

Hugo Majer

1. N-GKDPZ

Praha, 26. 11. 2021



Semestrální práce z předmětu Open GIS

Práce s vektorovými daty v rámci PostgreSQL/PostGIS

Univerzita Karlova

Přírodovědecká fakulta

1. Motivácia a teoretický úvod

Motiváciou k výberu témy bola moja bakalárska práca, ktorej náplňou bola tvorba priestorovej databázy prostredníctvom PostgreSQL/PostGIS. Pri tvorbe databázy bakalárska práca aj zostala a so samotnou funkcionalitou PostGISu sa nepracovalo. Z tohto dôvodu je cieľom tejto semestrálnej práce zoznámenie sa so základnými (priestorovými) funkciami PostGIS, ktoré sa týkajú vektorových dát a zároveň si práca kladie za cieľ vybrané funkcie aj prakticky vyskúšať zostavením SQL príkazov riešiacich konkrétne vytvorené modelové úlohy.

Prvá kapitola semestrálnej práce v krátkosti predstavuje samotný PostgreSQL, následne jeho priestorový extender PostGIS. Keďže zámerom práce je práca s vektorovými dátami, teoretická časť práce predstavuje dostupné vektorové priestorové dátové typy a geometrické typy v rámci PostGIS. Koniec kapitoly sa v krátkosti venuje možnostiam vkladu vektorových dát do priestorovej databázy PostGIS, od ktorých sa odráža aplikačná časť semestrálnej práce.

1.1. PostgreSQL/PostGIS

PostgreSQL je voľne dostupný objektovo-relačný databázový systém, ktorý využíva a zároveň rozširuje jazyk SQL. Funkcionalita PostgreSQL umožňuje bezpečné ukladanie obrovského množstva dát rôznych typov. Jeho začiatky siahajú do 80. rokov minulého storočia, kedy bol ešte prezentovaný pod názvom *Postgres*. Viac ako 30 rokov je aktívne vyvíjaný. Je považovaný za najlepší a najdokonalejší voľne dostupný systém najmä vďaka osvedčenej architektúre, spoľahlivosti, bezpečnosti, integrite dát, veľkému množstvu dostupných funkcií, neustálym aktualizáciám zdokonaľujúcim systém a vďaka jeho možnostiam rozširiteľnosti. Jedna z možností rozšírenia PostgreSQL sú tzv. *extendry*. Najpopulárnejším extendrom je priestorové rozšírenie PostGIS (PostgreSQL, 2021).

PostGIS je voľne dostupné priestorové rozšírenie pre PostgreSQL. Pomocou nových dátových typov umožňuje jednak uchovávať priestorové objekty v databáze, ale aj podrobovať ich analýze a spracovaniu v prostredí SQL (PostGIS, 2021). PostGIS teda umožňuje vytvárať *priestorové databázy*, ktoré sa od „bežných“ databáz líšia nasledujúcimi vlastnosťami a možnosťami (PostGIS, 2012):

- ❖ Priestorové dátové typy,
- ❖ Priestorové indexovanie,
- ❖ Priestorové funkcie.

Z toho vyplýva, že PostGIS, podobne ako iné priestorové databázy, kombinuje vlastnosti GIS s výhodami databázových systémov - ako príklad uved'me transakčné spracovanie, paralelný prístup alebo indexovanie (Strobl, 2008).

1.2. Vektorové priestorové dátové typy PostGIS

PostGIS sa riadi a je plne vyhovujúci štandardu konzorcia *Open Geospatial Consortium* (OGC) s názvom *Simple Feature Access* (SFA) (PostGIS, 2021), ktorý definuje priestorový dátový typ *Geometry* pre vektorové dáta. Štandard teda definuje, akými spôsobmi reprezentovať objekty reálneho sveta a zároveň ako implementovať tento priestorový dátový typ do SQL.

