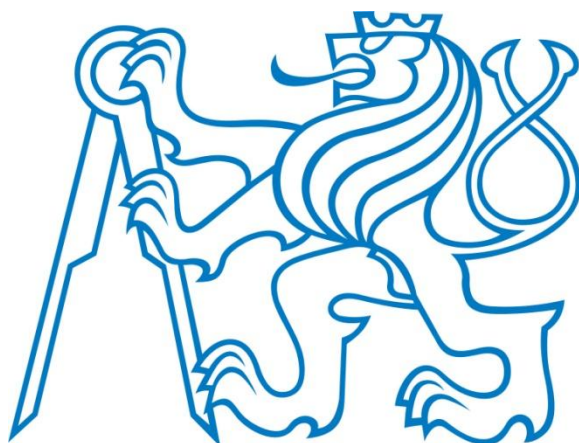


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Katedra geomatiky



Úvod do zpracování prostorových dat

Semestrální projekt

Obsah:

- Zadání
- 1. Úvod
- 2. Tematické vrstvy
 - 2.1. Funkce PostGIS
 - 2.2. Databázová schémata
 - 2.3. Validace dat
 - 2.4. Souřadnicový systém
 - 2.5. Transformace dat
 - 2.6. Železnice
 - 2.7. Opuštěné železnice
 - 2.8. Železniční stanice
 - 2.9. Tramvajové koleje
 - 2.10. Tramvajové stanice
 - 2.11. Koleje metra
 - 2.12. Stanice metra
 - 2.13. Města
 - 2.14. Parkovací plochy
 - 2.15. Jednotky
- 3. QGIS/PostGIS Tutoriál
- 4. Závěr
- Příloha A - Použité funkce PostGIS

Další přílohy:

- SQL dávka pro tvorbu vrstev. (Použité kódování – UTF-8 dle zadání) – [davka.sql](#)
- Tutoriál v podobě prezentace ve formátu pdf – [tutorial.pdf](#)

Zadání

- Navrhněte a vytvořte tématické vrstvy (např. vodní toky, vodní plochy, lesy, silnice, železnice apod.) na základě dat OpenStreetMap (viz schéma osm) a další otevřených zdrojů.
- Aplikujte testy datové integrity a odstraňte případné nekonzistence v datech
- Vytvořte tutoriál - tj. sadu atributových a prostorových dotazů nad databází pgis_uzpd.

1. Úvod

Cílem projektu je vytvoření tematických vrstev, odstranění topologických, logických a jiných chyb v těchto vrstvách a sestavení sady příkladů nad těmito vrstvami, které by mohly sloužit jako výukový materiál.

Jako zdroj pro tvorbu tematických vrstev posloužila data z komunitního projektu OpenStreetMap dále jen OSM a také data z registru RÚIAN (Registr územní identifikace adres a nemovitostí).

Zmíněná data byla pro účely projektu předem připravena v databázi s názvem pgis_uzpd, která je přístupná z fakultního serveru geo102.fsv.cvut.cz. Jedná se o databázi PostgreSQL s rozšířením PostGIS, které doplňuje databázový systém PostgreSQL o prostorové objekty a umožňuje tak systém využívat jako prostorovou databázi. Kromě toho PostGIS obsahuje celou řadu funkcí pro prostorové analýzy a manipulaci s prostorovými daty.

Projekt byl zpracován v prostředí programu QGIS verze 2.10.1 Pisa. QGIS umožňuje připojení prostorové databáze a práci s ní. Je vybaven nástrojem s názvem DB Manager, pomocí kterého je možné s databází pracovat skrze SQL příkazy a prohlížet schéma databáze. Data z databáze lze rovnou načítat jako zobrazované vrstvy a kromě toho je možné jako vrstvy zobrazit i výsledky konkrétních SQL dotazů.

Pro účely projektu bylo v databázi pgis_uzpd vytvořeno schéma s názvem b15, do kterého byly uloženy výsledné tematické vrstvy.

2. Tematické vrstvy

Vytvořené tematické vrstvy se zaměřují především na železniční (kolejovou) dopravu. Byly vytvořeny vrstvy tramvajových kolejí a tramvajových stanic, železničních kolejí a železničních stanic, kolejí metra a stanic metra a také vrstva obsahující města a vrstva obsahující parkovací plochy.

