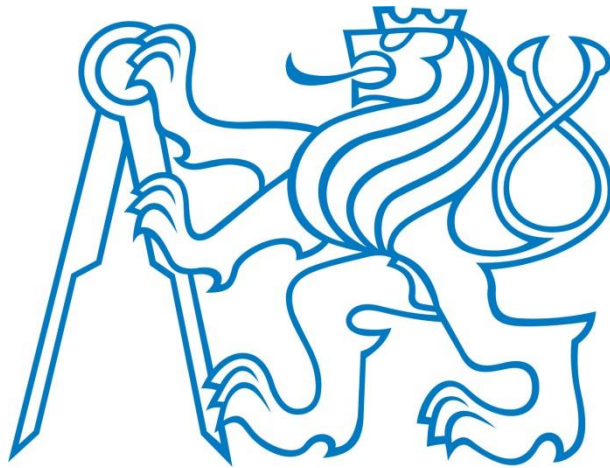


Úvod do zpracování prostorových dat

Semestrální projekt

Dokumentace



Skupina D

Martin Florian, Vojtěch Goby, Dominik Hladík

8. února 2016

Obsah

1	Zadání	2
2	Výběr tématu	2
3	Použitý software	2
4	Data	3
4.1	OSM	3
4.2	RÚIAN.....	6
4.3	Ruční sběr z maps.google.com.....	8
5	Dotazy	9
5.1	Atributové.....	9
5.2	Prostorové	9
6	pgRouting.....	11
6.1	Funkce.....	11
6.2	Síťové analýzy	14
7	Závěr	15

1 Zadání

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

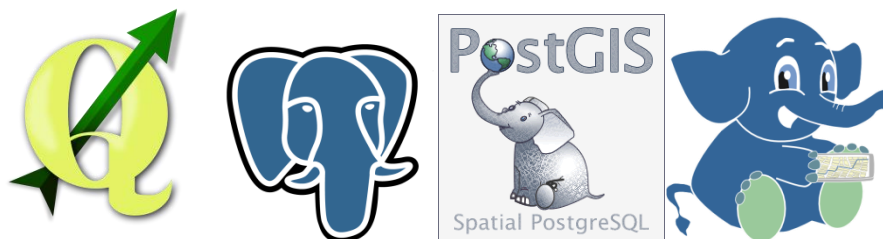
2 Výběr tématu

Jako téma pro tento projekt byl zvolen sport. Důvodem byla skutečnost, že byla k dispozici data v textové podobě pro vrstvu lední hokej.

Přehled	
název	hodnota
téma	sport
databáze	pgis_uzpd
schéma	d15
počet tabulek	7
EPSG	5514
datum vzniku	leden-únor 2016

3 Použitý software

- QGIS – Open Source geografický informační systém, který je k dispozici zdarma. K dispozici je [zde](#).
- Databáze PostgreSQL – Objektově-relační databázový systém, který je také zdarma. Na jeho vývoji se podílí komunita vývojářů a firem. Domovská stránka je na [této adrese](#).
- PostGIS – Je rozšířením objektově-relačního databázového systému PostgreSQL. Stránky projektu PostGIS jsou k dispozici [zde](#).
- pgRouting – Je rozšířením PostGISu pro řešení síťových analýz. Více informací o tomto rozšíření se můžete dozvědět [zde](#).



4 Data

Vrstvy byly získány z různých datových zdrojů, což samozřejmě znamenalo, že byly i v různých souřadnicových systémech.

Vrstvy			
název	typ	zdroj	EPSG
fotbal	bodová	OSM	3857
ledníhokeje	bodová	ruční sběr	4326
tenis	bodová	OSM	3857
silnice	liniová	OSM	3857
obce	plošná	RÚIAN	5514
okresy	plošná	RÚIAN	5514
staty	plošná	RÚIAN	5514

V případě ručního sběru bylo poněkud obtížné zjistit správný EPSG kód, ale nakonec se to podařilo. V datech jsme se setkali s těmito souřadnicovými systémy:

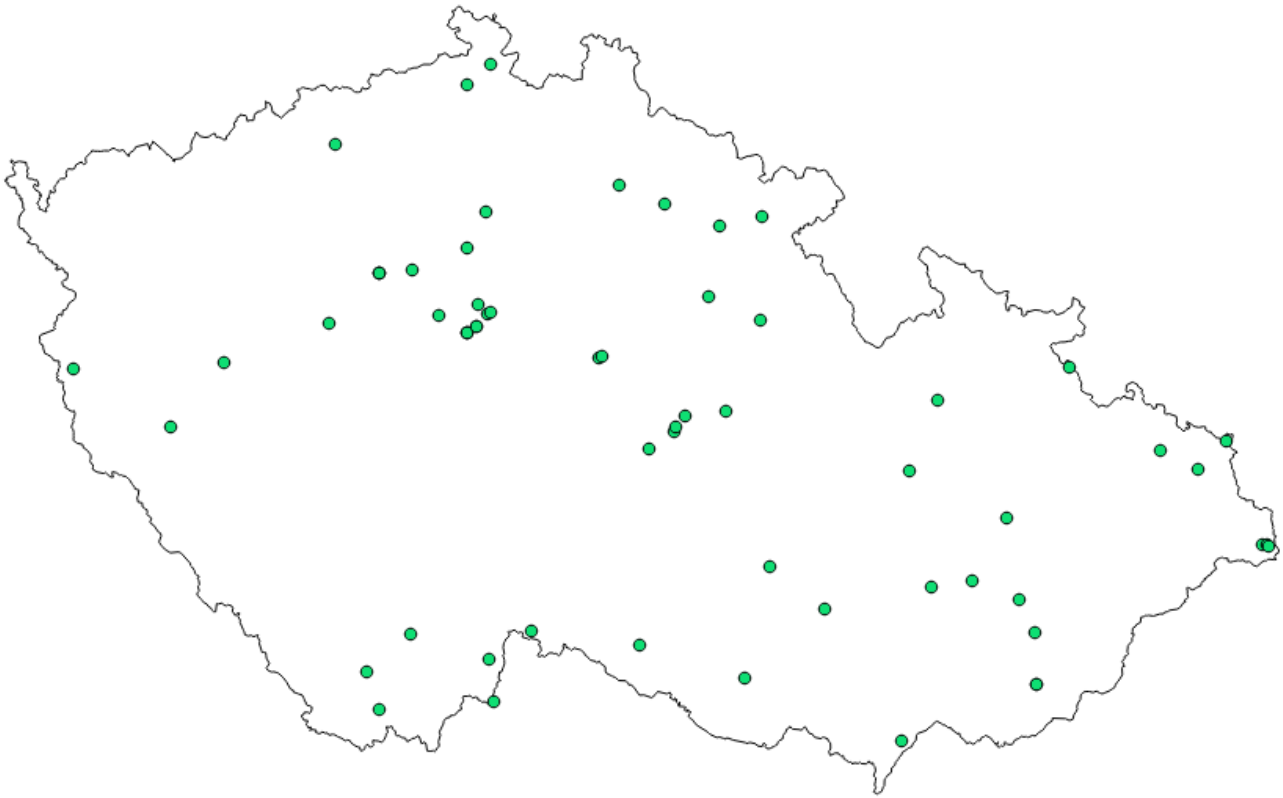
Souřadnicové systémy	
EPSG	souř. systém
3857	WGS84 Web Mercator
4326	WGS84
5514	S-JTSK / Krovak East North
https://www.epsg-registry.org	