Okrem priestorového dátového typu *Geometry* je pre vektorové dáta dostupný aj dátový typ *PostGIS Geography*, ktorý pracuje so sférickým tvarom Zeme, pričom dátový typ *Geometry* pracuje v rovine. Pre tento dátový typ je momentálne dostupné menšie množstvo funkcií, ich vykonanie trvá dlhšiu dobu, čo je dané komplikovanejšími výpočtami na sfére (PostGIS, 2021).

Vektorová reprezentácia reálnych objektov je možná štandardnými geometrickými typmi. PostGIS umožňuje dva spôsoby ako tieto geometrické typy vyjadrovať. Prvou je reprezentácia *Well-Known Text* (WKT), druhou je *Well-Known Binary* (WKB), pričom obidve obsahujú informáciu o geometrickom type objektu a súradnice, ktoré daný objekt tvoria (PostGIS, 2021). Na príklade nižšie je ukážka WKT reprezentácie jednotlivých geometrických typov:

```
POINT(0 0)
MULTIPOINT((0 0),(1 2))
LINESTRING(0 0,1 1,1 2)
MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))
MULTIPOLYGON(((0 0,4 0,4 0,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))
```

PostGIS nasleduje aj druhú normu s názvom *SQL Multimedia Applications Spatial* od *International Organization for Standardization*, čím je rozšírený o radu kruhových interpolačných kriviek (PostGIS, 2021):

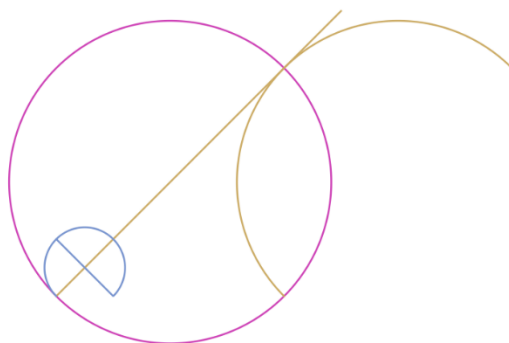
```
CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0)
COMPOUNDCURVE(CIRCULARSTRING(0 0, 1 1, 1 0),(1 0, 0 1))
MULTICURVE((0 0, 5 5),CIRCULARSTRING(4 0, 4 4, 8 4))
```

CIRCULARSTRING je základná krivka, jeden jej segment vyžaduje definíciu troch bodov – začiatočného, bodu na oblúku a koncového.

COMPOUNDCURVE je spojitá krivka, ktorá má aj zakrivené a aj lineárne segmenty.

MULTICURVE je súbor kriviek, ktorý môže obsahovať lineárne, zakrivené a zložené segmenty.

Vizualizácia vyššie uvedených kriviek je na Obrázku 1, na ktorom fialovou farbou je CIRCULARSTRING, modrou COMPOUNDCURVE a žltou MULTICURVE.



Obrázok 1: Vizualizácia vybraných kriviek dostupných v PostGIS.

Fialová – CIRCULARSTRING; Modrá – COMPOUNDCURVE; Žltá – MULTICURVE.

Zdroj: autor.

Okrem vektorových dát PostGIS plne podporuje aj spracovanie a analýzu rastrových dát, 3D modelov povrchu a dostupný je aj modul pre topológiu (PostGIS, 2021).

1.3. Vklad vektorových dát do databázy

Vkladanie vektorových dát do priestorovej databázy, resp. priestorovej tabuľky je momentálne možný dvoma spôsobmi. Prvý spôsob je pomocou SQL výrazu. Tento spôsob sa dá využiť, ak máme dáta v textovej reprezentácii (PostGIS, 2021). Za pomoci reprezentácie WKT, obdobným spôsobom uvedeným v predošlej kapitole, vkladáme priestorové objekty do tabuľky:

```

CREATE TABLE cesty
    (id INTEGER, geom GEOMETRY, name VARCHAR);
INSERT INTO cesty VALUES
    (1, 'LINESTRING(191232 243118,191108 243242)', 'Jeff Rd'),
    (2, 'LINESTRING(189141 244158,189265 244817)', 'Geordie Rd');

```

PostGIS umožňuje vklad priestorových dát do databázy aj v ESRI formáte SHP a to vďaka nástroju *shp2pgsql*, ktorý konvertuje SHP na SQL príkaz vhodný na vklad dát do *Geometry* formátu. Nástroj je dostupný vo forme príkazového riadku, ale od určitej verzie PostGIS má aj grafické rozhranie (PostGIS, 2021).

