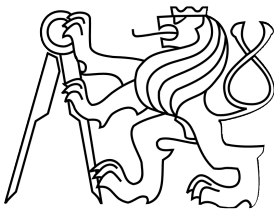
České vysoké učení technické



Fakulta stavební
Katedra mapování a kartografie

DOKUMENTACE

Úvod do zpracování prostorových dat

Skupina B

Barbora Faitová
Alžbeta Gardoňová
Anna Šubrtová

Letní semestr 2010/2011

Obsah

1. Úvod

Tato dokumentace vznikla v rámci předmětu Úvod do zpracování prostorových dat (153UZPD), který je zaměřen na zpracování (geo)prostorových dat, geoprostorové databáze, skladování geoprostorových dat a jejich zpracování. Cvičení jsou věnována práci s PostGIS a SpatiaLite a jsou zakončena projektem. Tento předmět volně navazuje na předmět Databázové systémy.

1.1. Zadání

- Navrhněte a vytvořte tematické vrstvy (např. vodní toky, vodní plochy, lesy, silnice, železnice apod.) na základě dat OSM (viz cvičná databáze pgis_student schéma osm). Pro tento účel byla na serveru 'geo102' založena databáze pgis_uzpd.
- Aplikujte testy datové integrity a odstraňte případné nekonzistence v datech.
- Vytvořte tutoriál pro výuku PostGIS - tj. sadu atributových a prostorových dotazů nad databází pgis_uzpd.

1.2. Stručně o OpenStreetMap (OSM)

OpenStreetMap je projekt zaměřený na vytváření svobodných geografických dat. U většiny ostatních volně dostupných map je ale užívání technicky a právně omezeno. Proto vznikl tento projekt, aby umožnil lidem volně nakládat s geografickými daty, používat je neobvyklými způsoby a v neposlední řadě, aby byla data dostupná v aktualizované a platné podobě bez dalších nákladů a omezení.

(zdroj: http://wiki.openstreetmap.org/wiki/Cz:Main_Page)

2. Tematické vrstvy

Protože tento projekt se opakuje již několikátý rok, nebylo jednoduché vybrat takové téma, abychom neopakovaly témata z minulých ročníků. Nechaly jsme se

inspirovat předmětem Katastr nemovitostí, ve kterém byla probírána ochranná pásma. Další věc, která nás na tomto tématu zaujala je, že mnoho lidí o tomto tématu nemá dostatek informací a přitom se jedná o velmi důležitou věc například při koupi pozemku a podobně. Nevýhodou tohoto tématu jsou poměrně časté změny, výjimky a někdy nejednoznačnost informací vycházejících z norem. V ochranných pásech se často nesmí nacházet různé stavby, provádět zemní práce, vysazovat zeleň a podobně.

Původně se nám zalíbila vrstva elektrické vedení, později jsme však zjistily, že tato data nejsou vhodná z důvodu velkého množství dat, nerovnoměrně zmapované České Republiky a nedostatečného označení dat. S nedostatečným označením jsme se potýkaly i u potrubí, které nakonec také nebylo do projektu zahrnuto. Nakonec byly vybrány letiště, elektrostatice, zdroje podzemní vody a silniční síť České Republiky. Tyto vrstvy byly ještě doplněny vrstvami ze schématu GIS1 jako je polygonová vrstva obcí, vrstva KLTM50 a vrstva hranice České Republiky, se kterými jsme zvyklí pracovat v rámci některých předmětů.

2.1. Tvorba vrstev

Vrstvy byly získány z nám zpřístupněné databáze pgis_uzpd ze schémat osm a gis1. Naše vrstvy byly ukládány ve stejné databázi do schématu b11 (skupina a rok projektu). Pro ulehčení práce a hlavně časové urychlení jsou vrstvy vytvořeny již s buffrem ochranného území, tudíž se z linií a bodů staly plošné polygony. V následujících odstavcích jsou podrobnější informace a jednotlivé příkazy pro vytvoření vrstev. U každé vrstvy bylo zajištěno, aby obsahovala geometrii objektu (vždy se jedná o sloupeček geom) a primární klíč. Pro funkčnost následujících řádků je ještě důležité mít správně nastavená práva. To je možné následujícím příkazem.

```
SET search_path TO b11, public, osm, gis1
```

2.1.1. Dálnice

Silnice byly rozděleny do několika částí a to hlavně z důvodu rozdílných ochranných pásem. Spornou částí jsou však nájezdy, které jsou vedeny zvlášť, ty mají buďto označení vyšší třídy cesty na kterou navazují anebo jsou označeny „null“. Tato problematika, do jaké kategorie tyto úseky patří,

není jednoznačně vyřešena ani v odborné literatuře. Z tohoto důvodu byly tyto kategorie vynechány.

