

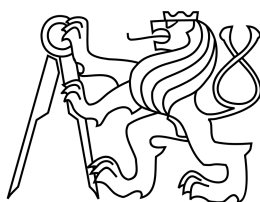
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

KATEDRA MAPOVÁNÍ A KARTOGRAFIE

## DOKUMENTACE

ÚVOD DO ZPRACOVÁNÍ PROSTOROVÝCH DAT



SKUPINA C:

**Eliška Kyzlíková**  
**Štěpán Turek**  
**Jan Zajíček**

2011

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	Zadání projektu . . . . .	2
<b>2</b>	<b>Validace vrstev</b>	<b>3</b>
<b>3</b>	<b>Tvorba tematických vrstev</b>	<b>8</b>
3.1	Zahrady . . . . .	8
3.2	Obce . . . . .	8
3.2.1	Vznik vrstvy . . . . .	8
3.2.2	Úprava vrstvy . . . . .	9
3.3	Integrovaný záchranný systém . . . . .	9
3.3.1	Vznik bodové a polygonové vrstvy . . . . .	9
3.3.2	Úprava vrstvy . . . . .	10
<b>4</b>	<b>Dotazy k tutoriálu pro výuku PostGIS</b>	<b>12</b>
4.1	Atributové dotazy a prostorové dotazy . . . . .	12
<b>5</b>	<b>Závěr</b>	<b>16</b>
5.1	Použité programy . . . . .	17

# 1 Úvod

Tato dokumentace slouží k seznámení se semestrálním projektem, který vznikl v rámci předmětu Úvod do zpracování prostorových dat. Hlavní náplní projektu je práce s prostorovými daty prostřednictvím databáze PostGIS. Naším cílem bylo vytvořit validní polygonovou vrstvu, která byla posléze, spolu s dalšími tématickými vrstvami, využita při tvorbě atributových a prostorových dotazů. Při tvorbě projektu bylo použito znalostí z předmětů Databázové systémy a Geografické informační systémy 1 a 2.

## 1.1 Zadání projektu

1. Návrh a tvorba tématických vrstev na základě dat OpenStreetMap ČR ze dne 9.3.2011.
2. Aplikace testů datové integrity a odstranění případné nekonzistence v datech.
3. Vytvoření tutoriálu pro výuku PostGIS (sada atributových a prostorových dotazů).

## 2 Validace vrstev

Pro validaci polygonu jsme zvolili vrstvu zahrad. Funkcí `ST_IsValid` bylo zjištěno, že vrstva obsahuje 181 nevalidních polygonů. Nejprve je z tabulky `polygonů czech_polygon` obvyklým způsobem vytvořena tabulka `zahrady`.

```
CREATE TABLE zahrady AS
SELECT osm_id, ST_Multi(way) AS geom
FROM czech_polygon WHERE leisure='garden';
SELECT populate_geometry_columns('c11.zahrady'::regclass);

ALTER TABLE zahrady ADD COLUMN gid serial;
ALTER TABLE zahrady ADD PRIMARY KEY(gid);
```

Jediná zvláštnost je převod geometrií typu `polygon` na `multipolygon` funkcí `ST_Multi`. Důvod pro tento převod je, že zvalidované polygony jsou sdruženy do `multipolygonů` (viz. níže) a při aktualizaci tabulky `zahrady` by v této tabulce došlo ke kombinaci dvou typů geometrií, k čemuž nemá docházet.

Poté se vytvoří tabulka `validni_polygony`, která obsahuje polygonizované `linestringy`, které tvoří hranici `polygonu`:

```
DROP TABLE IF EXISTS validni_polygony;
CREATE TABLE validni_polygony AS
SELECT (ST_Dump(ST_Polygonize(ST_Union(geom, ST_Startpoint(geom))))).geom
AS geom, gid, path[1] AS path
FROM
(SELECT (ST_ExteriorRing((ST_DumpRings((ST_Dump(geom)).geom)).geom))
AS geom, (ST_DumpRings((ST_Dump(geom)).geom)).path AS path , gid AS
gid
FROM zahrady
WHERE ST_IsValid(geom)= FALSE
) as ext_linestring
GROUP BY geom, gid, path;
ALTER TABLE validni_polygony ADD COLUMN gid_unique serial;
```