2. Aplikačná časť

Druhá kapitola práce má za cieľ prakticky ukázať a aplikovať vybrané základné priestorové funkcie PostGIS. Začína popisom vkladania dát do priestorovej databázy, na ktorých budú jednotlivé funkcie demonštrované a postupuje k ukážkam vybraných funkcií vykonávaných prostredníctvom SQL príkazov dotazujúcich sa prostredníctvom QGIS modulu DB Manager do priestorovej databázy.

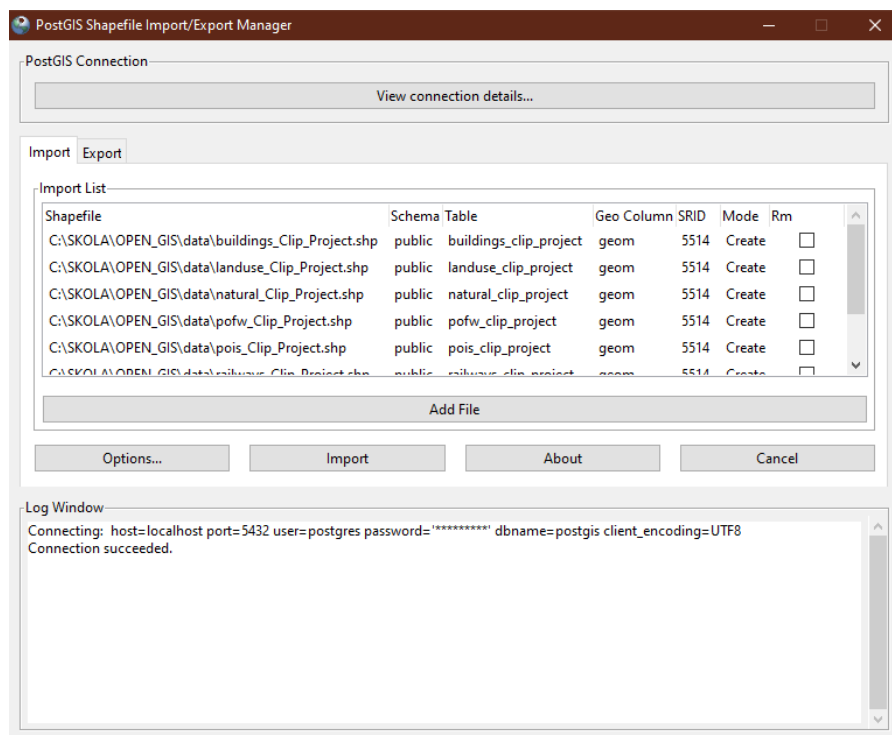
2.1. Použité dáta a ich vloženie do databázy

Dátami použitými v tejto práci sú voľne dostupné *OpenStreetMap* dáta. Konkrétne sa jedná o vrstvy ciest, železníc, vodných tokov, budov, landuse, prírodných prvkov, pietnych miest a bodov záujmu. Vrstvy pôvodne pokrývali celú Českú republiku, ale v GIS softvéri boli orezané len na kraj *Hlavní město Praha* a to za účelom rýchlejšieho importu do databázy, menšej výpočtovej náročnosti pri práci s nimi a s tým súvisiacim ušetrením času. Z ArcČR500 bola do databázy vložená vrstva mestských častí, opäť orezaná len na Prahu.

Na vklad týchto vrstiev do PostgreSQL/PostGIS priestorovej databázy bolo využité grafické rozhranie nástroja *shp2pgsql*, spomenuté v predošlej kapitole. Pred samotným vkladom je nutné sa k danej databáze pripojiť a následne zvoliť, ktoré SHP súbory chceme vkladať do databázy. Podstatné je pri každom SHP v zozname upraviť hodnotu SRID (súradnicový referenčný systém) a to z predvolenej hodnoty 0 na v tomto prípade 5514, čo je SRID pre *S-JTSK Krovak NorthEast* v ktorom sú vkladané SHP (pozri Obrázok 2).