Dálnice byly vybrány z tabulky czech_line, kde typ silnice (highway) byl označen jako motorway. Ochranné pásmo kolem dálnice je stanoveno na 100m. Tabulka byla vytvořena následujícím příkazem.

```
CREATE TABLE dialnice AS

SELECT COUNT(osm_id) AS pocet_usekov, ref AS nazov ,
ST_Area(ST_Union(ST_Buffer(way, 100))) AS plocha,
ST_Perimeter(ST_Union(ST_Buffer(way,100 ))) AS obvod,
ST_Length(ST_Union(way)) AS dlzka, ST_Union(ST_Buffer(way,
100)) AS geom

FROM osm.czech_line

WHERE highway = 'motorway'

GROUP BY ref;
```

Následně byl tabulce přidán sloupeček, do kterého byl vložen primární klíč.

```
ALTER TABLE dialnice ADD COLUMN gid serial;

ALTER TABLE dialnice ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

```
CREATE INDEX dialnice_geom ON dialnice USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('dialnice'::regclass);
```

2.1.2. Silnice 1. třídy

Další skupinou jsou silnice 1. třídy. V tabulce czech_line byly označeny ve sloupci highway jako primary. Tyto silnice mají ochranné pásmo 50m. Tabulka byla vytvořena následujícím příkazem.

```
CREATE TABLE silnice_1 AS
```

```
SELECT COUNT(osm_id) AS pocet_usekov,  ref AS nazov ,
ST_Area(ST_Union(ST_Buffer(way, 50))) AS plocha,
st_perimeter(ST_Union(ST_Buffer(way, 50))) AS obvod,
ST_Length(ST_Union(way)) as dlzka, ST_Union(ST_Buffer(way, 50))
AS geom

FROM osm.czech_line

WHERE highway = 'primary'

GROUP BY ref ;
```

Následně byl tabulce přidán sloupeček, do kterého byl vložen primární klíč.

```
ALTER TABLE silnice_1 ADD COLUMN gid serial;

ALTER TABLE silnice_1 ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

```
CREATE INDEX silnice_1_geom ON silnice_1 USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('silnice_1'::regclass);
```

2.1.3. Silnice 2. a 3. třídy

Poslední tabulkou jsou silnice 2. a 3. třídy. Tyto třídy nejsou rozděleny z důvodu stejného ochranného pásma 15m. Tabulka byla vytvořena následujícím příkazem.

```
CREATE TABLE silnice_2_3 AS

SELECT COUNT(osm_id) AS pocet_usekov, ref AS nazov ,
ST_Area(ST_Union(ST_Buffer(way, 15))) AS plocha,
ST_Perimeter(ST_Union(ST_Buffer(way, 15))) AS obvod,
AS_Length(ST_Union(way)) AS dlzka, ST_Union(ST_Buffer(way, 15))
AS geom

FROM osm.czech_line

WHERE highway = 'secondary'

OR highway = 'tertiary'
```

```
GROUP BY ref ;
```

