri predstavljaju raspon vrijednosti za koordinatu *m*.

Funkcije za prostornu analizu su (Open Geospatial Consortium, 2011:17):

- *distance (another: Geometry) : Double* - vraća najmanju udaljenost između dvije točke danih dva geometrijska objekta,
- *buffer (distance: Double) : Geometry* - vraća geometrijski objekt koji se sastoji od skupa točaka koji se nalaze unutar dane udaljenosti *distance*,
- *convexHull () : Geometry* - vraća geometrijski objekt koji predstavlja konveksnu ljusku trenutnog geometrijskog objekta,
- *intersection (another: Geometry) : Geometry* - vraća presjek dvaju geometrijskih objekata kao zasebni geometrijski objekt,
- *union (another: Geometry) : Geometry* - vraća geometrijski objekt kao skup točaka koji predstavlja uniju dva geometrijskih objekata,
- *difference (another: Geometry) : Geometry* - vraća geometrijski objekt kao skup točaka koji predstavlja razliku između dva geometrijskih objekata,
- *symDifference (another: Geometry) : Geometry* - vraća geometrijski objekt kao skup točaka koji predstavlja simetričnu razliku između dva geometrijska objekta.

2.4. Point

2.4.1. Opis

Prema OGC *Simple Feature Access* specifikaciji (2011:20-21) *Point* (točka) je “0-dimenzionalni geometrijski objekt koji predstavlja jednu poziciju u koordinatnom prostoru.” Također se navodi da *Point* sadrži koordinatne vrijednosti *x* i *y* te, ovisno o pridruženom

prostornom referentnom sustavu, može također imati koordinatne vrijednosti z i m .

U nekim implementacijama, kao npr. u MySQL sustavu za upravljanje bazom podataka, *Point* može sadržati samo x i y koordinatne vrijednosti (MySQL 5.6 Reference Manual, str. 1284).

2.4.2. Prostorne funkcije

Geometry	
Point	
+	X() : Double
+	Y() : Double
+	Z() : Double
+	M() : Double

Slika 4: *Point* (Prema: Open Geospatial Consortium, 2011:21)

U specifikaciji OGC je definirao 4 jednostavne funkcije za prostorni tip podatka *Point* (Open Geospatial Consortium, 2011:21):

- $X()$: *Double* - vraća x koordinatnu vrijednost točke,
- $Y()$: *Double* - vraća y koordinatnu vrijednost točke,
- $Z()$: *Double* - vraća z koordinatnu vrijednost točke, ako je točka ima, inače vraća *NULL*,
- $M()$: *Double* - vraća m koordinatnu vrijednost točke, ako je točka ima, inače vraća *NULL*.

2.5. Curve

2.5.1. Opis

Klasa *Curve* (krivulja) je 1-dimenzionalni geometrijski objekt koja je prikazana kao niz točaka (ili niz instanci klase *Points*) (MySQL 5.6 Reference Manual, str. 1284). Prema OGC-ovoj specifikaciji (2011:21) oblik interpolacije između točaka definiraju djeca, tj. podklase, klase *Curve*. Također, prema istoj navedenoj specifikaciji, u objektnom modelu geometrije postoji samo jedna podklasa od *Curve*: klasa *LineString*.

Curve je također apstraktna klasa, što znači da se ona ne može instancirati već samo definira osobine koje će njena djeca naslijediti.

Krivulja prikazana preko klase *Curve* može imati dvije osobine (Open Geospatial Consortium, 2011:22):

- krivulja može biti jednostavna ako ne prolazi kroz istu točku više od jednom (jedina iznimka su krajnje točke krivulje),
- krivulja može biti zatvorena ako joj je početna točka istovremena i krajnja točka,
- krivulja je topološki zatvorena.

Ako je krivulja i jednostavna i zatvorena, onda ona čini prsten, odnosno ona čini klasu *LinearRing* po objektnom modelu za geometriju (MySQL 5.6 Reference Manual, str. 1285). No, *LinearRing* se po istom objektnom modelu može jedino ostvariti preko klase *LineString*.

2.5.2. Prostorne funkcije

Geometry	
Curve	
+ length()	: Double
+ startPoint()	: Point
+ endPoint()	: Point
+ isClosed()	: Boolean
+ isRing()	: Boolean

Slika 5: *Curve* (Prema: Open Geospatial Consortium, 2011:22)

Curve je apstraktna klasa te sve definirane funkcije koriste samo djeca ove klase. Ono definira sljedeće funkcije (Open Geospatial Consortium, 2011:22):

- *length () : Double* - vraća duljinu krivulje,
- *startPoint () : Point* - vraća početnu točku krivulje,
- *endPoint () : Point* - vraća završnu točku krivulje,
- *isClosed () : Boolean* - vraća *TRUE* ako je krivulja zatvorena, tj. ako su joj početna i završna točka jednake (*startPoint() = endPoint()*),

- *isRing () : Boolean* - vraća *TRUE* ako je krivulja zatvorena i ako je jednostavna (ne prolazi kroz istu točku više od jednom).

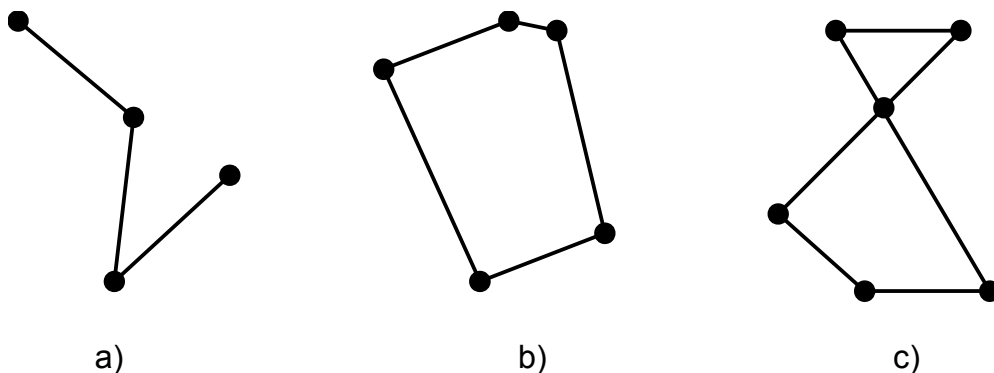
2.6. *LineString*

2.6.1. Opis

Klasa *LineString* je dijete klase *Curve*. OGC u *Simple Access Features* (2011:22) navodi da je *LineString* krivulja (tj. *Curve*) sa linearnom interpolacijom između točaka. Svaki par točaka čini liniju (Open Geospatial Consortium, 2011:22).