Obsah všech vrstev je omezen na území České republiky a veškerý obsah je z hlediska geometrie validní.

V této kapitole jsou popsány jednotlivé vrstvy, způsob jejich tvorby a řešení problémů při jejich tvorbě.

SQL Příkazy pro tvorbu všech tematických vrstev (tabulek) jsou součástí příloh a jsou k nalezení v souboru `avka.sql`.

2.1 Funkce PostGIS

V této kapitole bude zmíněna celá řada funkcí PostGIS. Všechny funkce PostGIS začínají prefixem `ST_` podle čehož je možné je odlišit od běžných SQL příkazů a funkcí PostgreSQL. V rámci tohoto dokumentu budou názvy funkcí PostGIS označeny **takto**. V příloze A je uveden seznam odkazů na dokumentaci PostGIS, ve kterém naleznete veškeré funkce použité v rámci tohoto projektu.

2.2 Databázová schémata

V rámci SQL příkazů, které budou uvedeny v této práci, jsou použity názvy příslušných databázových schémat. Proto jsou zde pro přehlednost uvedena.

Data z OSM se v databázi `pgis_uzpd` nachází ve schématu `osm`. Data z registru RÚIAN se nachází ve schématu `ruian` a veškeré vytvořené tematické vrstvy se nachází ve schématu `b15`. Každé tematické vrstvě odpovídá jedna tabulka ve schématu `b15`.

2.3 Validace dat

Jedním z úkolů je zkontrolovat integritu dat a odstranit případné nekonzistence. Pokud mají data špatnou integritu, znamená to, že jejich geometrie není v pořádku. PostGIS obsahuje funkci `ST_IsValid`, která slouží pro kontrolu geometrie. Pokud je geometrie v pořádku vrátí logickou hodnotu `true` v opačném případě vrátí hodnotu `false`. Vstupním parametrem funkce je sloupec obsahující geometrii, která má být zkontrolována.

Př. 1 kontrola validity dat

```
SELECT * FROM b15.metro WHERE ST_IsValid(geom) = false;
```

Dotaz v příkladu 1 vrací všechny řádky z tabulky metro, u kterých není geometrie v pořádku.

Pro účely kontroly integrity dat (geometrie) byla použita právě tato funkce. Pro odstranění logických chyb, jako např. duplicit, apod. byl vždy zvolen individuální přístup.

2.4 Souřadnicový systém

Jako výchozí souřadnicový systém byl použit S-JTSK ve variantě East-North, kterému odpovídá EPSG kód 5514. Veškerá geometrie ve schématu b15 je právě v tomto souřadném systému.

2.5 Transformace dat

Data OSM jsou v databázi pgis_uzpd uložena v souřadnicovém systému s EPSG kódem 3857 (WGS84/Pseudo-Mercator). Vzhledem k tomu, že jako výchozí systém pro schéma b15 byl zvolen S-JTSK s EPSG kódem 5514, muselo dojít k transformaci geometrie. Ta byla prováděna pomocí funkce **ST_Transform**. Ta jako první parametr přijímá geometrii, která má být transformována, jako druhý parametr přijímá EPSG kód a vrací transformovanou geometrii.

Př. 2 Transformace dat

```
SELECT st_transform(geom, 5514)
FROM osm.czech_line
WHERE railway = 'abandoned';
```

Dotaz v příkladu 2 transformuje geometrii všech opuštěných železnic ze stávajícího systému do systému s EPSG kódem 5514.

Data ze schématu ruian jsou v databázi pgis_uzpd uložena systému s EPSG kódem 5514 a tudíž nebylo nutné provádět u nich transformaci. Je ovšem nutné dbát na to, aby nedošlo k míchání různých systémů. Pokud by například v rámci jednoho dotazu byla použita geometrie ze schématu osm, a zároveň geometrie ze schématu ruian je nutné, aby obě geometrie byly ve stejném souřadnicovém systému.