Následně byl tabulce přidán sloupeček, do kterého byl vložen primární klíč.

```
ALTER TABLE silnice_2_3 ADD COLUMN gid serial;
ALTER TABLE silnice_2_3 ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

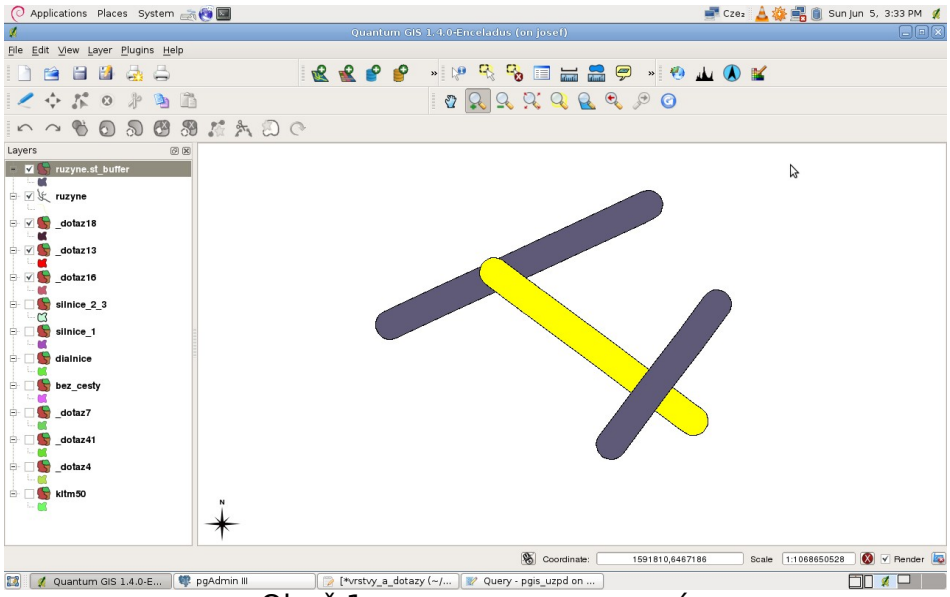
```
CREATE INDEX silnice_2_3_geom ON silnice_2_3 USING gist (geom);
```

Aktualizace metadatové tabulky.

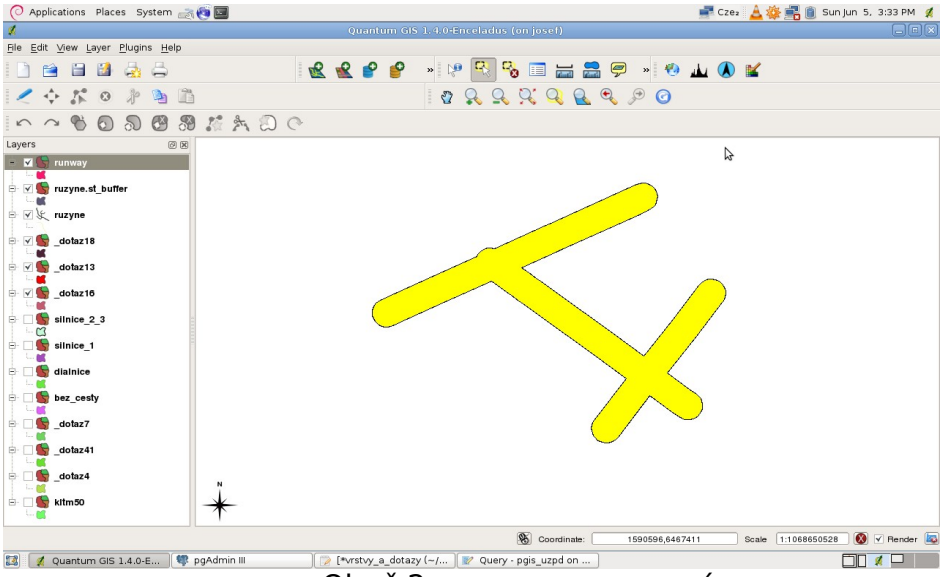
```
SELECT populate_geometry_columns('silnice_2_3'::regclass);
```

2.1.4. Runway

Po podrobnějším prozkoumání této vrstvy byla zjištěna značná nekonzistence dat. Dostupnými informacemi byly jenom osm_id, ref, atribut aeroway s hodnotou „runway“ a geometrie prvku. Některé přistávací plochy vůbec nebyly označeny, jiné měly duplicitní označení i přesto, že se nacházely v různých částech České Republiky. Data bez potřebného označení byla z tabulky vynechána. Protože se přistávacích ploch v České Republice mnoho nevyskytuje, bylo možné zbylé problémy řešit ručním přezkoumáním dat za pomoci grafického znázornění. Další problém se vyskytl po provedení funkce buffer, protože některé letištní dráhy se kříží a bylo nutné je spojit v jednu letištní plochu. Zjištění překrývajících se ploch bylo opět zjištěno ručním prohledáním grafického znázornění. Letištní plochy jsou v tabulce czech_line označeny ve sloupci aeroway jako runway. Výsledná kontrola byla doplněna ještě prostorovým dotazem ST_INTERSECTS, kde byla vstupem ta samá vrstva. Prázdný výstup takového dotazu potvrzoval správnost vrstvy. Níže uvedené příkazy popisují celý postup tvorby výsledné tabulky.



Obr.č.1 – runway neupravená



Obr.č.2 – runway upravená

Nejprve byly vybrány letištní plochy větší než 1800m, protože mají větší ochranné pásmo. Byly seskupeny podle hodnoty „ref“.

```
CREATE TABLE zoradene2 AS

SELECT ref, COUNT(osm_id) AS pocet_usekov

FROM czech_line

WHERE aeroway = 'runway'

AND ST_Length(way) >= 1800

GROUP BY ref

ORDER BY pocet_usekov;
```

U letišť, která jsou v tabulce zoradene2 jenom jednou, jsou vytvořeny buffery. Přímo jsou zde vyjmenována letiště, která mají nějaký problém a jsou řešena samostatně. Pokud se jedná o vynechání podle vícenásobného parametru „ref“, jsou blíže určena pomocí osm_id.

```
CREATE TABLE runway AS
SELECT zoradene2.ref, COUNT(zoradene2.ref),
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM zoradene2
RIGHT OUTER JOIN czech_line
ON zoradene2.ref = czech_line.ref
WHERE pocet_usekov = 1
AND aeroway = 'runway'
AND ST_Length(way) >= 1800
AND zoradene2.ref NOT IN('13-31', '06-24', '04-22', '06/24',
'09/27','03C/21C','03R/21L','03L/21R')
AND osm_id NOT IN (28042379, 28042372, 25451388, 25451389,
89985986)
GROUP BY zoradene2.ref
```

Zde jsou postupně řešena letiště, která se překrývají, a každá skupina je řešena samostatně, protože jediný společný atribut je aeroway. Upravená data jsou vložena do tabulky runway.