4.1 OSM

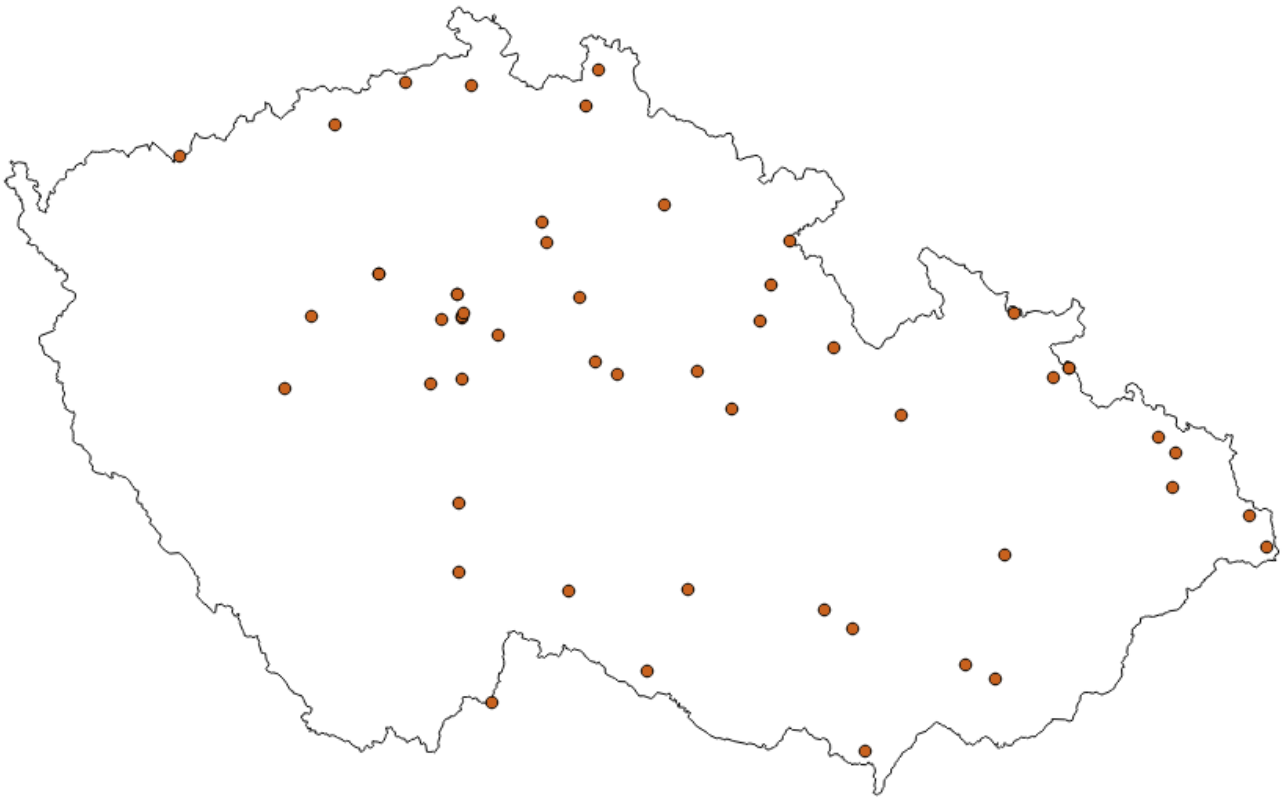
Data z OSM byla zkopírována výběrem sloupců a řádků z databáze. Dále musela být transformována do požadovaného souřadnicového systému. Původní geometrie byla smazána a do tabulky byl přidán sloupec s identifikátorem. Nad sloupcem s geometrií byl umístěn prostorový index a bylo znemožněno zadávání záznamu s prázdnou geometrií. Na závěr byla provedena kontrola dat.

```
CREATE TABLE [název tabulky] AS ([výběr dat z tabulky]);  
  
SELECT AddGeometryColumn([název tabulky], 'geom', 5514, [datový typ], 2);  
  
UPDATE [název tabulky] SET geom = ST_Transform(geomtmp, 5514);  
  
SELECT DropGeometryColumn([název tabulky], 'geomtmp');  
  
ALTER TABLE [název tabulky] ADD COLUMN id SERIAL;  
  
ALTER TABLE [název tabulky] ADD PRIMARY KEY (id);  
  
ALTER TABLE [název tabulky] ALTER COLUMN geom SET NOT NULL;  
  
CREATE INDEX [název indexu] ON [název tabulky] USING gist(geom);  
  
SELECT id, ST_IsValidReason(geom) AS reason FROM [název tabulky] WHERE ST_IsValid(geom) = FALSE;
```