Preko klase *LineString* se realizira klasa prsten, tj. *LinearRing* - ako je *LineString* jednostavan i zatvoren, onda ono čini prsten.

Jedna moguća primjena klase *LineString* je da se prikaže jedna cesta na nekoj mapi grada ili da se prikaže tok neke rijeke.



Slika 6: Primjeri jednostavnog *LineString*-a (a), zatvorenog *LineString*-a kao prstena (b) i zatvorenog *LineString*-a koji nije prsten (c)

2.6.2. Prostorne funkcije

LineString <i>Curve</i>	
+	numPoints() : Integer
+	pointN(n: Integer) : Point

Slika 7: *LineString* (Prema: Open Geospatial Consortium, 2011:23)

LineString je podklasa klase *Curve*, što znači da sve funkcije definirane za *Curve* vrijede i za

LineString. Ovaj prostorni tip podatka također ima svoje vlastite funkcije, a one su (Open Geospatial Consortium, 2011:23):

- *numPoints () : Integer* - broj točaka unutar *LineString* geometrijskog objekta,
- *pointN (n: Integer) : Point* - vraća traženu *n*-tu točku iz *LineString* geometrijskog objekta

2.7. Surface

2.7.1. Opis

Klasa *Surface* (površina) je 2-dimenzionalni geometrijski objekt (Open Geospatial Consortium, 2011:25). Ono je također apstraktna klasa i prema OGC-ovoj specifikaciji (2011:25) ima samo dva djeteta, odnosno dvije podklase: klase *Polygon* i *PolyhedralSurface*. Ovaj rad će se samo fokusirati na klasu *Polygon* jer klasa *PolyhedralSurface* nije implementirana u svim sustavima za upravljanje bazom podataka.

Surface se sastoji od jedne “vanjske granice” i 0 ili više “unutarnjih granica” (Open Geospatial Consortium, 2011:25). Vanjske i unutarnje granice su zapravo zatvorene krivulje (*Curve*). Prema tome, može se reći da se klasa *Surface* sastoji od jedne zatvorene krivulje koja unutar svojih granica može sadržavati 0 ili više drugih zatvorenih krivulja.

2.7.2. Prostorne funkcije

Geometry Surface	
+	<i>area() : Double</i>
+	<i>centroid() : Point</i>
+	<i>pointOnSurface() : Point</i>
+	<i>boundary() : MultiCurve</i>

Slika 8: *Surface* (Prema: Open Geospatial Consortium, 2011:26)

Surface je apstraktna klasa i sve funkcije definirane za ovu klasu koriste djeca ove klase, a te funkcije su sljedeće (Open Geospatial Consortium, 2011:26):

- *area () : Double* - vraća vrijednost površine *Surface* geometrijskog objekta,
- *centroid () : Point* - vraća težište *Surface* geometrijskog objekta,
- *pointOnSurface () : Point* - vraća točku koja se sigurno nalazi na *Surface* geometrijskom objektu,
- *boundary () : MultiCurve* - vraća set krivulja koji čine vanjsku i unutarnju granicu od *Surface* geometrijskog objekta (Open Geospatial Consortium, 2011:25).