```
INSERT INTO runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) >= 1800
AND ref IN ('13-31', '06-24', '04-22')
GROUP BY aeroway)
```

```
INSERT INTO runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line
WHERE aeroby = 'runway'
AND ST_Length(way) >= 1800
AND ref IN ('03C/21C','03R/21L','03L/21R')
GROUP BY aeroway)
```

```
INSERT INTO runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line
WHERE aeroby = 'runway'
AND ST_Length(way) >= 1800
AND osm_id IN (28042379, 28042372)
GROUP BY aeroway)
```

```
INSERT INTO runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line

WHERE aeroby = 'runway'
AND ST_Length(way) >= 1800
AND osm_id IN (25451388, 25451389)
GROUP BY aeroway)
```

U objektů, kde je stejné „ref“, ale jiná poloha, tudíž se domníváme, že je to chyba v datech, jsou plochy rozděleny podle osm_id.

```
INSERT INTO runway(
SELECT count(ref) AS ref, COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line
WHERE aeroby = 'runway'
AND ST_Length(way) >= 1800
AND ref IN ('09/27', '06/24')
GROUP BY osm_id)
```

```
INSERT INTO runway(
SELECT count(ref) AS ref, COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 300))) AS plocha,
ST_Union(ST_Buffer(way, 300)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) >= 1800
AND osm_id IN(102406127, 25451625, 48499285, 48503047,
53569376, 23048716, 19853714, 19853713, 75134835, 16280387))
GROUP BY osm_id)
```

Zde je vytvořena pomocná tabulka, kam jsou vloženy runway menší než 1800 m, které mají ochranné pásmo 150m. Používá se opět tabulka

seřazených objektů. Celý následující postup je stejný jako u velkých letišť. Bylo by možné záznamy rovnou přidávat do tabulky runway, ale kvůli průběžné kontrole vznikajících dat, byla zvolená samostatná tabulka, která se po naplnění pomocí INSERT přidala do tabulky runway.

```
CREATE TABLE zoradene AS
SELECT ref, COUNT(osm_id) AS pocet_usekov
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
GROUP BY ref
ORDER BY pocet_usekov;
```

```
CREATE TABLE mala_runway AS
SELECT zoradene.ref, COUNT(zoradene.ref),
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM zoradene
RIGHT OUTER JOIN czech_line
ON zoradene.ref = czech_line.ref
WHERE pocet_usekov = 1
AND aeroway = 'runway'
AND ST_Length(way) < 1800
AND zoradene.ref NOT IN
('18','36','10R/28L','10L/28R','05L/23R','05R/23L','09R/27L','1
6R/34L','07L/25R','07R/25L')
AND osm_id NOT IN (97169088, 97169087, 55985240, 55985239,
89366169, 89366159)
GROUP BY zoradene.ref
```

Zde jsou postupně řešeny malé letištní plochy, které se překrývají. Upravená data jsou vložena do tabulky mala_runway.

```
INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND ref IN('10R/28L', '10L/28R')
GROUP BY aeroway)
```

```
INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
```

```
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND ref IN('05R/23L', '05L/23R')
GROUP BY aeroway)


INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND ref in('18', '36')
GROUP BY aeroway)


INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND ref in('07R/25L','07L/25R')
GROUP BY aeroway)


INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND osm_id IN('97169088', '97169087')
GROUP BY aeroway)


INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND osm_id IN('55985240', '55985239')
GROUP BY aeroway)
```

```
INSERT INTO mala_runway(
SELECT COUNT(ref) AS ref , COUNT(osm_id) AS count,
ST_Area(ST_Union(ST_Buffer(way, 150))) AS plocha,
ST_Union(ST_Buffer(way, 150)) AS geom
FROM czech_line
WHERE aeroway = 'runway'
AND ST_Length(way) < 1800
AND osm_id IN('89366169', '89366159')
GROUP BY aeroway)
```

Následně jsou data z tabulky mala_runway vložena do tabulky runway.

```
INSERT INTO runway (

SELECT *

FROM mala_runway)
```

Následně byl tabulce přidán sloupeček, do kterého byl vložen primární klíč.