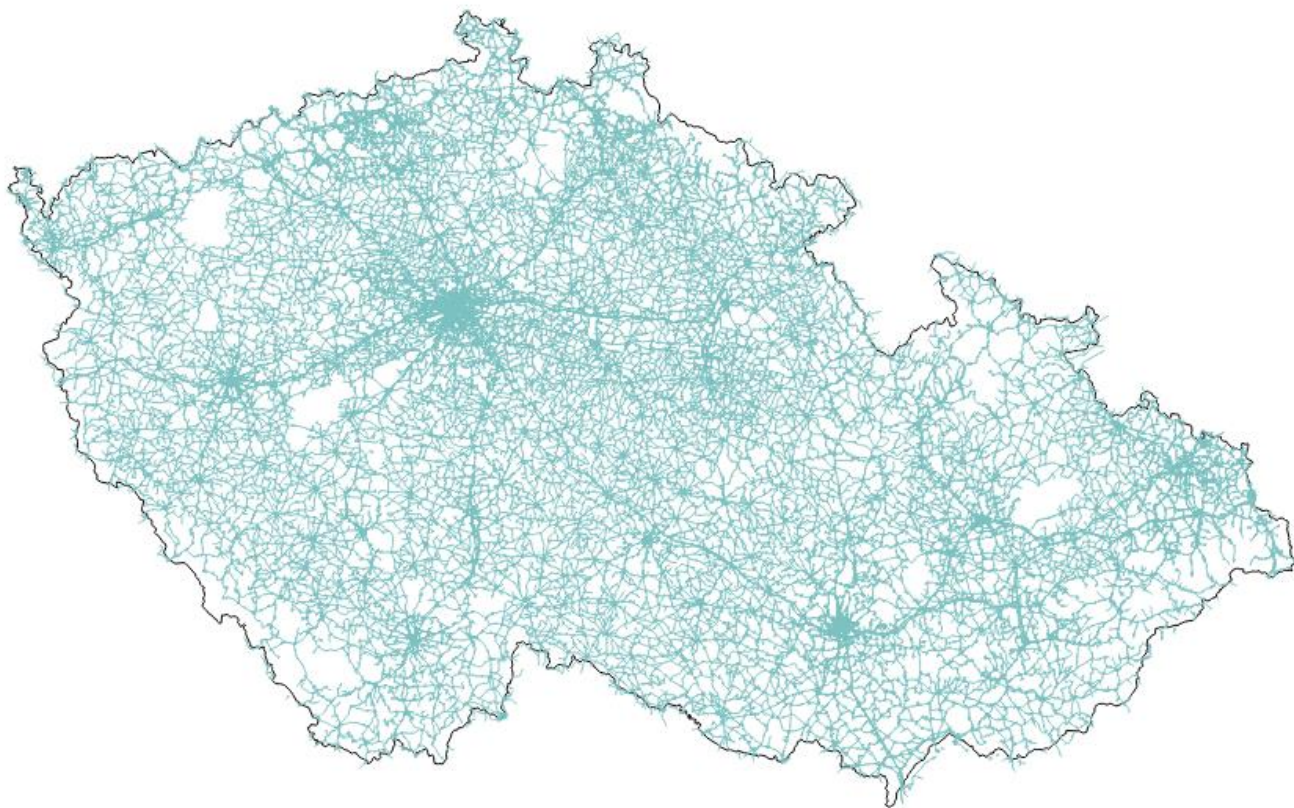
fotbal						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(Point,5514)	x			x	
id	int	x	x	x	x	nextval
nazev	text					
typ	text					



tenis						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(Point,5514)	x			x	
id	int	x	x	x	x	nextval
nazev	text					
typ	text					



silnice						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(LineString,5514)	x			x	
id	int	x	x	x	x	nextval
typ	text					



Pro výběr silnic z dat OSM bylo nutné si zjistit význam hodnot ve sloupci `osm_czechline.highway`.

OSM.czech_line	
sloupec: highway	význam
motorway	2 či víceprúdová silnice
trunk	významná silnice v rámci země
primary	silnice, která často spojuje velká města
secondary	silnice, která spojuje středně velká města
tertiary	silnice, která spojuje malá města
motorway_link	podjezd/nadjezd
trunk_link	podjezd/nadjezd
primary_link	podjezd/nadjezd
secondary_link	podjezd/nadjezd
tertiary_link	podjezd/nadjezd
http://wiki.openstreetmap.org/wiki/Key:highway	