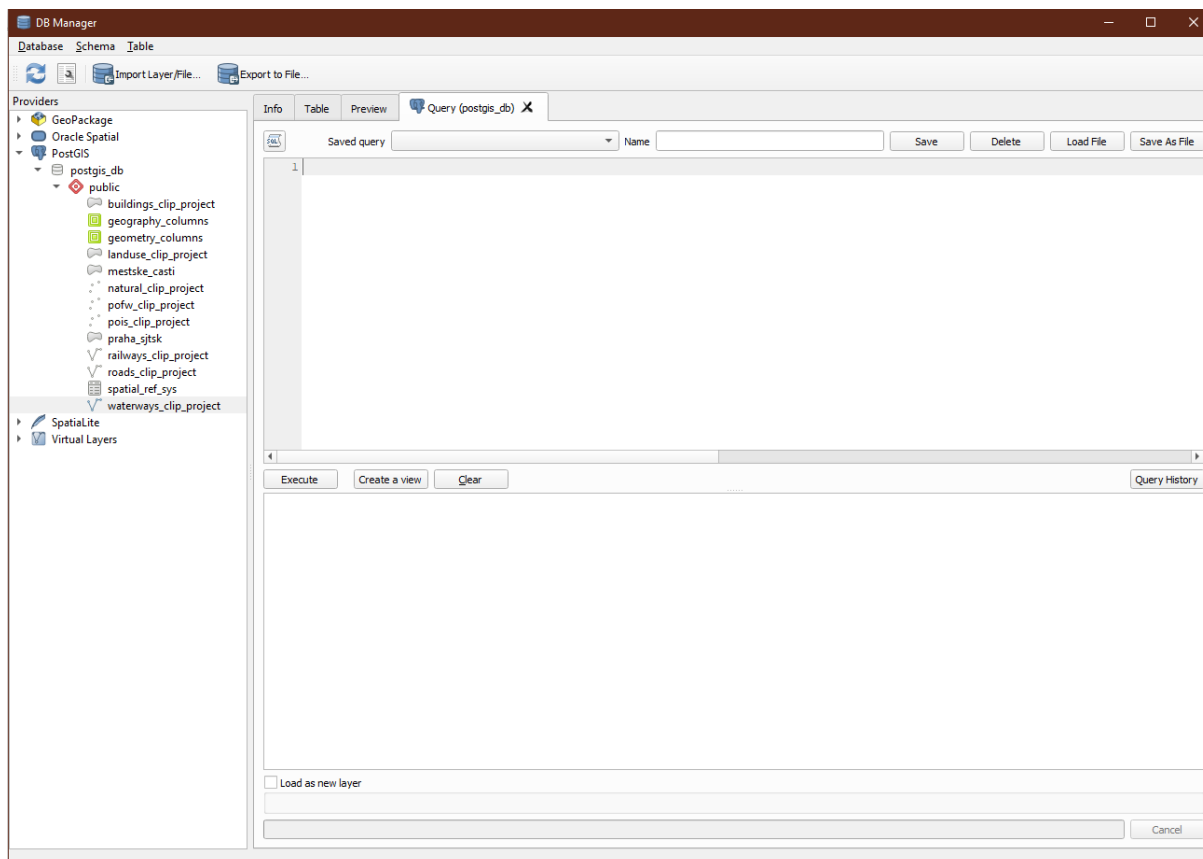
2.2. Pripojenie k databáze prostredníctvom QGIS

V QGIS je zabudovaný nástroj *DB Manager*, ktorý ponúka podobnú funkcionality ako *pgAdmin* – správu a manažment PostgreSQL databázy. Pri jeho spustení je potrebné sa najprv pripojiť k danej databáze. Po pripojení sa daná databáza objaví v ľavom zozname a to spoločne so zoznamom jej tabuliek, ktoré v našom prípade tvoria jednotlivé načítané vrstvy. Po kliknutí na tlačidlo *SQL Window* sa v pravej časti okna objaví konzola, ktorá slúži na tvorbu a vykonanie SQL príkazov, ktoré sa dotazujú na dáta uložené v pripojenej databáze (pozri Obrázok 3).