```
ALTER TABLE runway ADD COLUMN gid serial;

ALTER TABLE runway ADD PRIMARY KEY(gid);
```

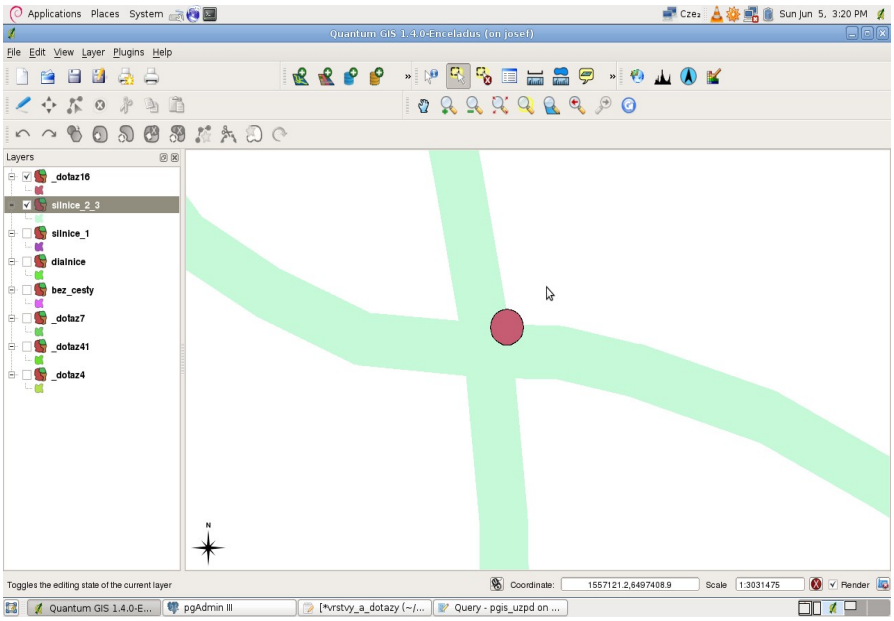
Vytvoření prostorového indexu.

```
CREATE INDEX runway_geom ON runway USING gist (geom);

SELECT populace_geometry_columns('runway'::regclass);
```

2.1.5. Podzemní voda

Další vybranou tabulkou je zdroj podzemní vody. Původně bodová vrstva umístěná v tabulce czech_point je pomocí funkce buffer převedena na plochu. Zdroj podzemní vody je označen ve sloupečku natural pojmem spring.



Obr.č.3. - umístění zdoja podzemní vody v křižovatce

Tabulka byla vytvořena následujícím příkazem.

```
CREATE TABLE podz_voda AS

SELECT osm_id, ST_Area(ST_Buffer(way, 10)) AS plocha,
(ST_Buffer(way, 10)) AS geom

FROM osm.czech_point

WHERE "natural" = 'spring';
```

Zde je jako primární klíč převzat sloupeček osm_id.

```
ALTER TABLE podz_voda ADD PRIMARY KEY(osm_id);
```

Vytvoření prostorového indexu.

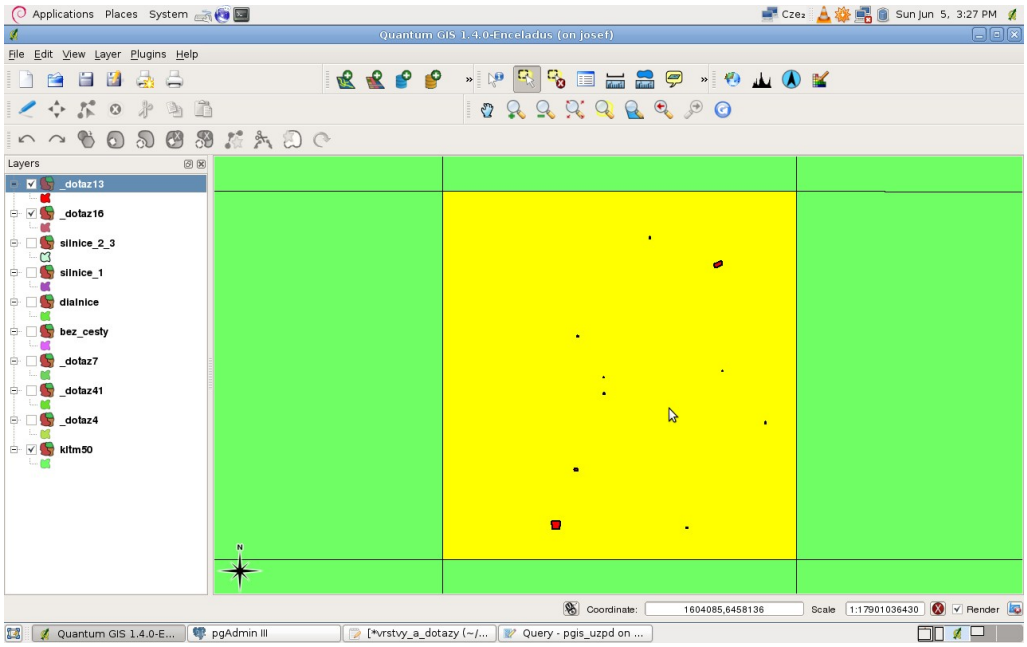
```
CREATE INDEX podz_voda_geom ON podz_voda USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('podz_voda'::regclass);
```

2.1.6. Elektrostatice

Poslední vrstvou přidanou z OSM jsou elektrostatice. Ty jsou označeny v tabulce czech_polygon ve sloupečku power označením station. Polygon je rozšířen o ochranné území 20m.



Obr.č.4 - rozmístění a jednotlivé plochy elektrostanic na daném mapovém listu

Tabulka byla vytvořena následujícím příkazem.