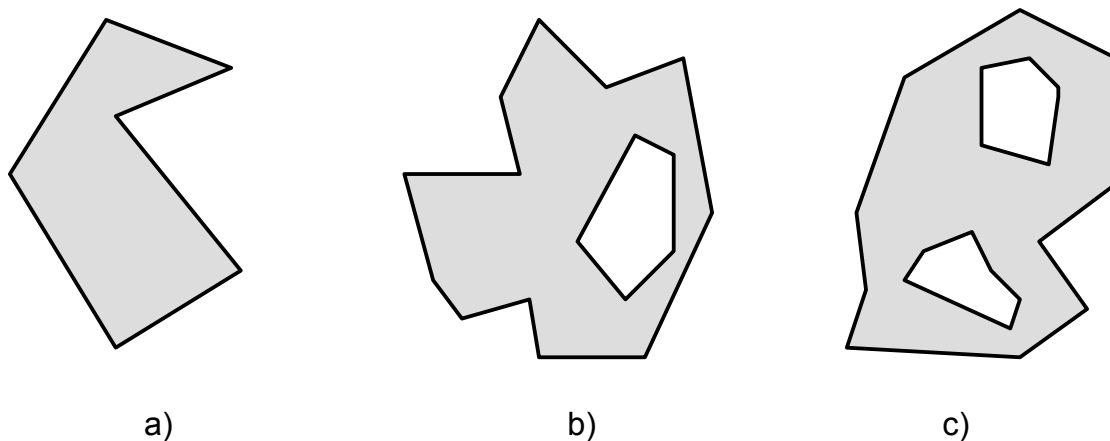
2.8. *Polygon*

2.8.1. Opis

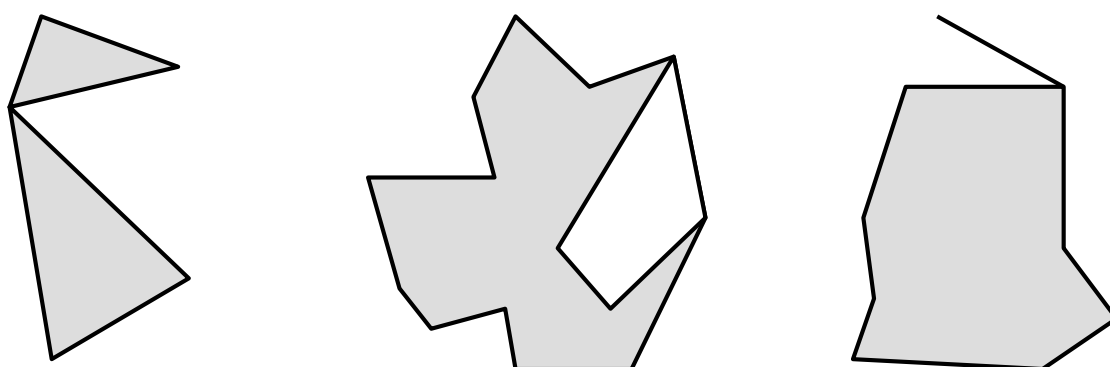
Klasa *Polygon* (poligon) je podklasa od klase *Surface*. Kao što vrijedi za *Surface*, klasu *Polygon* čini jedna vanjska granica i 0 ili više unutarnjih granica. Svaka unutarnja granica se može gledati kao rupa unutar poligona.

Pravila koja definiraju što je valjani *Polygon* su sljedeća (Open Geospatial Consortium, 2011:26-27):

- poligoni su topološki zatvoreni,
- granica poligona se sastoji od skupa zatvorenih krivulja, odnosno prstenja, koji čine unutarnju i vanjsku granicu,
- prsteni se ne smiju presijecati osim u zajedničkoj točki i to samo kao tangenta (tj. prsteni se samo “dodiruju”),
- poligon ne smije imati prekinute ili višak linija,
- unutarnji dio poligona se sastoji od skupa povezanih točaka,
- vanjski dio poligona koji ima jednu ili više rupa nije povezan.



Slika 9: Primjeri poligona s 0 internih prstena (a), 1 internim prstenom (b) i 2 interna prstena (c)



Slika 10: Primjeri nevaljanih poligona

2.8.2. Prostorne funkcije

Polygon		Surface
+ exteriorRing() : LineString + numInteriorRing() : Integer + interiorRingN(n: Integer) : LineString		

Slika 11: Klasa Polygon (Prema: Open Geospatial Consortium, 2011:28)

Polygon je dijete klase *Surface*, što znači da sve funkcije koje su definirane u klasi *Surface* vrijede i za *Polygon*. *Polygon* sadrži 3 dodatne funkcije, a te funkcije su (Open Geospatial Consortium, 2011:28):

- *exteriorRing () : LineString* - vraća vanjsku granicu (tj. prsten) poligona kao

LineString geometrijski objekt,

- *numInteriorRing () : Integer* - vraća broj unutarnjih granica (tj. prstenja) unutar poligona,
- *interiorRingN (n: Integer) : LineString* - vraća *n*-tu unutarnju granicu poligona kao *LineString* geometrijski objekt.

2.9. GeometryCollection

2.9.1. Opis

GeometryCollection je geometrijski objekt koji predstavlja zbirku geometrijskih objekata (Open Geospatial Consortium, 2011:20). Svi elementi unutar *GeometryCollection* se nalaze unutar istog prostornog referentnog sustava, tj. prostorni referentni sustav koji se odabere za *GeometryCollection* vrijedi i za sve elemente te kolekcije (Open Geospatial Consortium, 2011:20).

Osim ograničenja koji se tiče prostornog referentnog sustava, OGC-ova specifikacija (2011:20) navodi da *GeometryCollection* nema nikakvih drugih ograničenja na svoje elemente. No, djeca klase *GeometryCollection* mogu tražiti ograničenja na razini tipa podatka (npr. klasa *MultiPoint* u svoju kolekciju može imati samo elemente tipa *Point*), dimenziji i, riječima OGC-ove specifikacije (2011:20), “može postaviti druga ograničenja na stupnju prostornog preklapanja između elemenata.”

2.9.2. Prostorne funkcije

<i>Geometry</i> GeometryCollection	
+	<i>numGeometries() : Integer</i>
+	<i>geometryN(n: Integer) : Geometry</i>

Slika 12: Klasa *GeometryCollection* (Prema: Open Geospatial Consortium, 2011:20)

GeometryCollection je apstraktna klasa i sve definirane funkcije nasljeđuju djeca te klase. Ono sadrži samo dvije vlastite funkcije (djete je klase *Geometry* pa sadrži i funkcije iz te klase), a one su (Open Geospatial Consortium, 2011:20):

- *numGeometries () : Integer* - vraća ukupan broj geometrijskih objekata unutar kolekcije,
- *geometryN (n: Integer) : Geometry* - vraća *n*-ti geometrijski objekt iz kolekcije.

2.10. *MultiPoint*

2.10.1. Opis

MultiPoint je podklasa od *GeometryCollection* i ono je 0-dimenzionalna kolekcija geometrijskih objekata (Open Geospatial Consortium, 2011:21). Kao što se može pretpostaviti iz naziva ovog prostornog tipa podatka, elementi od *MultiPoint* su striktno točke, odnosno elementi ove kolekcije su tipa *Points*.

OGC (2011:21) u svojoj *Simple Feature Access* specifikaciji navodi da za *MultiPoint* vrijede sljedeće osobine:

- *MultiPoint* je jednostavan ako su sve točke međusobno nejednake,
- svaki *MultiPoint* je prostorno jednak.

2.10.2. Prostorne funkcije

MultiPoint je dijete prostornog tipa podatka *GeometryCollection*. Zasad ne postoje funkcije specifične samo za ovaj prostorni tip podatka.

2.11. *MultiCurve*

2.11.1. Opis

MultiCurve je podklasa od *GeometryCollection* i ono je 1-dimenzionalna kolekcija geometrijskih objekata čiji elementi su tipa *Curve* (Open Geospatial Consortium, 2011:23). *MultiCurve* je prema OGC-ovoj specifikaciji (2011:24) apstraktna klasa i definira samo metode koje će koristiti njegove podklase, a prema objektnom modelu geometrije to je samo klasa *MultiLineString*.