Obrázok 2: Vklad SHP súborov OpenStreetMap pomocou grafického rozhrania nástroja shp2pgsql.

Zdroj: autor.



Obrázok 3: Nástroj DB Manager v QGIS slúžiaci na správu a manažment pripojených priestorových databáz.

Zdroj: autor.

2.3. Práca s dátami v databáze – aplikácia funkcií PostGIS

Opisovanú funkciu *SQL Window* v *DB Manager* je možné využiť na vykonanie jednoduchých ale aj komplexných SQL príkazov. Za ako úplne najzákladnejšiu SQL požiadavku môžeme označiť vizualizáciu vrstiev uložených v databáze. To sa vykonáva nasledujúcim SELECT príkazom:

```
SELECT * FROM railways_clip_project;
```

Následne sa s danou vrstvou dá v QGIS pracovať ako s bežnou vrstvou uloženou na disku počítača, tzn. je možné ju zahrnúť do ako vstup do funkcií, operácií, exportovať, upravovať symbológiu atď. V takomto prípade je ale vykonanie operácií postavené na QGIS a použitom počítači. Naopak, ak v QGIS využívame *DB Manager*, tak požadované SQL dotazy jednotlivých operácií prebiehajú na strane PostGIS a na strane databázy. Navyše ak databáza „beží“ na servery (čo v prípade tejto semestrálnej práce neplatí), k jej dátam pristupujeme vzdialene – nie sú uložené na disku počítača.

Funkcie, ktoré sú predmetom záujmu tejto práce, oficiálna dokumentácia PostGIS delí na *výpočtové funkcie* - počítajú vzdialenosť, rozlohu a uhly; *priestorové funkcie* (nazývané aj *topologickými funkciami*) – vyšetrujú vzťah medzi geometriami, teda pracujú s viacerými geometriami v jeden čas. Pre úplnosť dodajme, že v PostGIS existuje viacero typov funkcií a tým pádom aj viacnásobne väčšie množstvo funkcií ako je spomenutých v tejto práci. Takisto v práci bude ukázaný len výber z funkcií patriacich do vyššie uvedených kategórií funkcií.

2.3.1. Výpočtové funkcie

Ako základnú výpočtovú funkciu môžeme označiť funkciu *ST_Length*, ktorá vracia dĺžku danej línie v jednotkách súradnicového systému. Príkladom využitia tejto funkcie v použitých dátach tejto práce môže byť úloha zistiť celkovú dĺžku pražského metra:

```
SELECT SUM(ST_Length(geom))/2 FROM railways_clip_project  
WHERE fclass='subway' AND name LIKE 'Metr%';
```

Výsledkom je 66,1 km čo viac-menej odpovedá oficiálnym údajom.

Druhou predstavenou funkciou môže byť *ST_Area*, ktorá vracia rozlohu polygónu. Ukážka aplikácie funkcie môže byť na úlohe zistiť akú percentuálnu časť územia Prahy pokrývajú parky:


```
SELECT (y/x)*100 AS podiel FROM (SELECT ST_Area(geom)/1000000 AS x
FROM praha_sjtsk) AS a, (SELECT SUM(ST_Area(geom))/1000000 AS y
FROM pois_polyg WHERE fclass='park') AS b;
```

Parky pokrývajú 2,29 % územia Prahy.

Na výpočet vzdušnej vzdialenosti medzi dvoma geometriami slúži funkcia *ST_Distance*. Aplikáciou funkcie môžeme zistiť napr. vzdušnú vzdialenosť medzi budovou Přírodovědecké Fakulty Univerzity Karlovy na Albertove a stanicou metra I. P. Pavlova:

```
SELECT ST_Distance(y,x) FROM (SELECT geom AS y FROM
buildings_clip_project WHERE name='Přírodovědecká fakulta Univerzity
Karlovy') AS a, (SELECT geom AS x FROM transport_points WHERE
fclass='railway_station' AND name LIKE '%Pavlova') AS b;
```

Vzdušná vzdialenosť týchto dvoch miest je cca 703 m.