```
CREATE TABLE el_stanice AS

SELECT osm_id, ST_Area(ST_Buffer(way, 20)) AS plocha,
(ST_Buffer(way, 20)) AS geom

FROM osm.czech_polygon

WHERE power = 'station';
```

Zde je jako primární klíč převzat sloupeček osm_id.

```
ALTER TABLE el_stanice ADD PRIMARY KEY(osm_id);
```

Vytvoření prostorového indexu.

```
CREATE INDEX el_stanice_geom ON el_stanice USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('el_stanice'::regclass);
```

2.1.7. Obce

Tato tabulka byla převzata ze schématu gis1 následujícím příkazem.

```
CREATE TABLE obce AS

SELECT *

FROM gis1.obce;
```

Tabulka byla upravena kvůli transformaci souřadnicového systému, podle návodu od vyučujícího.

```
UPDATE obce

SET geom = ST_Transform(geom, 900913);
```

Zde je jako primární klíč převzat sloupeček gid.

```
ALTER TABLE obce ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

```
CREATE INDEX obce_geom ON obce USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('obce'::regclass);
```

2.1.8. KLTM 50

Vrstva KLTM 50 byla převzata ze schématu gis1 následujícím příkazem.

```
CREATE TABLE kltm50 AS

SELECT *

FROM gis1.kltm50
```

Tabulka byla upravena kvůli transformaci souřadnicového systému, podle návodu od vyučujícího.

```
UPDATE kltm50

SET geom = ST_Transform(geom, 900913);
```

Zde je jako primární klíč převzat sloupeček gid.

```
ALTER TABLE kltm50 ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

```
CREATE INDEX kltm50_geom ON kltm50 USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('kltm50'::regclass);
```

2.1.9. Hranice ČR

Vrstva hranice ČR byla převzata ze schématu gis1 následujícím příkazem.

```
CREATE TABLE cr AS
SELECT ST_BuildArea(ST_Union(geom)) AS geom
FROM obce
GROUP BY pomoc
```

Zde je jako primární klíč převzat sloupeček gid.

```
ALTER TABLE cr ADD PRIMARY KEY(gid);
```

Vytvoření prostorového indexu.

```
CREATE INDEX cr_geom ON cr USING gist (geom);
```

Aktualizace metadatové tabulky.

```
SELECT populate_geometry_columns('cr'::regclass);
```

3. Prostorové dotazy

Při tvorbě dotazů jsme se snažili zaměřit na různorodost. Na konci příkazu je vždy uvedena správná odpověď a čas zpracování dotazu. Dotazy byly testovány pomocí PgAdmin3 s připojením na schéma b11 na serveru Josef. Správnost dotazu byla kontrolována pomocí vytvoření nové vrstvy a jejím zobrazení v QGIS. Některé dotazy by dle úvahy měly mít jasnou odpověď, ale výsledek není vždy jasný. Může jít o špatně zmapované prvky nebo taky o časovou posloupnost při budování, které se nám dnes zdá nelogické. Při některých takovýchto dotazech byla tato nesrovnalost zobrazena na některém prvku.

3.1. Určete počet listů KLTM50, na kterých se nachází objekty tabulky runway a kolik objektů (runway) se nachází v ČR.

```
SELECT COUNT( DISTINCT kltm50.gid ) AS pocet_listov,
COUNT(DISTINCT runway.gid) AS pocet_runway

FROM kltm50

JOIN runway

ON ST_Intersects(runway.geom, kltm50.geom)


Answer: ( 42 ; 38 )

Total query runtime: 224 ms.
```

3.2. Určete počet objektů tabulky runway, které alespoň částí svého ochranného území leží ve více obcích.

```
SELECT (SELECT COUNT(DISTINCT runway.gid) FROM runway) -
COUNT(DISTINCT runway.gid)

FROM obce

JOIn runway
```

```
ON ST_Contains(obce.geom, runway.geom);
```

Answer: (22)

Total query runtime: 21 ms.

3.3. **Určete plochu ochranných pásem všech objektů tabulky runway a průměrnou hodnotu plochy ochranných pásem runway (v km²).**

```
SELECT ROUND(SUM(ST_Area(runway.geom))/1e6) AS plocha_letisk,
SUM(st_area(runway.geom))/(1e6*COUNT(gid))  AS
priemer_procenta
FROM runway
```

Answer: (77 ; 2.03)

Total query runtime: 14 ms.

3.4. Určete počet ochranných pásem objektů runway, které se křižují s ochrannými pásmy silnic 2. a 3. třídy.