Funkce `ST_DumpRings` vrací dvě hodnoty. S příponou `.geom` vrátí polygony, jež byly vytvořeny z hraničních linestringů vstupního polygonu. S příponou `.path` vrátí 0, pokud daný polygon (dále vnější polygon) je vytvořen z obalového linestringu nebo vrátí hodnotu větší než 0, což se děje u polygonu (dále vnitřní polygon) vytvořených z děr. Jako argument této funkce musí být geometrie typu polygon. Jelikož vrstva zahrady obsahuje multipolygony, musely být převedeny na polygony funkcí `ST_Dump`.

Funkce `ST_ExteriorRing` vrací obálkový linestring polygonu. Díky této funkci získáme linestringy, které kopírují hranice původního polygonu.

Pomocí `ST_Union` přidáme do linestringu jeho počáteční bod (`ST_Startpoint`). Touto operací dosáhneme přidání neznámých průsečíků do linestringu. Tyto neznáme průsečíky se objevují u self-intersection polygonů, kdy se křížují hrany jednoho linestringu. Po přidání těchto průsečíků je možno z tohoto nevalidního polygonu vytvořit validní polygony. Toto provádí pomocí funkce `ST_Polygonize`, která vytvoří ze všech linestringů polygony popř. multipolygony.

Pomocí funkce `ST_Dump` se multipolygony rozbijí na jednotlivé polygony. Současně díky atributu `path` si každý polygon nese informaci o typu hranice linestringu ze kterého byl vytvořen (viz. výše).

Všem polygonům v tabulce `validni_polygony` je přiřazeno unikátní číslo:

```
ALTER TABLE validni_polygony
ADD COLUMN gid_unique serial;
```

Tímto dotazem se odečtou polygony vnitřní hranice od polygonu vnější hranice:

```
UPDATE validni_polygony
SET geom = with_holes.geom
FROM
(SELECT ST_Difference(shells.geom,holes.geom)as geom, shells.gid_unique
FROM validni_polygony as shells
JOIN
(SELECT ST_Union(geom) as geom, gid
FROM validni_polygony
WHERE path!=0
GROUP BY gid) as holes
ON holes.gid=shells.gid
AND shells.path=0);
```

Všechny vnitřní polygony se nejprve sloučí do jedné geometrie funkcí `ST_Union` a poté se odečtou od vnějších polygonů funkcí `ST_Difference`. Následně je geometrie vnějších polygonů aktualizována v tabulce `validni_polygony` příkazem `UPDATE`.

Vnitřní polygony již nebudou potřeba a tak jsou smazány:

```
DELETE FROM validni_polygony
WHERE path!=0;

DELETE FROM validni_polygony
WHERE gid_unique IN
(SELECT holes.gid_unique
FROM validni_polygony AS shells
JOIN validni_polygony AS holes
ON shells.gid = holes.gid
AND shells.gid_unique != holes.gid_unique
AND ST_Relate(holes.geom,shells.geom,'FF2F1F212'))
```

Pomocí tohoto příkazu se smažou polygony, které jsou celé obklopeny jiným polygonem. Tato anomálie vzniká, pokud se hraniční linestring sebe dotýká. Můžeme si to představit jako dva uzavřené linestringy (ve skutečnosti je

to jeden linestring), kdy jeden je uvnitř druhého, a oba se v jednom bodě dotýkají. V tomto případě funce `ST_Polygonize` vytvoří z tohoto linestringu dva polygony, z nichž jeden je obklopen druhým. Obklopený polygon se vyhledá funkcí `ST_Relate` s takto definovanou tabulkou průniku `'FF2F1F212'`. Tento polygon je třeba smazat, aby tvar odpovídal původnímu polygonu.