4.2 RÚIAN

Data z RÚIAN již byla v požadovaném souřadnicovém systému, tudíž je již nebylo třeba transformovat. Ostatní kroky byly stejné jako při importu dat z OSM.

```
CREATE TABLE [název tabulky] AS ([výběr dat z tabulky]);

ALTER TABLE [název tabulky] ADD COLUMN id SERIAL;

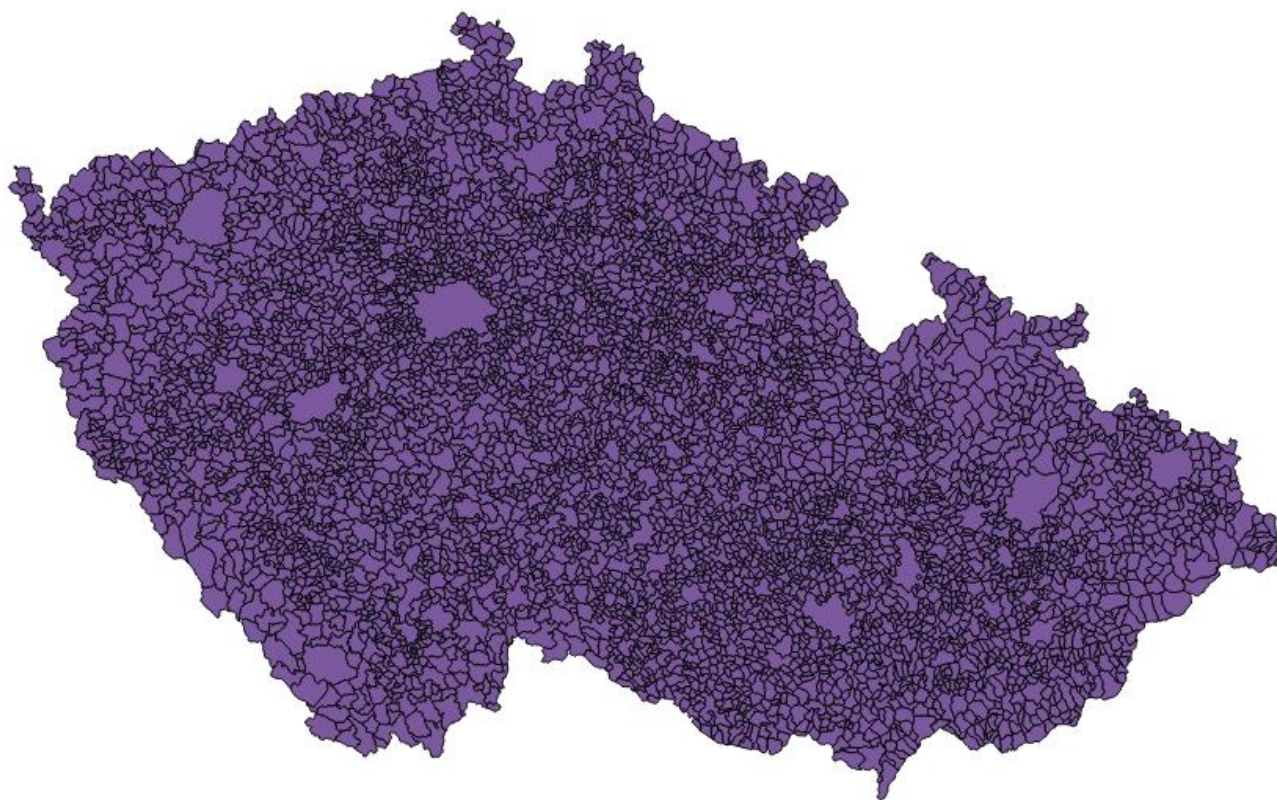
ALTER TABLE [název tabulky] ADD PRIMARY KEY (id);

ALTER TABLE [název tabulky] ALTER COLUMN geom SET NOT NULL;

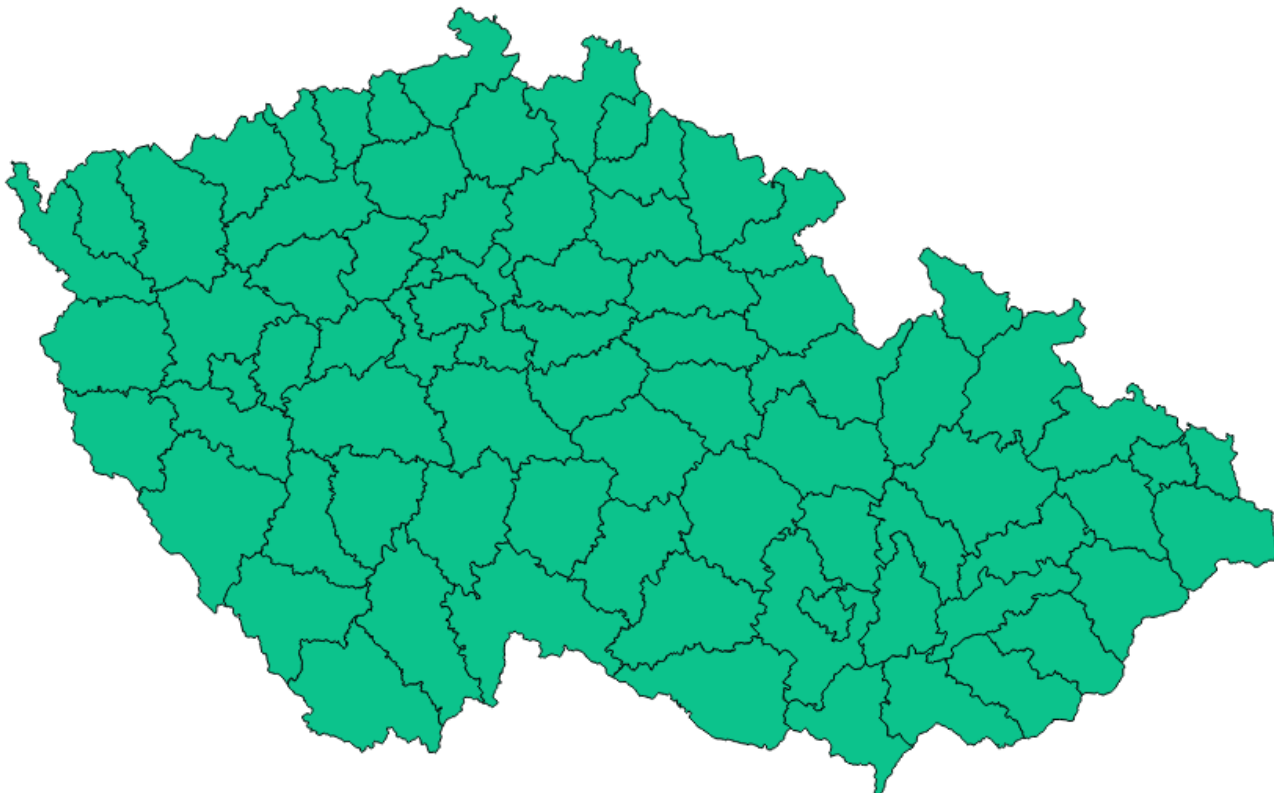
CREATE INDEX [název indexu] ON [název tabulky] USING gist(geom);

SELECT id, ST_IsValidReason(geom) AS reason FROM [název tabulky] WHERE ST_IsValid(geom) = FALSE;
```

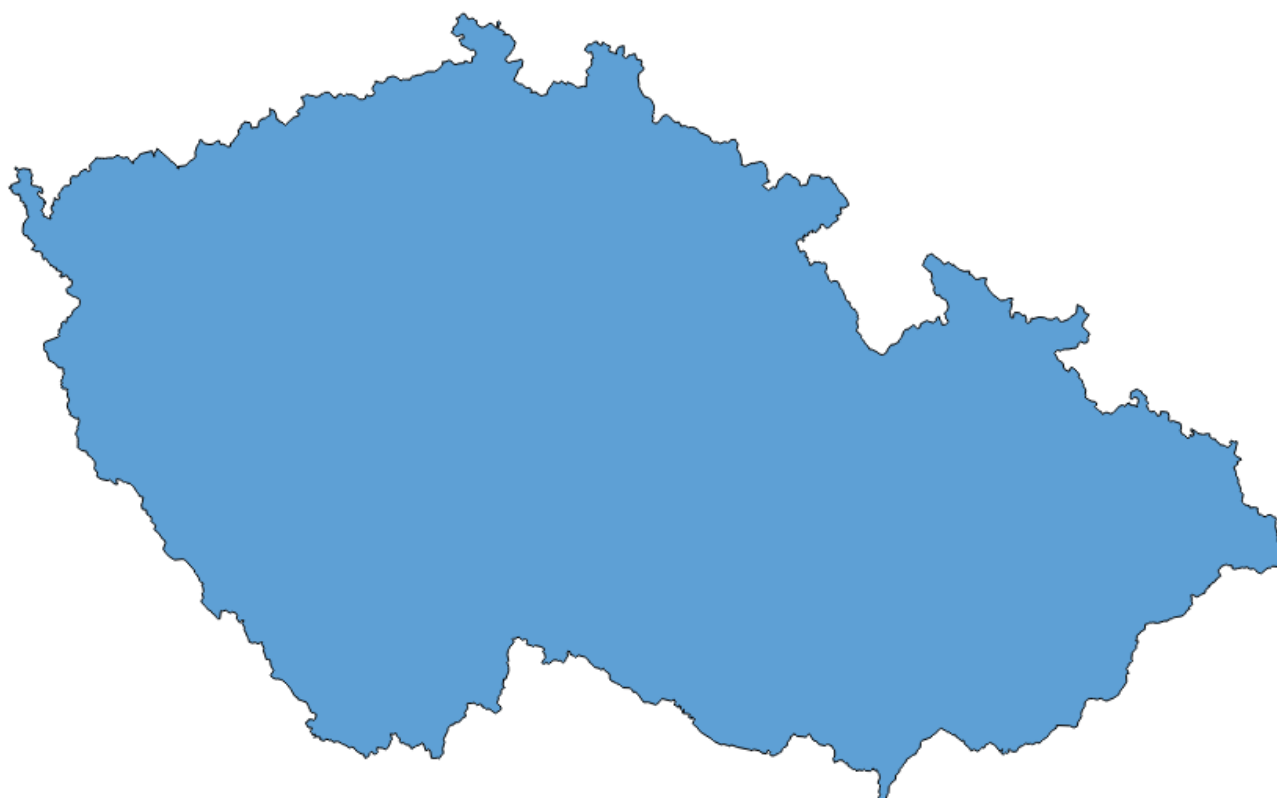
obce						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(MultiPolygon,5514)	x			x	
id	int	x	x	x	x	nextval
obec	text					



okresy						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(MultiPolygon,5514)	x			x	
id	int	x	x	x	x	nextval
okres	text					