Funkciu *ST_Distance* môžeme v SQL dotaze zameniť za funkciu *ST_ShortestLine*, ktorá nevracia vzdušnú vzdialenosť len ako číslo, ale vráti ju ako geometriu – líniu (pozri Obrázok 4).



Obrázok 4: Výsledná línia vrátená funkciou *ST_Distance* predstavujúca vzdušnú čiaru medzi PĚF UK na Albertove a stanicou metra I. P. Pavlova

Zdroj: autor.

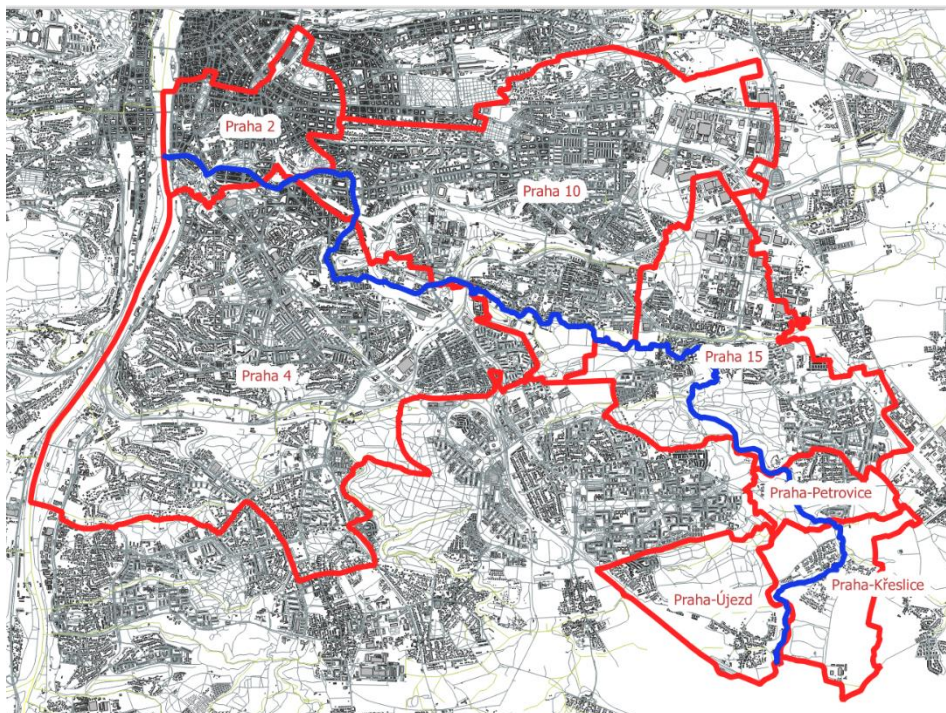
Okrem spomenutých funkcií z tejto kategórie funkcií stoja za zmienku aj funkcie *ST_Perimeter*, ktorá vracia obvod polygónu; *ST_Centroid*, ktorá vráti stred polygónu. Pri charakteristike *výpočtových funkcií* bolo spomenuté, že existujú funkcie, ktoré vracajú uhly. Jedná sa o funkciu *ST_Angle* – vracia uhol medzi tromi bodmi alebo dvoma vektormi (štyri body alebo dve línie), a o funkciu *ST_Azimuth*, ktorá vracia azimut v radiánoch medzi dvoma bodmi (PostGIS, 2021).