Poté se polygony sdruží do multipolygonů podle původního gidu a vytvoří se tabulka `validni_polygony_collect`:

```
DROP TABLE IF EXISTS validni_polygony_collect;
CREATE TABLE validni_polygony_collect AS
SELECT ST_Collect(geom) AS geom, gid
FROM
(SELECT (ST_Dump(geom)).geom as geom, gid
FROM validni_polygony
)as val_p
GROUP BY gid ;
```

Funkce `ST_Collect` někdy vrací typ `geometrycollection`. Aby vracela vždy typ `multipolygon` byl použit trik, který je uveden v Postgis referenci s funkcí `ST_Dump`. Může se stát, že výsledné multipolygony jsou nevalidní (např. dotyk slučovaných polygonů má dimenzi 1), a proto je třeba je spojit do jednoho polygonu a vytvořit tak validní polygon. Tento problém řeší funce `ST_BuildArea`:

```
UPDATE validni_polygony_collect
SET geom = with_holes.geom
FROM
(SELECT ST_BuildArea(geom) as geom, gid
FROM validni_polygony_collect
WHERE st_isvalid(geom)=False
) as with_holes
WHERE validni_polygony_collect.gid= with_holes.gid;
```

V posledním kroku jsou aktualizovány nevalidní polygony ve zdrojové tabulce zahrady polygony validními:

```
UPDATE zahrady AS updated_table
SET geom = ST_Multi(validni_polygony_collect.geom)
FROM validni_polygony_collect
WHERE updated_table.gid= validni_polygony_collect.gid;
```

Tabulka `validni_polygony_collect` obsahuje geometrie typu `polygon` i `multipolygon`, protože funkce `ST_BuildArea` v předchozím kroku vrací typ `polygon`. Aby tabulka zahrady obsahovala pouze typ `multipolygon`, jsou zbylé typy `polygon` převedeny na `multipolygon` funkcí `ST_Multi`.

Nyní vrstva zahrady obsahuje pouze validní polygony.

Na závěr porovnání vybraného polygonu před a po validaci:



polygon před validací    polygon po validaci



## 3 Tvorba tematických vrstev

Tematické vrstvy byly vytvořeny na základě dat ze schématu *osm* a *gis1*. Pro potřeby validace byla použita vrstva zahrady. Výběr této vrstvy byl opodstatněn tím, že obsahuje 181 nevalidních polygonů. Za účelem rozšíření tvorby atributových a prostorových dotazů byly vytvořeny vrstvy územních celků a Integrovaného záchranného systému.

### 3.1 Zahrady

Polygonová vrstva určená pro validaci dat. Vrstva zahrady byla použita ze schématu *osm*, vrstva *czech\_polygon*. Polygonová vrstva zahrady byla vytvořena následovně:

```
CREATE TABLE IF EXISTS zahrady AS SELECT osm_id, way AS geom FROM
czech_polygon WHERE leisure='garden';
ALTER TABLE zahrady ADD COLUMN gid serial;
ALTER TABLE zahrady ADD PRIMARY KEY(gid);
```

Takto vzniklá vrstva byla posléze zvalidována (viz. Validace vrstev).

### 3.2 Obce

Vrstva obce byla použita ze schématu *gis1*. Pro kombinaci s vrstvami ze schématu *osm* bylo nutno vzniklou vrstvu transformovat do souřadnicového systému dat *osm*. Z důvodu častého použití v atributových a prostorových dotazech byla z dat tabulky obce vytvořena vrstva kraje.

#### 3.2.1 Vznik vrstvy

```
CREATE TABLE obce AS SELECT * from gis1.obce;
UPDATE obce SET geom=ST_TRANSFORM(geom,900913);
SELECT Populate_geometry_column(pgclass::'obce');
```

### 3.2.2 Úprava vrstvy

1. Redukce sloupců:

```
ALTER TABLE obce DROP COLUMN nazev_eng;
```

2. Přejmenování sloupců:

```
ALTER TABLE obce RENAME COLUMN area to vymera;
```

3. Přepsání kodů krajů a okresů na názvy:

```
UPDATE obce SET kraj='Pardubický' WHERE kraj='CZ053';
```

## 3.3 Integrovaný záchranný systém

Vrstva obsahuje záznamy nemocnic, požárních stanic a policejních stanic. Požadovaná data byla ve formě bodů a polygonů. Vrstvy IZS\_bod a IZS\_poly byly vytvořeny na základě dat ze schématu *osm*.