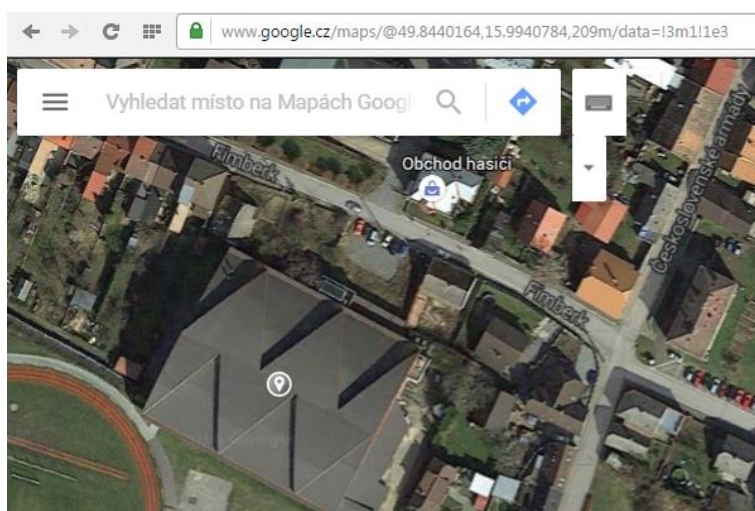


staty						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(MultiPolygon,5514)	x			x	
id	int	x	x	x	x	nextval
stat	text					



4.3 Ruční sběr z maps.google.com

Data o zimních stadionech byla pořízena v roce 2014 na maps.google.com. Souřadnice ve WGS84 jsou vidět na obrázku jako součást URL adresy.



Informace o klubu sídlícím v daném městě byla převzata z webu zimnistadiony.cz. Informace o zastřešení je vidět přímo na obrázku. Z dat v textovém souboru byly vytvořeny SQL „inserty“ pomocí programu MS EXCEL. Postup se od předchozích zdrojů dat lišil pouze v tom, že bylo nutné vytvořit sloupec s geometrií.

```
CREATE TABLE [název tabulky] ([seznam parameterů]);

INSERT INTO [název tabulky] ([názvy sloupců]) VALUES ([hodnoty]);

ALTER TABLE [název tabulky] ADD COLUMN geomtmp geometry([datový typ],4326);

UPDATE [název tabulky] SET geomtmp = st_setsrid(st_point(lambda,fi),4326);

ALTER TABLE [název tabulky] DROP COLUMN [souřadnice lambda];

ALTER TABLE [název tabulky] DROP COLUMN [souřadnice fi];

SELECT AddGeometryColumn([název tabulky], 'geom', 5514, [datový typ], 2);

UPDATE [název tabulky] SET geom = ST_Transform(geomtmp, 5514);

SELECT DropGeometryColumn([název tabulky], 'geomtmp');

ALTER TABLE [název tabulky] ADD COLUMN id SERIAL;

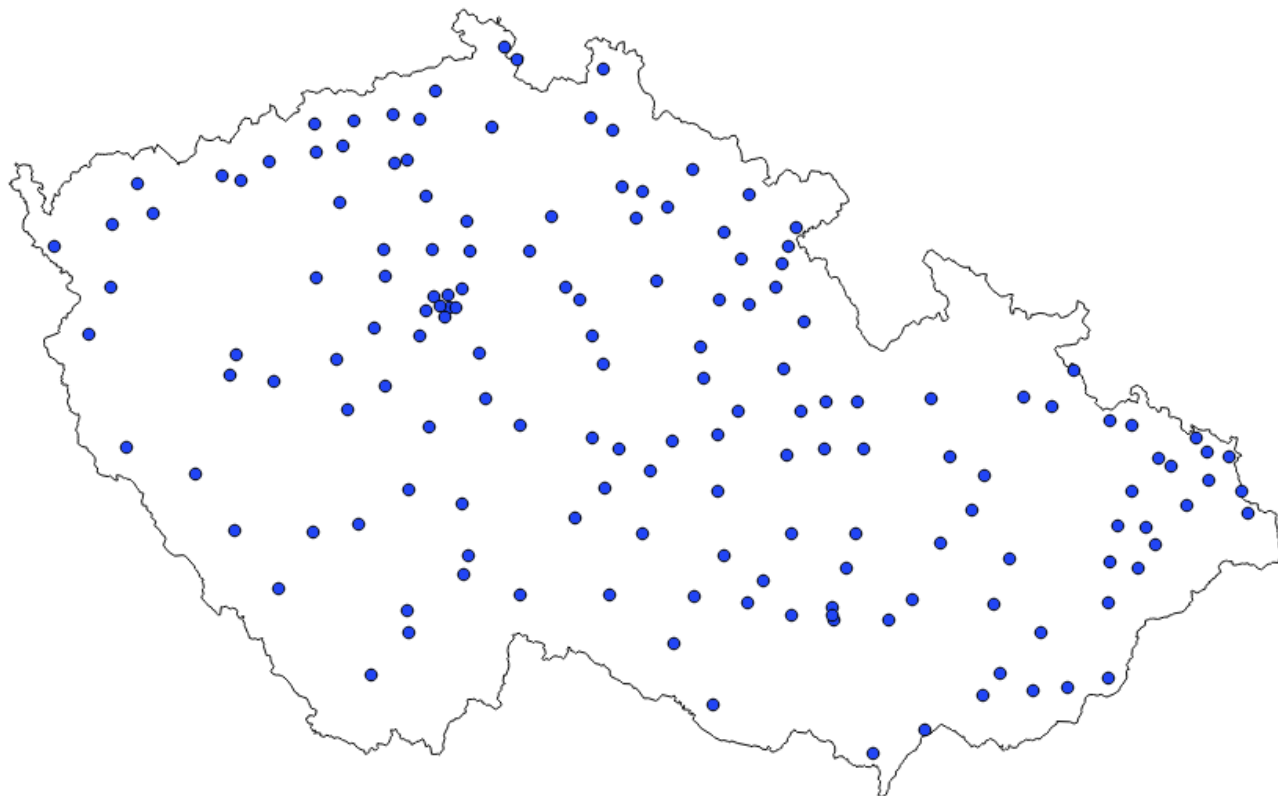
ALTER TABLE [název tabulky] ADD PRIMARY KEY (id);

ALTER TABLE [název tabulky] ALTER COLUMN geom SET NOT NULL;

CREATE INDEX lednihokej_geom_idx ON [název tabulky] USING gist(geom);

SELECT id, ST_IsValidReason(geom) AS reason FROM [název tabulky] WHERE ST_IsValid(geom) = FALSE;
```