MultiCurve ima sljedeće osobine (Open Geospatial Consortium, 2011:24):

- *MultiCurve* je jednostavan ako su svi njegovi elementi jednostavni (krivulje ne prolaze kroz istu točku više od jednom),
- presjek između elemenata kolekcije se smije dogoditi samo na razini točaka,
- *MultiCurve* je zatvoren ako su svi njegovi elementi zatvoreni (početna točka krivulje je istovremeno i završna točka),
- *MultiCurve* je topološki zatvoren.

2.11.2. Prostorne funkcije

<i>GeometryCollection</i>
MultiCurve
+ isClosed() : Boolean + length() : Double

Slika 13: *MultiCurve* (Prema: Open Geospatial Consortium, 2011:24)

MultiCurve je dijete prostornog tipa podatka *GeometryCollection* i također je apstraktna klasa. OGC (2011:24) u specifikaciji navodi sljedeće dvije funkcije specifične za *MultiCurve*:

- *isClosed () : Boolean* - vraća *TRUE* ako su sve krivulje unutar ove kolekcije zatvorene,
- *length () : Double* - vraća duljinu *MultiCurve* geometrijskog objekta, a to je ekvivalentno duljine svake pojedine krivulje u kolekciji.

2.12. *MultiLineString*

2.12.1. Opis

MultiLineString je podklasa od *MultiCurve* i njegovi elementi su tipa *LineString* (Open Geospatial Consortium, 2011:24). Za *MultiLineString* vrijede iste osobine kao i za *MultiCurve*.

Jedan primjer korištenja *MultiLineString* je za prikaz svih mogućih cesta na karti nekog grada.

2.12.2. Prostorne funkcije

MultiLineString je dijete klase *MultiCurve*, što znači da ono nasljeđuje sve funkcije definirane za *MultiCurve*. Zasad ne postoje funkcije specifično definirane za ovaj prostorni tip podatka.

2.13. *MultiSurface*

2.13.1. Opis

MultiSurface je podklasa od *GeometryCollection* i ono je 2-dimenzionalna kolekcija geometrijskih objekata čiji elementi su tipa *Surface* (Open Geospatial Consortium, 2011:30). Prema OGC-ovoj specifikaciji (2011:30) *MultiSurface* je prostorni tip podatka koji se može instancirati, no MySQL Reference Manual (str. 1287) navodi da se njihova implementacija ne može instancirati već samo njegoa podklasa, *MultiPolygon*.

Za *MultiSurface* vrijede sljedeće osobine (Open Geospatial Consortium, 2011:30):

- interijeri 2 elementa kolekcije (npr. zatvorene krivulje koje se nalaze unutar jedne veće zatvorene krivulje) se ne smiju sijeći,
- granice između dva koplanarna elementa u kolekciji se smiju sjeći u konačnom broju točaka,
- dva *Surface* elementa unutar kolekcije se mogu spojiti u jedan *Surface* element ako im se vanjske granice dodiruju.

2.13.2. Prostorne funkcije

<i>GeometryCollection</i> MultiSurface
+ area() : Double + centroid() : Point + pointOnSurface() : Point

Slika 14: *MultiSurface* (Prema: Open Geospatial Consortium, 2011:31)

MultiSurface je dijete klase *GeometryCollection*. Prema OGC-ovoj specifikaciji (2011:31) *MultiSurface* ima definirane 3 dodatne funkcije, a one su:

- *area () : Double* - vraća površinu *MultiSurface* geometrijskog objekta,
- *centroid () : Point* - vraća centroidnu točku od *MultiSurface* geometrijskog objekta,
- *pointOnSurface () : Point* - vraća točku koja se definitivno nalazi na *MultiSurface* geometrijskom objektu.

Primjetite da su navedene funkcije prema nazivu iste kao i kod klase *Surface*. Jedina razlika je u izvođenju: dok se kod klase *Surface* funkcije izvode na jednoj površini, kod *MultiSurface* funkcije moraju uzeti u obzir sve površine koje se nalaze u toj kolekciji.