2.3.2. Priestorové funkcie

Priestorová funkcia, ktorá testuje pretnutie dvoch geometrií (resp. dochádza k ich priestorovému prieniku) a ktorá nesmie chýbať v žiadnom GIS nástroji je *ST_Intersects*. Jej SQL použitie v rámci PostGIS môžeme ukázať na celi zistiť, ktorými pražskými mestskými časťami preteká Botič:

```
SELECT naz_moamc,geom FROM mestske_casti WHERE  
ST_Intersects(geom,(SELECT ST_Union(geom) FROM waterways_clip_project  
WHERE name='Botič'));
```

V rámci tejto úlohy bolo nutné použiť aj jednu z *agregačných funkcií* a to síce *ST_Union*, nakoľko Botič nie je v OpenStreetMap reprezentovaný jednou súvislou líniou, ale viacerými a pre úspešné vykonanie dotazu bolo nutné línie predstavujúce Botič zlúčiť do jednej geometrie (spojitej línie). Výsledok operácie je na Obrázku 5. Obdobou *ST_Intersect* je funkcia *ST_Overlaps*, ktorá ale vracia pravdivú hodnotu len vtedy, ak prienik je rovnakého geometrického typu ako vstupné prvky, tj. prienikom plôch je plocha, prienikom línií je línia atď. Ako opak funkcie *ST_Intersects* sa dá označiť funkcia *ST_Disjoint*, ktorá vracia prvky priestorovo rôzne.



Obrázok 5: Mestské časti Prahy, ktorými preteká Botič získané za pomoci funkcie *ST_Intersects*.

Zdroj: autor.

Ďalšou funkciou s bohatou využiteľnosťou môže byť *ST_DWithin*, ktorá testuje, či sú dva objekty od seba vzdialené určitou zadanou vzdialenosťou (vzdušnou čiarou). Aplikovať ju môžeme napríklad na úlohu nájsť hotely do vzdialenosti 1 km od pražského Hlavního nádraží:

```
SELECT a.* FROM pois_clip_project AS a WHERE ST_DWithin(a.geom,
(SELECT b.geom FROM transport_points AS b WHERE name='Praha hlavní
nádraží'),1000) AND a.fclass='hotel';
```

Príkaz vyššie vráti vyhovujúce hotely vo forme bodov (pretože vrstva bodov záujmu z OpenStreetMap je bodová) ale pre zaujímavejšiu vizualizáciu výsledkov môžeme SQL príkaz doplniť a dostať vrstvu budov vyhovujúcich hotelov, teda polygónovú vrstvu (pozri Obrázok 6):

```
SELECT build.* FROM buildings_clip_project AS build WHERE ST_Intersects
(build.geom, (SELECT ST_UNION(a.geom) FROM pois_clip_project AS a
WHERE ST_DWithin(a.geom,(SELECT b.geom FROM transport_points AS b WHERE
name='Praha hlavní nádraží'),1000) AND a.fclass='hotel'));
```

Aby sme ukázali aj ďalšiu funkciu – *ST_Buffer*, ktorá tvorí okolo prvku buffer s určitou zadanou veľkosťou, tak môžeme výsledok vyššie uvedeného SQL príkazu overiť vytvorením kilometrového bufferu okolo Hlavního nádraží (pozri Obrázok 6):

```
SELECT ST_Buffer((SELECT b.geom FROM transport_points AS b WHERE
name='Praha hlavní nádraží'),1000);
```



Obrázok 6: Budovy hotelov (tmavozelené objekty), ktoré sa nachádzajú do 1 km od Hlavného nádraží vrátené funkciou *ST_DWithin*. Červená prerušovaná línia značí 1 km buffer okolo Hlavného nádraží, ktorý vznikol funkciou *ST_Buffer*.

Zdroj: autor.

Okrem už spomenutých *priestorových funkcií* PostGIS ponúka aj funkcie *ST_Contains* a *ST_Within*, ktoré testujú, či jedna geometria je *plne* obsiahnutá v druhej geometrii. *ST_Equals* zase testuje priestorovú zhodu geometrií. Funkcia *ST_Touches* vyšetruje, či sa dve geometrie dotýkajú na svojich hraniciach a ich vnútra sa nepretínajú.