### 3.3.1 Vznik bodové a polygonové vrstvy

```
CREATE TABLE IZS_bod
AS SELECT osm_id, way AS geom, amenity AS druh ,czech_point.name
AS popis
FROM czech_point
WHERE amenity = 'fire_station' OR amenity = 'police' OR amenity =
'hospital';
```

```
CREATE TABLE IZS_poly
AS SELECT osm_id, way AS geom, amenity AS druh ,czech_polygon.name
AS popis
FROM czech_polygon
WHERE amenity = 'fire_station' OR amenity = 'police' OR amenity =
'hospital';
```

### 3.3.2 Úprava vrstvy

1. Odstranění duplicit z tabulky bodu:

```
DELETE FROM IZS_bod
WHERE IZS_bod.geom IN
(SELECT IZS_bod.geom
FROM IZS_bod
JOIN IZS_poly
ON ST_WITHIN(IZS_bod.geom, IZS_poly.geom));
```

2. Odstranění zahraničních IZS:

```
DELETE FROM IZS_bod
WHERE IZS_bod.geom NOT IN
( SELECT izs_bod.geom
FROM IZS_bod
JOIN kraje
ON ST_Within(IZS_bod.geom, kraje.geom));
```

```
DELETE FROM IZS_poly
WHERE IZS_poly.geom NOT IN
( SELECT izs_poly.geom
FROM IZS_poly
JOIN kraje
ON ST_Within(IZS_poly.geom, kraje.geom));
```

3. Spojení do jedné společné bodové vrstvy:

```
DROP TABLE if exists centr;
CREATE TABLE centr
AS SELECT osm_id,ST_Centroid(izs_poly.geom) as geom,druh,popis,gid
FROM izs_poly;
INSERT INTO izs_bod (osm_id,geom,druh,popis)
SELECT osm_id,geom,druh,popis
FROM centr;
```

4. Přeložení názvů druhů IZS do češtiny:

```
UPDATE IZS_bod SET druh = 'policie' WHERE druh='police';
UPDATE IZS_bod SET druh = 'nemocnice' WHERE druh='hospital';
UPDATE IZS_bod SET druh = 'požární stanice' WHERE druh='fire_station';
```

5. Definice primárního klíče nad sloupcem gid:

```
ALTER TABLE IZS_bod ADD COLUMN gid serial;
ALTER TABLE IZS_bod ADD PRIMARY KEY(gid);
```

6. Vytvoření prostorového indexu:

```
CREATE INDEX IZS_bod_geom_gist
ON IZS_bod
USING GIST(geom);
```

## 4 Dotazy k tutoriálu pro výuku PostGIS

### 4.1 Atributové dotazy a prostorové dotazy

Pracující s atributy geografických dat, jejich geometrií a topologií.

Příklad několika našich dotazů:

1. Kolik zahrad je do vzdálenosti 1 km od obce Jihlava?

```
CREATE TABLE pokusna_zajcev AS SELECT ST_Buffer(geom, 1e3) AS
geom
FROM obce
WHERE obec = 'Jihlava';
SELECT COUNT(*)
FROM zahrady
JOIN pokusna_zajcev
ON ST_Intersects(pokusna_zajcev.geom, zahrady.geom);
```

4476
------

2. Jaký je nejmenší obvod obce?

```
SELECT obec,obvod
FROM obce
ORDER BY obvod
LIMIT 1;
```

Závist	3084 m
--------	--------

3. V kterém kraji a obci ordinuje Mudr. Konrádová?

```
SELECT kraj,obec
FROM obce
JOIN izs_bod
ON ST_Within(izs_bod.geom, obce.geom)
WHERE izs_bod.popis = 'MuDr. Konrádová';
```

Olomoucký	Horka nad Moravou
-----------	-------------------

4. Kolik městských policejních stanic ('M(m)ěstská policie%') je ve vzdálenosti do 100 km od Prahy?

```
CREATE TABLE praha100 AS
SELECT ST_Buffer(geom, 1e5) AS geom
FROM obce
WHERE obec = 'Praha';
SELECT count(*)
FROM iza_bod
JOIN praha100
ON ST_Intersects(praha100.geom, iza_bod.geom)
WHERE popis LIKE '%ěstská policie%';
```