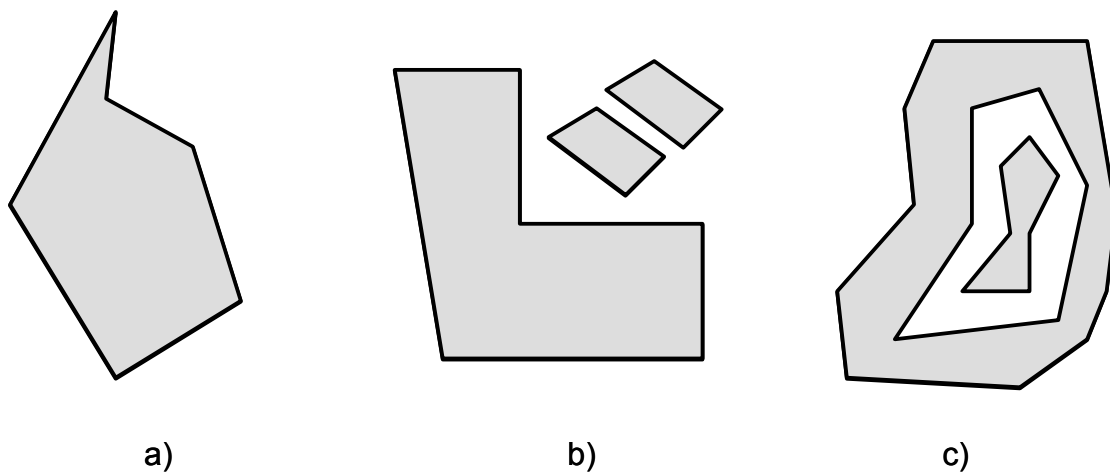
2.14. *MultiPolygon*

2.14.1. Opis

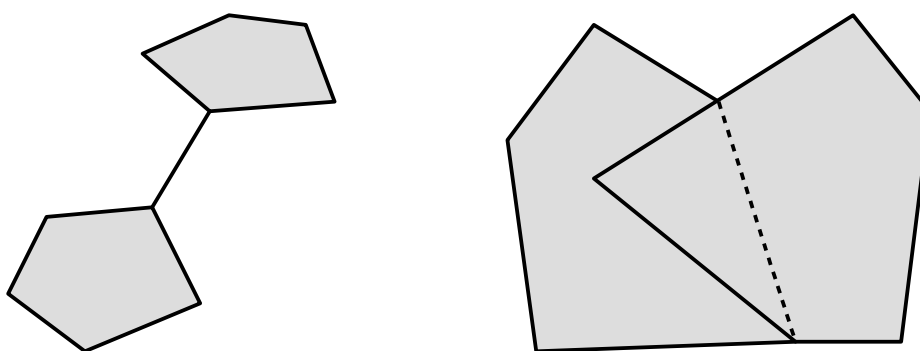
MultiPolygon je podklasa od *MultiSurface*, njegovi elementi su tipa *Polygon* (Open Geospatial Consortium, 2011:31). Za njega vrijede sljedeće osobine (Open Geospatial Consortium, 2011:31-32):

- interijeri bilo koja 2 poligona (*Polygon*) unutar kolekcije se ne smiju sjeći,
- vanjske granice bilo koja 2 poligona se ne smiju sjeći, ali se mogu dodirivati samo u konačnom broju točaka,
- *MultiPolygon* je topološki zatvoren,
- *MultiPolygon* ne smije imati prekinute ili višak linija, već mora biti zatvoren skup točaka,
- broj povezanih komponenti interijera od *MultiPolygon* jednak je broju poligona u kolekciji.

MultiPolygon se može koristiti da se prikažu zgrade i njihov oblik na karti nekog grada.



Slika 15: Primjer MultiPolygon kolekcije sa 1 poligonom (a), 3 poligona (b) i 2 poligona (c)



Slika 16: Primjer nevaljanih MultiPolygon kolekcija

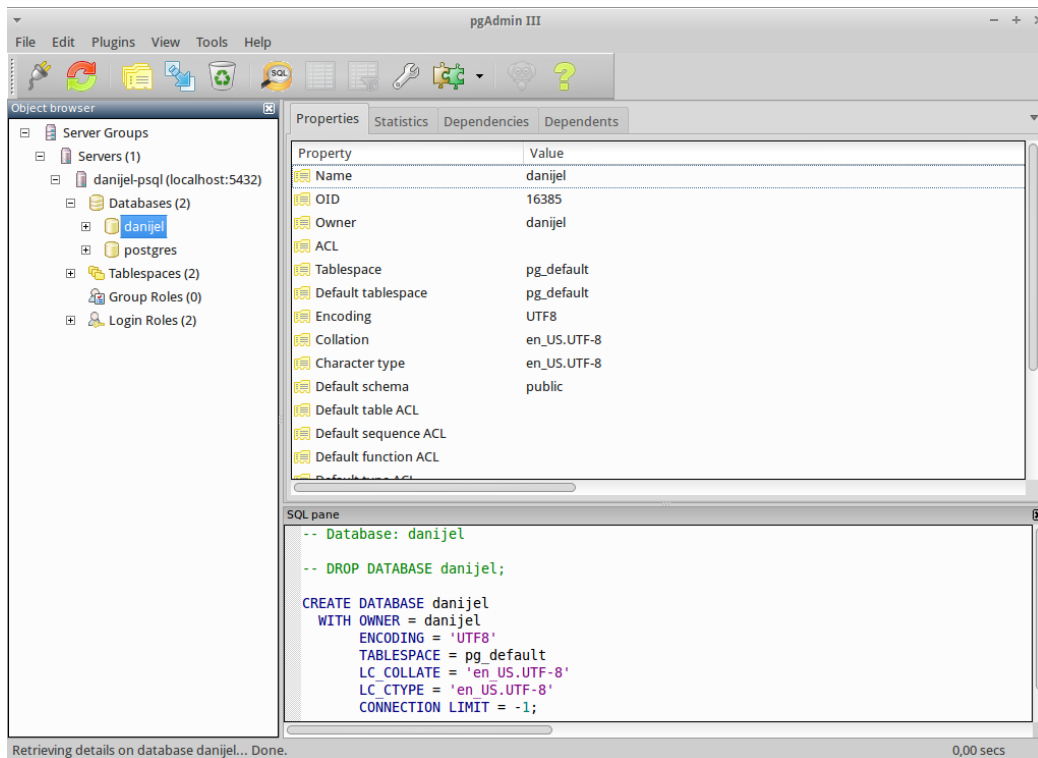
2.14.2. Prostorne funkcije

MultiPolygon je dijete klase *MultiSurface* i nema dodatnih, specifičnih, funkcija koje bi bile definirane samo za ovaj prostorni tip podatka.

3. Praktičan primjer

3.1. Korištena tehnologija

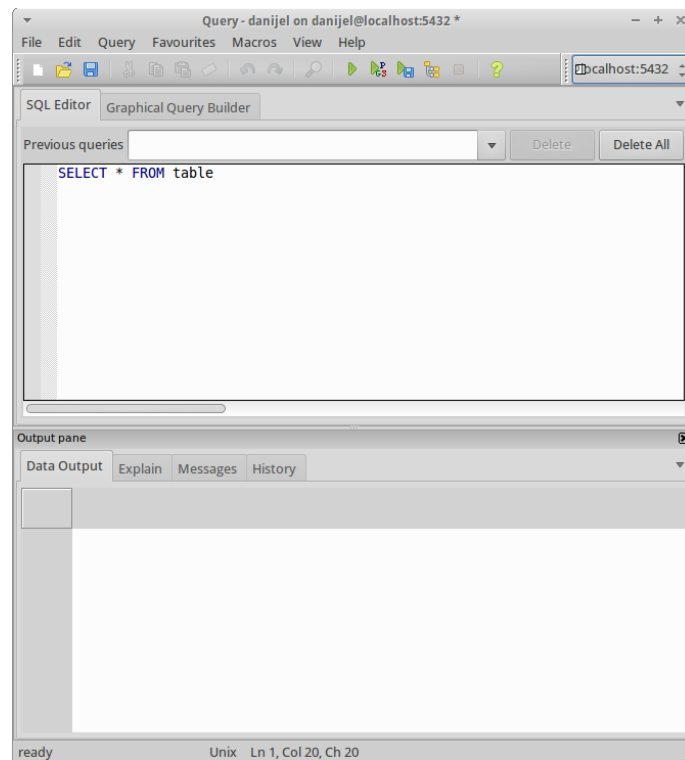
Sljedeći primjeri izrađeni su s PostgreSQL sustavom za upravljanje bazom podataka sa PostGIS dodatkom koji daje potporu za prostorne tipove podataka. Uz taj sustav dolazi i pgAdmin III aplikacija za administraciju baza podataka, a izgled sučelja te aplikacije se može vidjeti na slici 17.