Př. 3 Míchání souřadnicových systémů

```
CREATE TABLE b15.parkoviste AS
SELECT osm.osm_id, osm.name, st_transform(osm.geom, 5514) as geom
FROM osm.czech_polygon AS osm
JOIN ruian.staty AS s ON st_within(st_transform(osm.geom, 5514), s.geom)
WHERE amenity = 'parking';
```

*V příkladu 3 je vytvářena tabulka obsahující parkovací plochy převzaté ze schématu osm (epsg: 3857). Výběr je omezen na území ČR, k čemuž je použit polygon státu ze schématu ruian (epsg: 5514). Uvnitř funkce **ST_Within** je geometrie z osm transformována, aby mohla být správně porovnána s geometrií státu ze schématu ruian.*

2.6 Železnice

Vrstva železnice obsahuje linie reprezentující železnice na území ČR. Sloupec *length* obsahuje délku segmentu linie vypočtenou pomocí funkce **ST_Length**. Sloupce *tunnel* a *bridge* obsahují hodnotu 'yes' pokud je segment linie tunelem nebo mostem. V opačném případě obsahují hodnotu 'no'.

Pro tvorbu této vrstvy byly použity data z OSM. Že jde o železnici, se pozná dle atributu *railway*, jehož hodnota musí být *rail*. Omezení na území ČR bylo provedeno pomocí polygonu státu ze schématu ruian.

Př. 4 Výběr železnic z tabulky *osm.czech_line*

```
SELECT osm_id, bridge, tunnel, st_transform(geom, 5514) FROM osm.czech_line  
WHERE railway = 'rail';
```

b15. zeleznice	
osm_id	Int8
gid PK	Int4
bridge	varchar
tunnel	varchar
length	numeric
geom	geometry(LineString, 5514)

2.7 Opuštěné železnice

Vrstva opuštěné železnice má strukturu totožnou s vrstvou železnice a obsahuje všechny opuštěné železnice na území ČR. Opuštěná železnice se pozná tak, že atribut *railway* má hodnotu *abandoned*.

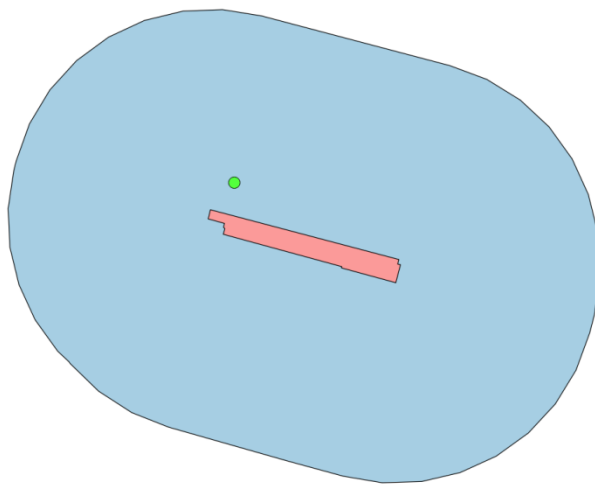
2.8 Železniční stanice

Vrstva železniční stanice je bodovou vrstvou obsahující železniční stanice. Železniční stanice se pozná opět podle atributu *railway*, který má v tomto případě hodnotu *station*. Kromě toho je možné použít také atribut *public_transport* a jeho hodnoty *stop_position*, nebo *platform*, který je společný pro veškerou veřejnou dopravu. Ovšem vzhledem k tomu že je tento atribut obecný, není možné poznat, zda se jedná o železniční, autobusovou, nebo jinou zastávku a proto je byl použit atribut *railway*.

Data byla převzata především z tabulky *osm.czech_point*. Ovšem železniční stanice byly nalezeny i v tabulce *osm.czech_polygon* a dokonce i v tabulce *osm.czech_line*. V případě linií se jednalo o zjevné chyby, kdy se někdo snažil nakreslit polygon a místo toho nakreslil jen jeho obrys. Takových případů bylo celkem 5.

Mezi polygony byly i stanice, které se nevyskytovali mezi body. Pokud by byla jednoduše převzata pouze bodová data, tyto stanice by chyběly. Proto byly identifikovány polygony těchto stanic a pomocí funkce *ST_Centroid* z nich byly vytvořeny body, které byly přidány do výsledné vrstvy železničních stanic. Ještě před tím, bylo všech pět zvláštních případů v liniových datech převedeno na polygony pomocí funkce *ST_MakePolygon*, která přijímá pouze uzavřené linie, což bylo ověřeno pomocí funkce *ST_IsClosed*.

K identifikaci polygonů stanic, které se nevyskytují mezi bodovými daty, musel být použit buffer kolem polygonu stanice. Důvod skvěle demonstruje obrázek 1.