lednihokej						
sloupec	datový typ	not null	unique	primary key	index	default
geom	geometry(Point,5514)	x			x	
id	int	x	x	x	x	nextval
nazevklubu	text					
mesto	text					
kryty	int					



5 Dotazy

Dotazy lze rozdělit na atributové a prostorové a jejich řešení je možno provést třemi způsoby.

Dotazy			
řešení	atributové	prostorové	sum
dotaz/poddotaz	2	2	4
CTE	0	6	6
dočasná tabulka	0	0	0
sum	2	8	-

5.1 Atributové

Kolik je v databázi hokejových stadionů, které nejsou zastřešeny a kolik je těch zastřešených?

```
SELECT kryty, count(*) FROM lednihokej GROUP BY kryty;
```

13 nezastřešených, 146 zastřešených

Kolik je v databázi hokejových klubů, které nemají v názvu jméno města?

```
SELECT count(*) FROM lednihokej WHERE NOT nazevklubu ~ mesto;
```

22 klubů

5.2 Prostorové

Vyberte všechny hokejové kluby v okrese Chrudim.

```
SELECT a.nazevklubu AS klub FROM lednihokej AS a JOIN (SELECT * FROM okresy WHERE okres = 'Chrudim') AS b
ON st_intersects(a.geom, b.geom) ORDER BY klub ASC;
```

HC Chrudim, HC Hlinsko, HC Skuteč

Vyberte okres(y) s nejvyšším počtem hokejových stadionů.

```
WITH cte AS (  
  
SELECT a.okres,count(*) AS count FROM okresy AS a JOIN lednihokej AS b ON st_intersects(a.geom, b.geom)  
  
GROUP BY a.okres ORDER BY count DESC  
  
) SELECT okres,count FROM cte WHERE count = (SELECT max(count) FROM cte);
```

Hlavní město Praha (8 klubů)

Vyberte hokejový klub, který je vzdušnou čarou nejvzdálenější od klubu HC Skuteč.

```
WITH cte AS (  
  
SELECT nazevklubu,st_distance(geom,(SELECT geom FROM lednihokej WHERE nazevklubu = 'HC Skuteč')) AS distance  
FROM lednihokej  
  
) SELECT nazevklubu,round(cte.distance/1e3) AS distance_km FROM cte WHERE distance = (SELECT max(distance)  
FROM cte);
```

HC Stadion Cheb (216 km)

Vyberte okresy, ve kterých více fotbalových stadionů než hokejových (pokud je počet NULL tak neporovnávejte)?

```
WITH cte_hokej AS (  
  
SELECT a.okres,count(*) AS count FROM okresy AS a JOIN lednihokej AS b ON st_intersects(a.geom,  
b.geom)  
  
GROUP BY a.okres  
  
) , cte_fotbal AS (  
  
SELECT a.okres,count(*) AS count FROM okresy AS a JOIN fotbal AS b ON st_intersects(a.geom, b.geom)  
  
GROUP BY a.okres  
  
)  
  
SELECT okresy.okres /* ,cte_hokej.count,cte_fotbal.count */ FROM okresy  
  
JOIN cte_hokej ON okresy.okres = cte_hokej.okres  
  
JOIN cte_fotbal ON okresy.okres = cte_fotbal.okres  
  
WHERE cte_fotbal.count > cte_hokej.count;
```

Frýdek-Místek, Kladno, Praha-východ, Jindřichův Hradec, Kutná Hora, Český Krumlov

(hodnoty null by se musely nahradit nulami a použít FULL JOIN)

Ve kterých okresech není žádný hokejový stadion?

```
SELECT okres FROM okresy WHERE okres NOT IN(  
  
SELECT DISTINCT a.okres FROM okresy AS a JOIN lednihokej AS b ON st_intersects(a.geom, b.geom) ) ORDER BY  
okres ASC;
```

Jeseník, Plzeň-jih

Ve kterém okrese je nejvíce sportovišť (hokej + fotbal + tenis) (pokud je počet NULL tak neporovnávejte)?

```
WITH cte_hokej AS (  
  

```

```

SELECT a.okres,count(*) AS count FROM okresy AS a JOIN lednihokej AS b ON st_intersects(a.geom,
b.geom)

GROUP BY a.okres

), cte_fotbal AS (

SELECT a.okres,count(*) AS count FROM okresy AS a JOIN fotbal AS b ON st_intersects(a.geom, b.geom)

GROUP BY a.okres

) , cte_tenis AS (

SELECT a.okres,count(*) AS count FROM okresy AS a JOIN tenis AS b ON st_intersects(a.geom, b.geom)

GROUP BY a.okres

), cte_sum AS (

SELECT okresy.okres,cte_hokej.count+cte_fotbal.count+cte_tenis.count AS sum FROM okresy

JOIN cte_hokej ON okresy.okres = cte_hokej.okres

JOIN cte_fotbal ON okresy.okres = cte_fotbal.okres

JOIN cte_tenis ON okresy.okres = cte_tenis.okres

ORDER BY sum DESC

) SELECT okres FROM cte_sum WHERE sum = (SELECT max(sum) FROM cte_sum);

```

Hlavní město Praha

(hodnoty null by se musely nahradit nulami a použít FULL JOIN)

Který zimní stadion je nejbližší hranic?

```

WITH cte AS (

SELECT nazevklubu,mesto,st_distance(a.geom,b.geom) AS dist FROM lednihokej AS a,

(SELECT st_exteriorring((st_dump(geom)).geom) AS geom FROM staty) AS b

)

SELECT mesto FROM cte WHERE dist = (SELECT min(dist) FROM cte);

```

Český Těšín

Které obce mají více než jeden zimní stadion?

```

WITH cte AS (

SELECT a.obec,count(*) AS count FROM obce AS a JOIN lednihokej AS b ON st_intersects(a.geom, b.geom) GROUP BY
a.obec

) SELECT obec FROM cte WHERE count >1 ORDER BY obec ASC;

```

Praha, Brno, Ostrava

6 pgRouting

Pro síťové analýzy bylo vytvořeno několik funkcí. Ty byly poté použity.