```
SELECT COUNT(DISTINCT runway.gid)
FROM silnice_2_3
JOIN runway
ON ST_Intersects(runway.geom, silnice_2_3.geom)
```

Answer: (10)

Total query runtime: 2927 ms.

3.5. Určete počet obci, ve kterých se nachází ochranné pásmo silnic 1. třídy.

```
SELECT COUNT(DISTINCT obce.gid)
FROM silnice_1
JOIN obce
ON ST_Intersects(obce.geom, silnice_1.geom)
```

Answer: (1518)

Total query runtime: 6812 ms.

3.6. Určete počet křižování ochranného pásma dálnice D5 (nazov) s ochrannými pásmy silnic 1. třídy.

```
SELECT COUNT( dialnice.gid)
FROM silnice_1
JOIN dialnice
ON ST_Intersects(dialnice.geom, silnice_1.geom)
WHERE dialnice.nazov='D5'
```

Answer: (5)

Total query runtime: 72 ms.

3.7. Určete počet obcí, ve kterých se nachází ochranná pásma cest všech kategorií (dálnice, silnice 1., 2., 3. třídy).

```
SELECT COUNT(DISTINCT obce.gid)
FROM silnice_1
JOIN obce
ON ST_Intersects(obce.geom, silnice_1.geom)
JOIN dialnice
ON ST_Intersects(obce.geom, dialnice.geom)
```

```
JOIN  silnice_2_3
ON ST_Intersects(obce.geom, silnice_2_3.geom)
```

Answer: (52)

Total query runtime: 14034 ms.

3.8. Určete počet obcí, v kterých se nenachází ochranná pásma silnic 2. a 3. třídy.

```
SELECT (
SELECT COUNT(obce.gid) FROM obce)-COUNT(DISTINCT(obce.gid))
FROM obce
JOIN silnice_2_3
ON ST_Intersects(obce.geom, silnice_2_3.geom)
```

Answer: (42)

Total query runtime: 458708 ms.

3.9. Určete vzdálenost nejvýchodnější a nejzápadnější ranwaye (použijte centroidy, odpověď v km).

```
SELECT ROUND(ST_Distance(
     (SELECT geom
     FROM runway
     ORDER BY ST_x(CENTROID(runway.geom))  DESC LIMIT 1),
     (SELECT geom
     FROM runway
     ORDER BY st_x(CENTROID(runway.geom)) ASC LIMIT 1))/1e3)
```

Answer: (635)

Total query runtime: 14114 ms.

3.10. Určete součet ploch ochranných pásem dálnice a 1. třídy a jejich procentuální vyjádření v rámci plochy ČR. (plochu v m^2)

```
SELECT SUM(ST_Area(dialnice.geom)) + SUM(ST_Area(silnice_1.geom)),

    (SUM(ST_Area(dialnice.geom)) +
SUM(ST_Area(silnice_1.geom)))/SUM(ST_Area(cr.geom))*100

FROM dialnice

JOIN cr

ON ST_Intersects(cr.geom, dialnice.geom)

JOIN  silnice_1

ON ST_Intersects(cr.geom, silnice_1.geom)
```

```
Answer: ( 37626463628.6449 ; 0.0179500656356264 )

Total query runtime: 749 ms.
```

3.11. Určete nejmenší vzdálenost mezi runway a elektrostanicí, gid runway a osm_id elektrostatice.

```
SELECT ROUND( Max_Distance(runway.geom, el_stanice.geom)) AS vzd,
runway.gid, El_stanice.osm_id

FROM  runway

JOIN el_stanice

ON ST_Disjoint(runway.geom, el_stanice.geom)

 ORDER BY vzd ASC LIMIT 1
```

```
Answer: ( 3042 ; 14; 42939782 )

Total query runtime: 395 ms.
```

3.12. Najděte list KLTM50 (nazev), na kterém se nachází nejvyšší počet elektrostanic, určete jejich počet.

```
SELECT kltm50.nazev, COUNT(DISTINCT el_stanice.osm_id) AS pocet

FROM kltm50

JOIN el_stanice

ON ST_Intersects(kltm50.geom, CENTROID(el_stanice.geom))

GROUP BY nazev

ORDER BY pocet DESC LIMIT 1
```

```
Answer: ("M-33-65-D";10)

Total query runtime: 22 ms.
```

3.13. Určete plochu ochranných pásem a počet elektrostanic nacházejících se na listě KLTM50 "M-33-65-D" (nazev).

```
SELECT COUNT(DISTINCT el_stanice.osm_id) AS pocet,
SUM(ST_Area(el_stanice.geom))

FROM kltm50

JOIN el_stanice

ON ST_Intersects(kltm50.geom, CENTROID(el_stanice.geom))

WHERE nazev='M-33-65-D'
```

```
Answer: (10;730512.443806648)

Total query runtime: 14 ms.
```

3.14. Určete počet ochranných pásem elektrostanic, které se nacházejí v ochranném pásu cest 1. třídy.