Obr. 1 Železniční stanice vyjádřená bodem leží mimo svůj polygon. Buffer 100 m

Z obrázku 1 je patrné, že pouhé nalezení polygonů stanic, uvnitř kterých neleží žádný bod, by nestačilo. Proto byl kolem polygonů vytvořen buffer pomocí funkce *ST_Buffer* o velikosti 100 m a byly hledány takové polygony, v jejichž bufferu neleží žádný bod znázorňující stanici. Takto bylo získáno dalších zhruba 300 železničních stanic.

Sloupec *name* obsahuje název stanice. Jako vedlejší produkt vznikla také tabulka obsahující polygonové reprezentace stanic. Tato tabulka je ovšem výrazně menší. Obsahuje pouze celkem 364 stanic, zatímco tabulka *zeleznice_stanice_body* obsahuje 1320 stanic. A samozřejmě platí, že tabulka *zeleznice_stanice_body* obsahuje všechny stanice z tabulky *zeleznice_stanice_polygony*.

b15. zeleznice_stanice_body	
osm_id PK	Int8
name	varchar
geom	geometry(Point, 5514)

b15. zeleznice_stanice_polygony	
osm_id PK	Int8
name	varchar
geom	geometry(MultiPolygon, 5514)

2.9 Tramvajové koleje

Vrstva tramvaje obsahuje linie reprezentující tramvajové koleje v ČR. Podobně jako železnice tramvajové koleje se v datech OSM hledají podle atributu *railway*, který musí mít hodnotu *tram*.

Sloupec *length* obsahuje délku segmentu linie vypočtenou pomocí funkce *ST_Length*.

b15. tramvaje	
osm_id	Int8
gid PK	Int4
length	numeric
geom	geometry(LineString, 5514)

2.10 Tramvajové stanice

Vrstva tramvajových stanic je bodová reprezentace zástavek tramvají v ČR. Data byla převzata z OSM. V případě tramvajových stanic má atribut *railway* hodnotu *tram_stop*.

b15. tramvaje_stanice	
osm_id PK	Int8
name	varchar
geom	geometry(Point, 5514)

2.11 Koleje metra

Vrstva metro je liniovou vrstvou vytvořenou z dat OSM, která obsahuje koleje (trasy) metra. Koleje metra lze v datech OSM najít pomocí atributů *railway*, jehož hodnota je *subway*.

b15. metro	
osm_id	Int8
gid PK	Int4
length	numeric
geom	geometry(Point, 5514)

2.12 Stanice metra

Vrstva stanice metra obsahuje stanice metra. Data jsou převzata z dat OSM, ale byla získána z vrstvy železničních stanic. Stanice metra jsou totiž v datech OSM označena jako železniční stanice. Naštěstí jsou ale kresleny přímo na linii kolejí metra a lze je tak identifikovat pomocí funkce *ST_Intersects*.

Př. 5 výběr stanic metra z vrstvy železničních stanic

```
SELECT DISTINCT b.osm_id, b.name, st_transform(b.geom, 5514) as geom
FROM b15.zeleznice_stanice_body as b
JOIN b15.metro as m ON st_intersects(b.geom, m.geom) ORDER BY b.osm_id;
```

V příkazu je klíčové slovo DISTINCT, aby některé stanice nebyly vyhodnoceny dvakrát.

b15. metro_stanice	
osm_id PK	Int8
name	varchar
geom	geometry(Point, 5514)

V tomto případě je nutné dát si pozor na použití SQL dávky. Tabulka `zeleznice_stanice_body` musí být vytvořena před tabulkou `metro_stanice`, která zase musí být vytvořena až po tabulce `metro`. Poté co je vytvořena tabulka `metro_stanice` je vhodné zavolat SQL příkaz, který odstraní stanice metra z tabulky železničních stanic. SQL dávka přiložená k této práci je seřazena tak, aby vše proběhlo správně a stanice metra byly nakonec z tabulky železničních stanic odstraněny.