Slika 17: pgAdmin III

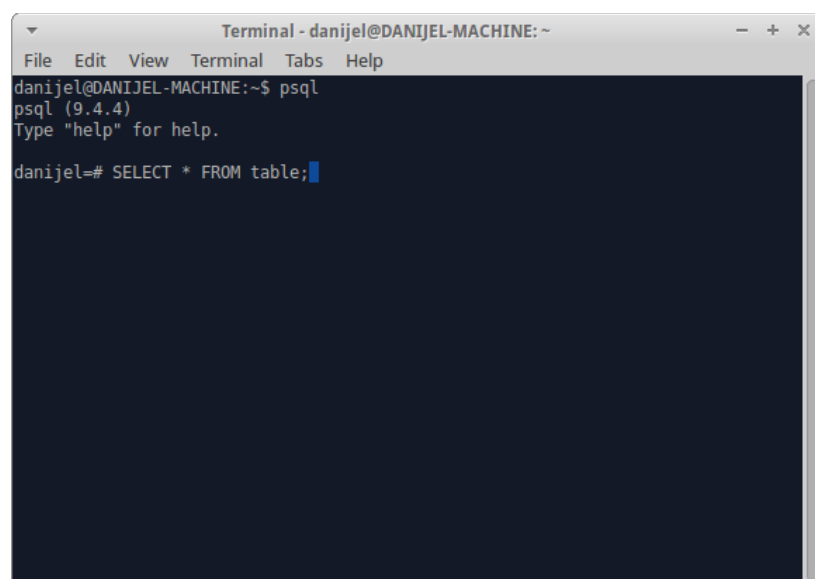
Aplikacija omogućava dodavanje novih korisnika, mijenjanja uloga korisnika, kreiranje novih baza podataka, pregled svih relacijskih tablica u nekoj bazi podataka, itd. Ukratko, pgAdmin III nudi skoro svu funkcionalnost koja je potrebna za administratora baza podataka, bez potrebe ijedne linije SQL koda.

No, to što nema potrebe za pisanjem SQL koda ne znači da pgAdmin III ne nudi i tu mogućnost. Na svaku bazu podataka se mogu izvršiti SQL upiti na način da se označi željena baza podataka u *Object browser*-u te se klikne na *Tools > Query tool* ili koristi kombinacija tipki *Ctrl + E*. Tako će se prikazati prozor za pisanje SQL upita, a ono izgleda kao na slici 18.



Slika 18: Prozor za unos SQL upita

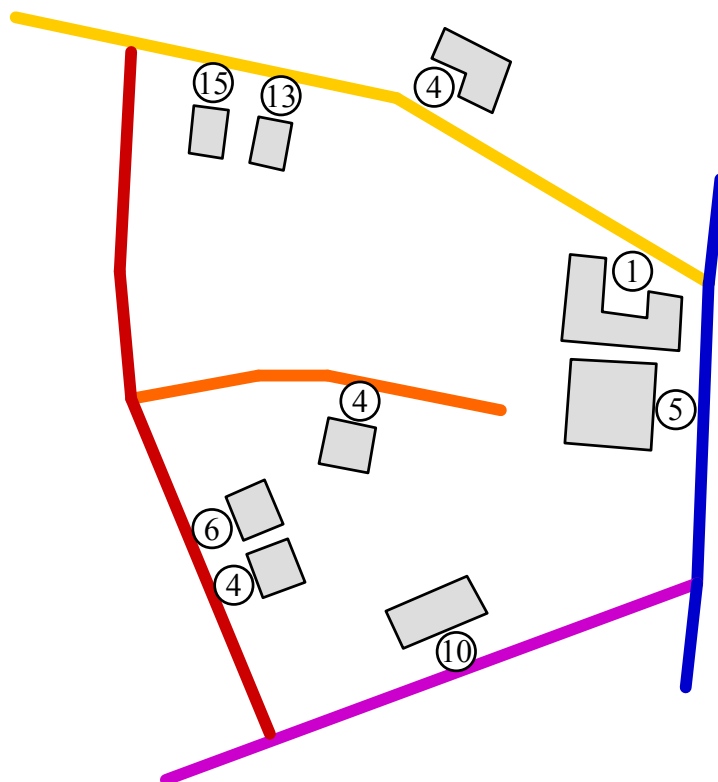
Ako se PostgreSQL sustav za upravljanje bazom podataka instalira na Linux-baziranim operacijskim sustavima, onda se SQL upiti mogu pisati u naredbenoj ljesci (ili terminalu, kako ljubitelji Linux-baziranih operacijskih sustava to vole zvati). To se postiže tako da se u naredbenoj ljesci upiše naredba *psql* i, ako je prethodno konfiguriran korisnik i dana mu je vlastita baza podataka, moći će se izvršavati SQL upiti.



Slika 19: PostgreSQL u naredbenoj ljesci (ili terminalu)

3.2. Primjer

Kao primjer je uzet jedan manji dio grada Novi Marof koji se sastoji od 5 ulica i nekoliko zgrada. Taj dio je prikazan na slici 20.