2.3.3. Priestorové spojenie tabuliek (*Spatial Join*)

JOIN je SQL funkcia, ktorá spája dve alebo viacero tabuliek do jednej virtuálnej a to na základe nejakého spoločného stĺpca. PostGIS umožňuje tabuľky (vrstvy) spojiť na základe polohy, resp. priestorového vzťahu a to tak, že za (JOIN) ON sa použije jedna z *priestorových funkcií*, ktorá testuje priestorový vzťah dvoch geometrií. SQL príkaz uvedený nižšie vracia polygóny mestských častí, v ktorých sa nachádzajú konečné stanice pražského metra, vykonaný pomocou JOIN a za priestorovej podmienky, že bod konečnej zastávky sa *plne* nachádza v polygóne určitej mestskej časti (*ST_Contains*):


```
SELECT casti.* FROM mestske_casti AS casti JOIN (SELECT * FROM
transport_points AS trans WHERE fclass='railway_station' AND name IN
('Letňany','Černý Most','Háje','Zličín','Nemocnice Motol','Depo Hostivař'))
AS m ON ST_Contains(casti.geom,m.geom)
```

Príklad uvedený vyššie nie je najlepším príkladom *Spatial Join*, pretože daný SQL dotaz by sa dal napísať aj bez využitia JOIN, teda tak ako predošlé SQL dotazy v predchádzajúcich príkladoch a úlohách. Efektivita *Spatial Join* by sa ukázala vtedy, ak by sme mali samostatnú bodovú vrstvu konečných staníc metra (tzn. nemuseli by sme stanice selektovať z bodovej vrstvy *transport_points* ako v príkaze vyššie). SQL príkaz by v takomto prípade vyzeral takto:

```
SELECT casti.* FROM mestske_casti AS casti JOIN konecne_stanice AS ks
ON ST_Contains(casti.geom,ks.geom);
```

Pre porovnanie ten istý dotaz, ale bez využitia *Spatial Join* v príkaze:

```
SELECT casti.* FROM mestske_casti AS casti WHERE
ST_Contains(casti.geom,(SELECT ks.geom FROM konecne_stanice AS ks));
```

Vidíme, že musel byť použitý ďalší (vnorený) SELECT, od ktorého nás *Spatial Join* ušetril.

3. Záver

Ako bolo v priebehu práce spomenuté, voľne dostupné priestorové rozšírenie PostGIS ponúka funkcionality GIS pre databázový systém PostgreSQL. Do priestorových databáz umožňuje ukladať a následne analyzovať nielen vektorové dáta, ale aj rastrové. V prípade vektorových dát to koniec-koncov bolo dokázané aj v rámci tejto práce. Všetky, v rámci tejto práce, vykonané operácie a analýzy je možné vykonať v GIS softvéri s jeho užívateľským rozhraním, ale všetky operácie a funkcie, ktoré PostGIS ponúka môžeme napríklad implementovať do PostgreSQL databázy a to formou vnorených procedúr a dané procesy s dátami v databázy takto zautomatizovať. Takýmto spôsobom môžeme získať napríklad automatizované generovanie výsledkov implementovaných analýz nad dátami nahrávanými do databázy alebo implementované operácie sa môžu týkať nejakých úprav priestorových dát v databáze a s takto automaticky upravenými dátami na strane databázy môžeme následne pracovať v GIS softvéri.

4. Použité zdroje

POSTGIS 2012. Workshop: Introduction to PostGIS [online]. Dostupné z: <https://postgis.net/workshops/postgis-intro/index.html> [cit. 27. 11. 2021].

POSTGIS 2021. *PostGIS Manual* [online]. Ver. 3.1.2dev. Dostupné z: <https://postgis.net/docs/manual-3.1/> [cit. 27. 11. 2021].

POSTGRESQL 2021. *About PostgreSQL* [online]. Dostupné z: <https://www.postgresql.org/about/> [cit. 27. 11. 2021].

STROBL, Ch. 2008. PostGIS. In: SHEKHAR, S., XIONG, H. (eds.): *Encyclopedia of GIS*. Boston (MA) : Springer. s. 891-898. ISBN 978-0-387-35973-1.