2.13 Města

Polygonová vrstva města byla vytvořena z dat RÚIAN a obsahuje všechny obce, které mají status město, statutární město, nebo hlavní město. Tato vrstva byla vytvořena především pro účely výukových příkladů.

b15. mesta	
gid PK	Int4
area	numeric
population	Int8
name	varchar
status	Int4
geom	geometry(MultiPolygon, 5514)

Sloupec `area` obsahuje plochu města vypočtenou pomocí funkce `ST_Area`. Sloupec `name` obsahuje název města. Sloupec `population` obsahuje počet obyvatel. Tento údaj byl převzat z dat OSM a bohužel je vyplněn jen u několika větších měst. Sloupec `status` obsahuje číselné vyjádření statutu města. Tento údaj je převzat z RÚIAN.

Sloupec `status` může obsahovat tři hodnoty:

- 3 – Město
- 4 – Statutární město
- 5 – Hlavní město

2.14 Parkovací plochy

Polygonová vrstva parkoviště obsahuje parkovací plochy z celého území ČR. Data jsou převzata z OSM. Parkoviště lze nalézt pomocí atributu *amenity*, který má hodnotu *parking*.

b15. parkoviste	
gid PK	Int4
osm_id	Int8
area	numeric
name	varchar
geom	geometry(MultiPolygon, 5514)

Sloupec area obsahuje plochu parkoviště vypočtenou pomocí funkce **ST_Area**. Sloupec name obsahuje název, pokud je vyplněn.

2.15 Jednotky

Jednotky v jakých jsou uloženy hodnoty sloupců, jako jsou length, nebo area odpovídají vždy jednotce použité v rámci souřadnicového systému, ve kterém je uložena geometrie.

V případě S-JTSK (EPSG 5514) je jednotkou metr. Hodnoty sloupců obsahujících délku jsou tedy metry a jednotky sloupců obsahujících plochu jsou metry čtvereční.

3. QGIS/PostGIS Tutoriál

Tutoriál jsem se rozhodl vytvořit jako samostatný pdf dokument v podobě prezentace. Obsahuje sadu příkladů v rozmezí od těch nejjednodušších až po složitější. Příklady jsou zaměřeny jak na atributové, tak na prostorové dotazy, ale také na funkce PostGIS a logiku jejich použití. Každý příklad obsahuje kromě zadání také správný výsledek, SQL příkaz, který vede ke správnému výsledku a vysvětlení navrženého postupu spolu s vysvětlením použitých funkcí PostGIS.

4. Závěr

Bylo vytvořeno celkem devět tematických vrstev. Tramvajové koleje a tramvajové stanice, stanice metra a koleje metra, železnice, opuštěné železnice a železniční stanice, města a parkovací plochy. Data byla převzata převážně z OpenStreetMap, ale v některých případech byla použita i data RÚIAN a to především pro definici hranic měst a státu ČR. Geometrie převzatá z OSM byla transformována do S-JTSK přesněji do varianty s EPSG kódem 5514. Liniová data byla doplněna o atribut délky a plošná data o atribut plochy. Při přebírání dat z OSM bylo snahou převzít co nejvíce atributů, bohužel většina atributových dat nebyla v OSM často k dispozici, jako například počet parkovacích míst na parkovišti apod. K dispozici byla většinou jen geometrie, jedinečný identifikátor a název pokud se jednalo o prvek, který z logiky věci název má.

Nad vytvořenými vrstvami byl vytvořen tutoriál v podobě prezentace, který obsahuje sérii příkladu s vysvětlením a ukázkou jejich řešení.

S prostorovou databází PostgreSQL/PostGIS se pracuje velmi dobře. Ve srovnání například s databází Oracle je práce s prostorovými dotazy velmi snadná a jasná. Použití DB Manageru a prostorové databáze v prostředí QGIS považuji za velmi efektivní.

Příloha A – Použité funkce PostGIS

ST_IsValid	http://postgis.net/docs/ST_IsValid.html
ST_Transform	http://postgis.org/docs/ST_Transform.html
ST_Within	http://postgis.net/docs/manual-1.4/ST_Within.html
ST_Length	http://postgis.org/docs/ST_Length.html
ST_Area	http://postgis.org/docs/ST_Area.html
ST_MakePolygon	http://postgis.net/docs/ST_MakePolygon.html
ST_IsClosed	http://postgis.net/docs/ST_IsClosed.html
ST_Centroid	http://postgis.org/docs/ST_Centroid.html
ST_Buffer	http://www.postgis.org/docs/ST_Buffer.html
ST_DWithin	http://postgis.net/docs/ST_DWithin.html
ST_Intersects	http://postgis.org/docs/ST_Intersects.html