6.1 Funkce

clipNetwork

Před prováděním síťových analýz je dobré zmenšit oblast výběru, kvůli rychlosti výpočtu. Pro to je třeba ořezat síť podle nějakého polygonu. K tomu slouží funkce *clipNetwork*.

```
CREATE OR REPLACE FUNCTION clipNetwork(networkTableName TEXT,clippingArea TEXT,clippedNetworkTableName TEXT)
RETURNS text AS $$

DECLARE

    count INTEGER;

BEGIN

    EXECUTE 'DROP TABLE IF EXISTS ' || clippedNetworkTableName || ' ';

    EXECUTE 'CREATE TABLE ' || clippedNetworkTableName || ' AS SELECT st_intersection(a.geom,
b.geom) AS geom FROM ' || networkTableName || ' AS a JOIN ' || clippingArea || ' AS b ON a.geom && b.geom';

    EXECUTE 'SELECT count(*) FROM ' || clippedNetworkTableName || ' INTO count;

    IF count = 0 THEN RETURN 'Table not contains any rows!';

    ELSE RETURN 'Done!';

    END IF;

END;

$$ LANGUAGE plpgsql;
```

multiFeatures2SingleFeatures

Síť je poté nutné převést z multiprvků na singleprvky. K tomu byla napsána další funkce.

```
CREATE OR REPLACE FUNCTION multiFeatures2SingleFeatures(selectedGeometry TEXT,tableName TEXT, tol FLOAT)
RETURNS text AS $$

DECLARE

    count INTEGER;

BEGIN

    EXECUTE 'DROP TABLE IF EXISTS ' || tableName || ' ';

    EXECUTE 'CREATE TABLE ' || tableName || ' AS SELECT (st_dump(geom)).geom AS geom FROM ' ||
selectedGeometry || ' AS foo';

    EXECUTE 'ALTER TABLE ' || tableName || ' ADD COLUMN id serial;';

    EXECUTE 'ALTER TABLE ' || tableName || ' ADD PRIMARY KEY (id);';

    EXECUTE 'CREATE INDEX ON ' || tableName || ' USING gist (geom);';

    EXECUTE 'ALTER TABLE ' || tableName || ' ADD COLUMN source integer;';

    EXECUTE 'ALTER TABLE ' || tableName || ' ADD COLUMN target integer;';

    EXECUTE 'SELECT pgr_createTopology('' ' || tableName || ', ' || tol || ', ''geom'', ''id'');';

    EXECUTE 'CREATE INDEX ON ' || tableName || ' (source);';

    EXECUTE 'CREATE INDEX ON ' || tableName || ' (target);';

    EXECUTE 'ALTER TABLE ' || tableName || ' ADD COLUMN cost double precision;';

    EXECUTE 'UPDATE ' || tableName || ' SET cost = st_length(geom);';

    EXECUTE 'SELECT count(*) FROM ' || tableName || ' INTO count;
```

```

        IF count = 0 THEN RETURN 'Table not contains any rows!';

        ELSE RETURN 'Done!';

    END IF;

END;

$$ LANGUAGE plpgsql;

```

getNearestVerticeld

Dále je třeba zjistit identifikátor nejbližšího bodu topologicky správné sítě.

```

CREATE OR REPLACE FUNCTION getNearestVerticeId(verticesTableName TEXT, pointGeometry TEXT) RETURNS integer AS
$$
    DECLARE

        result INTEGER;

    BEGIN

        EXECUTE 'SELECT id FROM ' || verticesTableName || ' AS a,' || pointGeometry ||

            ' AS b order by st_distance(a.the_geom,b.geom) limit 1' INTO result;

        RETURN result;

    END;

$$ LANGUAGE plpgsql;

```

createShortestPathGeometry

A na závěr je tu funkce, která vytvoří geometrii nejkratší cesty na základě Dijkstrova algoritmu.

```

CREATE OR REPLACE FUNCTION createShortestPathGeometry(vertexA INTEGER,vertexB INTEGER, networkTableName
TEXT,tableName TEXT

) RETURNS text AS $$

    DECLARE

        count INTEGER;

    BEGIN

        EXECUTE 'DROP TABLE IF EXISTS ' || tableName || ' ';

        EXECUTE 'CREATE TABLE ' || tableName || ' AS

            SELECT seq, id1 AS node, id2 AS edge, a.cost, b.geom AS geom FROM pgr_dijkstra(

                ''SELECT id, source, target, cost FROM ' || networkTableName || '',

                ' || vertexA || ', ' || vertexB || ', false, false) AS a LEFT JOIN ' ||

networkTableName || ' AS b ON (a.id2 = b.id)';

        EXECUTE 'SELECT count(*) FROM ' || tableName || ' INTO count;

        IF count = 0 THEN RETURN 'Table not contains any rows!';

        ELSE RETURN 'Done!';

    END IF;

END;

$$ LANGUAGE plpgsql;

```


6.2 Síťové analýzy

PgRouting byl proveden na jednom příkladu:

Najděte nejkratší cestu ze zimního stadionu v Pardubicích na zimní stadion v Moravské Třebové.