.Slika 20: Manji dio grada Novi Marof

Svaka ulica je obojana zasebnom bojom, a brojevi predstavljaju kućne brojeve zgrada koji su na slici 20 predstavljeni kao sivi poligoni. Tablica 1 daje imena ulica i koordinate lomnih točaka obojanih linija koje predstavljaju ulice. Koordinate su geografska dužine i širine preuzete sa *Google Maps* servisa.

Tablica 1: Imena ulica po bojama i koordinate njihovih lomnih točaka

Boja	Ime ulice	Koordinate	
		x	y
	Ulica kralja Tomislava	16.333058	46.163945
		16.332404	46.165030
		16.332297	46.165371
		16.332275	46.166077

	Varaždinska ulica	16.334861	46.165641
		16.334840	46.165362
		16.334802	46.164393
		16.334786	46.164140
	Zagorska ulica	16.334802	46.164393
		16.332512	46.163839
	Ulica Matije Gupca	16.334840	46.165362
		16.333509	46.165854
		16.331649	46.166151
	Ulica Augusta Šenoe	16.332404	46.165030
		16.332961	46.165074
		16.333344	46.165036
		16.333854	46.164945

Koordinate lomnih točaka zgrada (sivih poligona na slici 20) su prikazane u tablici 2 uz boju ulice na kojoj se nalazi i kućni broj. Koordinate lomnih točaka su također preuzete za *Google Maps* servisa.

Tablica 2: Koordinate lomnih točaka i kućni broj zgrada, te boja ulice u kojima se nalaze

Boja ulice	Kućni broj	Koordinate	
		x	y
	4	16.332921	46.164401
		16.332893	46.164506
		16.332994	46.164545
		16.333061	46.164436
	6	16.332826	46.164568
		16.332804	46.164680
		16.332910	46.164723
		16.332977	46.164615
	5	16.334264	46.164859
		16.334290	46.165102
		16.334640	46.165086
		16.334605	46.164839
	10	16.333597	46.164215
		16.333540	46.164316
		16.333883	46.164418
		16.333952	46.164316

	1	16.334247	46.165187
		16.334278	46.165384
		16.334429	46.165353
		16.334408	46.165229
		16.334609	46.165217
		16.334614	46.165291
		16.334725	46.165267
		16.334715	46.165147
	4	16.333684	46.165966
		16.333712	46.166019
		16.333980	46.165947
		16.333914	46.165817
		16.333768	46.165858
		16.333796	46.165920
	13	16.332931	46.165726
		16.332971	46.165837
		16.333088	46.165818
		16.333048	46.165711
	15	16.332967	46.165745
		16.332709	46.165857
		16.332831	46.165853
		16.332803	46.165730
	4	16.333245	46.164805
		16.333278	46.164937
		16.333468	46.164909
		16.333409	46.164787

Prije nego se počnu raditi tablice potrebno je prvo u bazi, koja je rezervirana za rad s prostornim tipovima podataka, uključiti PostGIS proširenje. U općenitom programiranju, to bi bilo ekvivalentno uvoženju, odnosno uključivanju, biblioteka u programski kod. Uključivanje proširenja je jednostavno, samo se izvrši sljedeći SQL upit (naravno, preduvjet je da imate PostGIS instaliran):

```
CREATE EXTENSION postgis;
```

Naredba će dodati tablice, poglede (eng. *views*) i funkcije u bazu podataka koje su potrebne za rad s prostornim tipovima podataka.

Sljedeće, kreirat će se tablica ulica koja će se sastojati od sljedećih podataka: ID ulice, ime ulice i prostorni tip podatka koji će predstavljati ulicu. Za geometrijski objekt uzet će se prostorni tip podatka *LineString*. Postoje dva načina kako se može ova kreirati tablica.

```
-- Prvi način
```

```
CREATE TABLE ulice (  
    id SERIAL PRIMARY KEY,  
    ime VARCHAR(100) NOT NULL,  
    geom GEOMETRY(LINESTRING, 4326)  
);
```

```
-- Drugi način
```

```
CREATE TABLE ulice (  
    id SERIAL PRIMARY KEY,  
    ime VARCHAR(100) NOT NULL  
);  
  
SELECT AddGeometryColumn(  
    'ulice',  
    'geom',  
    4326,  
    'LINESTRING',  
    2  
);
```

U prvom načinu dodaje se *LineString* objekt unutar definicije tablice 'ulice'. Funkcija *GEOMETRY()* predstavlja konstruktor koji prima dva parametra: naziv prostornog tipa podatka koji se želi koristiti i SRID (*Spatial Reference ID*). SRID 4326 je 'najopćenitiji' prostorni referentni sustav koji se koristi, pa se zato koristi i u ovom primjeru.

U drugom načinu se prvo kreira tablica 'ulice' bez definiranog prostornog tipa podatka, a poslije se preko *SELECT* upita poziva funkcija *AddGeometryColumn()*. Prvi parametar je naziv tablice u koji će se dodati novi prostorni tip podatka, drugi parametar je naziv atributa/stupca, treći parametar je SRID, četvrti parametar je naziv prostornog tipa podatka koji se želi dodati i zadnji atribut je dimenzija koordinatnog prostora. 2 je za 2-dimenzionalno koordinatni prostor, 3 je za 3-dimenzionalni, itd. Ako se za dimenziju koordinatnog prostora stavi 3, kreirat će se *LineStringZ* objekt, a za vrijednost manju 2 PostgreSQL će javiti grešku (samo za *LineString* prostorni tip podatka).