```
SELECT COUNT(DISTINCT el_stanice.osm_id)
```

```
FROM silnice_1

JOIN el_stanice

ON ST_Intersects(el_stanice.geom, silnice_1.geom)
```

Answer: (10)

Total query runtime: 2528 ms.

3.15. Určete název obce, ve které se nachází elektrostatice 28477054 (osm_id).

```
SELECT nazev

FROM obce

JOIN el_stanice

ON ST_Intersects(el_stanice.geom, obce.geom)

WHERE osm_id='28477054'
```

Answer: "Dolní Lukavice"

Total query runtime: 12 ms.

3.16. Určete počet ochranných pásem zdroje podzemní vody, které se alespoň částečně překrývají s ochranným pásmem silnic 2. a 3. třídy.

```
SELECT COUNT(DISTINCT podz_voda.osm_id)

FROM  podz_voda

JOIN silnice_2_3

ON ST_Intersects(podz_voda.geom, silnice_2_3.geom)
```

Answer: (6)

```
Total query runtime: 40778 ms.
```

3.17. Určete název KLTM50 nejjižnějšího ochranného pásma zdroje podzemní.

```
SELECT DISTINCT(nazev)

FROM kltm50

JOIN podz_voda

ON ST_Intersects( (SELECT podz_voda.geom

     FROM podz_voda

ORDER BY ST_y(CENTROID(podz_voda.geom)) ASC LIMIT 1), kltm50.geom
)
```

```
Answer: ("M-33-114-C")

Total query runtime: 21 ms.
```

3.18. Najděte název obce, kde se nachází runway s největším poměrem plochy ochranného pásma k obvodu. Vypište název a hledaný poměr.

```
SELECT nazev, ST_Area(runway.geom)/st_perimeter(runway.geom) AS
pomer

FROM runway

JOIN obce

ON ST_Intersects(obce.geom, runway.geom)

ORDER BY pomer DESC LIMIT 1
```

```
Answer: ("Kunovice";339.195296803988)

Total query runtime: 23 ms.
```

3.19. Určete souřadnice zdroje podzemní vody v obci Horní Kozolupy. (použijte centroid)

```
SELECT st_astext(CENTROID(podz_voda.geom))

FROM podz_voda

JOIN obce

ON ST_Intersects(CENTROID(podz_voda.geom), obce.geom)

WHERE nazev = 'Horní Kozolupy'
```

```
Answer: ("POINT(1443969.56495223 6422518.29573358)")

Total query runtime: 52 ms.
```

3.20. Určete gid a vzdálenost nejbližší runway od obce Horní Kozolupy. (pro obě vrstvy použijte centroid)

```
SELECT ref, st_distance(CENTROID(obce.geom),
CENTROID(runway.geom)) AS vzd

FROM runway

JOIN obce

ON ST_Disjoint(CENTROID( obce.geom), CENTROID(runway.geom))

WHERE nazev = 'Horní Kozolupy'

ORDER BY vzd ASC LIMIT 1
```

```
Answer: (22;45781.7490865062)

Total query runtime: 12 ms.
```

3.21. Určete, v jaké obci leží elektrostanice a letiště z otázky 11.

```
SELECT nazev
```

```
FROM obce

JOIN runway

ON ST_Contains(obce.geom,runway.geom)

JOIN el_stanice

ON ST_Contains(obce.geom,el_stanice.geom)

WHERE el_stanice.osm_id = (SELECT el_stanice.osm_id

        FROM runway

        JOIN el_stanice

        ON ST_DISJOINT(runway.geom, el_stanice.geom)

        ORDER BY ROUND(MAX_DISTANCE(runway.geom,
el_stanice.geom))

        LIMIT 1)

AND runway.gid = (SELECT runway.gid

        FROM runway

        JOIN el_stanice

        ON ST_DISJOINT(runway.geom, el_stanice.geom)

        ORDER BY ROUND(MAX_DISTANCE(runway.geom,
   el_stanice.geom))

        LIMIT 1);
```

```
Answer: ( "Praha" )

Total query runtime: 786 ms.
```

3.22. Kolik je v ČR dálnic a jaká je jejich průměrná délka?

```
SELECT COUNT(*), SUM(dlzka)/(COUNT(*)*1e3) as delka

FROM dialnice;
```

```
Answer: ( 10 ; 229.869734399394 )
Total query runtime: 12 ms.
```

4. Závěr

Naším cílem bylo vytvořit sadu prostorových dotazů, které se nebudou navzájem opakovat. Snažily jsme se, aby se již použité dotazy nestávaly dotazy novými přepsáním použité vrstvy a aby každý další dotaz byl řešen jiným způsobem než dotaz předchozí.

Problém použitých dat spočíval v nejednotném označení, kvůli kterému některá data nemohla být použita.

Při tvorbě dotazů jsme narazili na zajímavé naměty, které však museli být zjednodušeny, protože samotná doba vyhodnocování byla neúměrně dlouhá.