23

5. Jakou výměru má největší zahrada Zlínského kraje? Plochu uveďte v hektarech s přesností na 3 desetinná místa.

```
SELECT ROUND(MAX(ST_AREA(zahrady.geom)/1e4) :: NUMERIC ,3)
FROM zahrady
JOIN kraje
ON ST_Within(zahrady.geom,kraje.geom)
WHERE kraj = 'Zlínský';
```

21.969 ha

6. kolik procent bývalých okresních měst je dnes ORP?

```
SELECT ROUND((stejne.cislo/(vse.cislo/100)))
FROM
(SELECT COUNT(*) ::NUMERIC AS cislo
FROM obce
WHERE okres=orp AND orp=obec) AS stejne
CROSS JOIN
(SELECT COUNT(distinct okres) ::NUMERIC AS cislo
FROM obce) AS vse ;
```

87%

7. Kolik nemocnic, hasičských stanic a policejních stanic se nachází na území Prahy?

```
SELECT COUNT(*) AS počet,druh
FROM izs_bod
JOIN obce
ON ST_Within(izs_bod.geom,obce.geom)
WHERE obec='Praha'
GROUP BY druh;
```

nemocnice	40
hasičská stanice	9
policejní stanice	50

8. Kde se nachází Hamzova léčebna? Uveďte obec, okres a kraj.

```
SELECT obec,okres,kraj,popis
FROM izs_bod
JOIN obce
ON ST_Within(izs_bod.geom,obce.geom)
WHERE popis LIKE 'Hamzova léčebna';
```

Luže	Chrudim	Pardubický
------	---------	------------

9. V jaké vzdálenosti se nachází 5 nejbližších požárních stanic od obce Chropyně? Vzdálenost uveďte v kilometrech.

```
SELECT popis,ROUND(ST_Distance(obce.geom,izs_bod.geom)/1e3)
AS vzdalenost
FROM obce,izs_bod
WHERE obec ='Chropyně' AND druh='požární stanice'
ORDER BY vzdalenost
LIMIT 5;
```

Hasičská zbrojnice	4 km
Kajetín	4 km
	5 km
Hasičská zbrojnice	14 km
	22 km

10. Z jaké nemocnice je nejkratší přímá časová vzdálenost (na minuty) do obce Dlouhopolsko při průměrné rychlosti sanitky 60 km/h ?

```
SELECT popis,ROUND(((ST_Distance(obce.geom,izs_bod.geom))/(60/3.6))/60)
AS čas
FROM obce,izs_bod
WHERE obec ='Dlouhopolsko' AND druh='nemocnice'
ORDER BY čas
LIMIT 1;
```

Městská nemocnice Městec Králové	4 min.
----------------------------------	--------



## 5 Závěr

Tato dokumentace mapuje vznik a shrnuje výsledky našeho semestrálního projektu. Práce na projektu nám přinesla zkušenost se zpracováním dat prostřednictvím prostorové databáze PostGIS. Vytvořili jsme atributové a prostorové dotazy nad vrstvami IZS\_bod, obce, kraje a provedli jsme validaci polygonové vrstvy zahrady. Použité vrstvy jsou uloženy ve schématu c11, v databázi *pgis\_uzpd* na serveru *geo102.fsv.cvut.cz*. Postup řešení dokumentuje SQL dávkový soubor.

### Autoři projektu:

Eliška Kyzlíková

Štěpán Turek

Jan Zajíček

## 5.1 Použité programy

1. pgAdminIII 1.10.2
2. Quantum GIS 1.6.0
3. Kile 2.0.86
4. MikTeX 2.8

## Reference

- [1] *<http://geo101.fsv.cvut.cz/gwiki/153UZPD>*
- [2] *<http://postgis.refrations.net/documentation/manual-1.3/ch06.html>*
- [3] *<http://openstreetmap.org>*
- [4] *[http://wiki.openstreetmap.org/wiki/Map\\_Features](http://wiki.openstreetmap.org/wiki/Map_Features)*