Dalje, dodaju se redovi u tablicu 'ulice' na temelju tablice 1. Da bi se dodala vrijednost *LineString* objekta korstit će se funkcija *ST_GeomFromText()*.

```
INSERT INTO ulice(ime, geom) VALUES  
    ('Ulica kralja Tomislava',  
    ST_GeomFromText('LINESTRING(16.333058 46.163945,
```

```

16.332404 46.165030,
16.332297 46.165371,
16.332275 46.166077)', 4326)),
('Varaždinska ulica',
 ST_GeomFromText('LINESTRING(16.334861 46.165641,
16.334840 46.165362,
16.334802 46.164393,
16.334786 46.164140)', 4326)),
('Zagorska ulica',
 ST_GeomFromText('LINESTRING(16.334802 46.164393,
16.332512 46.163839)', 4326)),
('Ulica Matije Gupca',
 ST_GeomFromText('LINESTRING(16.334840 46.165362,
16.333509 46.165854,
16.331649 46.166151)', 4326)),
('Ulica Augusta Šenoje',
 ST_GeomFromText('LINESTRING(16.332404 46.165030,
16.332961 46.165074,
16.333344 46.165036,
16.333854 46.164945)', 4326));

```

Funkcija *ST_GeomFromText()* prima dva parametra. Prvi parametar je tekst u kojem definiramo željeni prostorni tip podatka i koordinate točaka od kojih će se sastojati taj tip podatka. Prvi parametar mora biti pisan u sljedećem formatu (ovo ne vrijedi za prostorni tip podatka *Polygon*):

```
NazivProstornogTipa(x1 y1 [z1 m1], x2 y2 [z2 m2], ...)
```

Ako se radi o prostornom tipu podatka *Polygon*, format je drugačiji:

```
POLYGON((xv1 yv1 [zv1 mv1], ...), (xu1 yu1 [zu1 mu1], ...))
```

Prvi parametar u gornjem parametru su koordinate točaka vanjske granice poligona, a drugi parametar su koordinate točaka unutarnje granice poligona.

Prostorni tipovi podataka se u tablici zapisuje u heksadekadskom formatu, što znači da upit poput *'SELECT * from ulice'* nama ljudima ne bi previše pomogao ako želimo vidjeti vrijednosti koje su unesene u tablicu. Zato se koristi funkcija *ST_AsText()* koja heksadekadski zapis pretvara u, ljudima čitljiviji, tekstualni zapis. Ta funkcija kao parametar prima naziv stupca unutar tablice. Koristi se na sljedeći način:

```
SELECT id, ime, ST_AsText(geom) FROM ulice;
```

Rezultat tog upita je, u ovom slučaju, veoma širok, pa se neće pokazati u ovom primjeru.

Sljedeće što treba napraviti je dodati građevinske objekt, tj. zgrade, u bazu podataka. Tablica za zgrade bi se sastojala od sljedećih atributa: ID, ulica, kućni broj i prostorni tip podatka koji će predstavljati zgradu. Atribut za ulice će biti vanjski ključ na tablicu 'ulice'. Za prostorni tip podatka koji će predstavljati zgradu koristiti će se *Polygon*.

```
CREATE TABLE zgrade (  
    id SERIAL PRIMARY KEY,  
    ulica INTEGER REFERENCES ulice(id) NOT NULL,  
    broj INTEGER NOT NULL,  
    geom GEOMETRY(POLYGON, 4326)  
);
```

Na temelju tablice 2 (str. 25) mogu se unositi zgrade u tablicu. Kod vrijednosti za prostorni tip podatka *Polygon* treba početnu koordinatu lomne točke unijeti dvaput: jednom kao početna koordinata i drugi put kao završna koordinata. Razlog tome je što *Polygon* mora biti zatvorena krivulja, što znači da početna lomna točka mora biti jednaka krajnjoj lomnoj točki krivulje.

```
INSERT INTO zgrade(ulica, broj, geom) VALUES  
    (1, 6,  
     ST_GeomFromText('POLYGON((16.332826 46.164568,  
                                16.332804 46.164680,  
                                16.332910 46.164723,  
                                16.332977 46.164615,  
                                16.332826 46.164568))', 4326)),  
    (2, 5,  
     ST_GeomFromText('POLYGON((16.334264 46.164859,  
                                16.334290 46.165102,  
                                16.334640 46.165086,  
                                16.334605 46.164839,  
                                16.334264 46.164859))', 4326)),  
    (3, 10,  
     ST_GeomFromText('POLYGON((16.333597 46.164215,  
                                16.333540 46.164316,  
                                16.333883 46.164418,
```

```

16.333952 46.164316,
16.333597 46.164215))', 4326)),
(4, 1,
  ST_GeomFromText('POLYGON((16.334247 46.165187,
16.334278 46.165384,
16.334429 46.165353,
16.334408 46.165229,
16.334609 46.165217,
16.334614 46.165291,
16.334725 46.165267,
16.334715 46.165147,
16.334247 46.165187))', 4326)),
(4, 4,
  ST_GeomFromText('POLYGON((16.333684 46.165966,
16.333712 46.166019,
16.333980 46.165947,
16.333914 46.165817,
16.333768 46.165858,
16.333796 46.165920,
16.333684 46.165966))', 4326)),
(4, 13,
  ST_GeomFromText('POLYGON((16.332931 46.165726,
16.332971 46.165837,
16.333088 46.165818,
16.333048 46.165711,
16.332931 46.165726))', 4326)),
(4, 15,
  ST_GeomFromText('POLYGON((16.332967 46.165745,
16.332709 46.165857,
16.332931 46.165853,
16.332803 46.165730,
16.332967 46.165745))', 4326)),
(5, 4,
  ST_GeomFromText('POLYGON((16.333245 46.164805,
16.333278 46.164937,
16.333468 46.164909,
16.333409 46.164787,
16.333245 46.164805))', 4326));

```

Literatura

- [1] Open Geospatial Consortium (2011) *OpenGIS Implementation Specification for Geographic Information – Simple feature access – Part 1: Common architecture*. Preuzeto 6. lipnja 2015. s <http://www.opengeospatial.org/standards/sfa>
- [2] MySQL Development Team (s.a.) *MySQL 5.6 Reference Manual*. Preuzeto 9. lipnja 2015. s <http://downloads.mysql.com/docs/refman-5.6-en.pdf>
- [3] PostGIS (s.a.) *PostGIS 2.1.8dev Manual - SVN Revision (13357)*. Preuzeto 20. lipnja 2015. s <http://postgis.net/stuff/postgis-2.1.pdf>