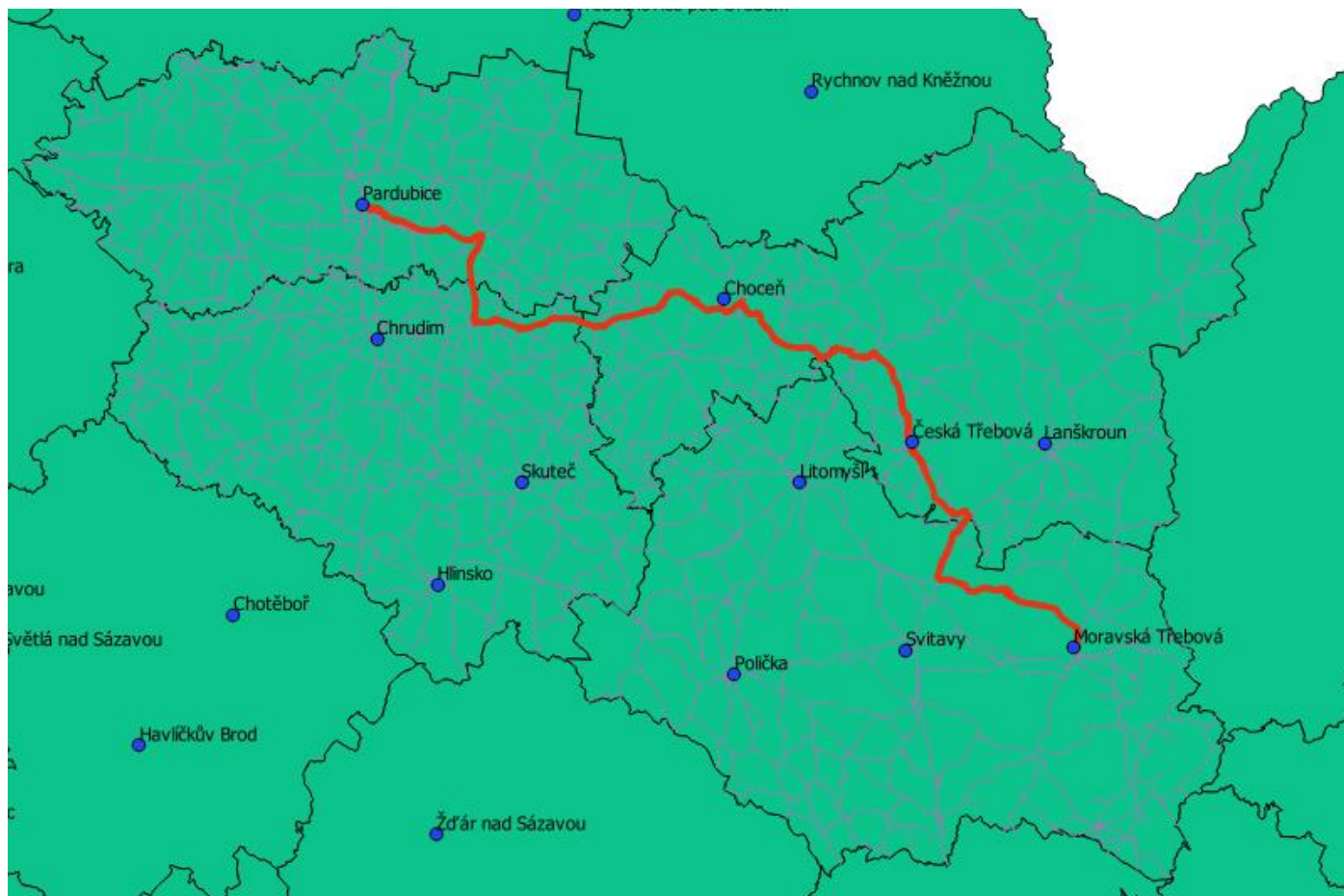
```
SELECT clipNetwork('silnice','(SELECT geom FROM okresy WHERE okres IN(''Chrudim'',''Pardubice'',''Ústí nad Orlicí'',''Svitavy''))', 'silnice_pardubickykraj');
```

```
SELECT multiFeatures2SingleFeatures('(SELECT geom FROM silnice_pardubickykraj WHERE NOT st_isempty(geom))','silnice_pardubickykraj_s',10.0);
```

```
SELECT createShortestPathGeometry(
```

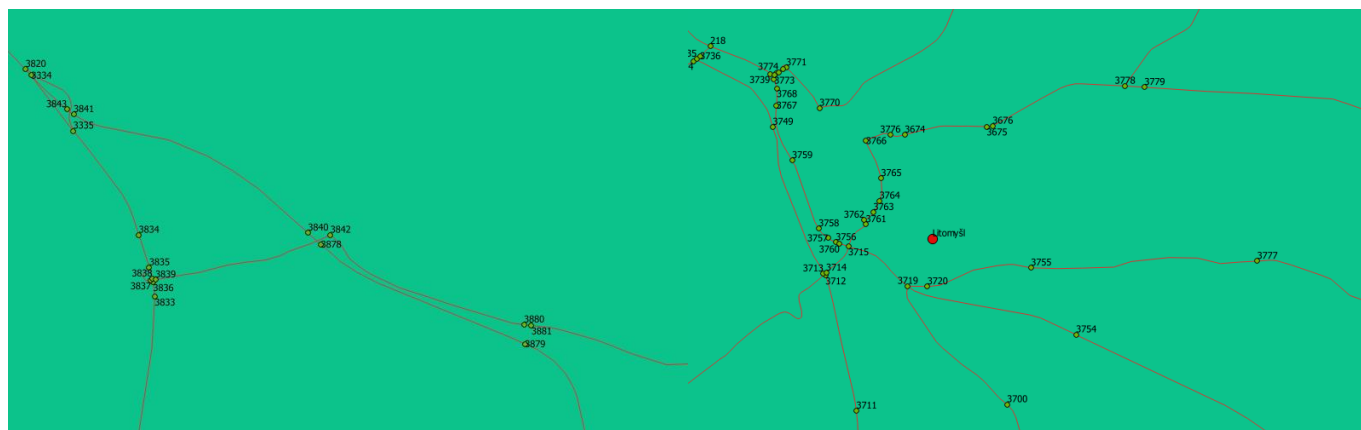
```
(SELECT getNearestVertexId('silnice_pardubickykraj_s_vertices_pgr','(SELECT geom FROM lednihokej WHERE mesto = ''Pardubice'')'),
```

```
(SELECT getNearestVertexId('silnice_pardubickykraj_s_vertices_pgr','(SELECT geom FROM lednihokej WHERE mesto = ''Moravská Třebová'')')), 'silnice_pardubickykraj_s', 'silnice_neohodnocena');
```



7 Závěr

Byla vytvořena databáze obsahující prostorová data. Ta byla získána ze třech různých zdrojů: z OSM, z RÚIAN a za pomoci ručního sběru na maps.google.com. Databáze obsahuje 7 základních vrstev (ostatní jsou výsledkem síťové analýzy). Dále bylo nad těmito daty formulováno 10 dotazů (2 atributové + 8 prostorových). Jelikož tabulky obsahují malý počet sloupců s atributy, nebylo snadné vůbec vymyslet, alespoň trochu obtížnější atributový dotaz. Dotazy byly řešeny pomocí dotazů/poddotazů či CTE. Pro provedení síťových analýz byly sestaveny funkce. Provedena byla pouze jedna síťová analýza a to pomocí Dijkstrova algoritmu, kde hrany sítě byly ohodnoceny jejich délkou. Pokoušeli jsme se provést výpočet s jiným ohodnocením, ale dosáhli jsme stejného výsledku. Důvodem je pravděpodobně špatná kvalita dat ve vrstvě silnice. Na následujících obrázcích je vidět situace, kde jsou nadjezdy a podjezdy a dále také situace ve městě Litomyšl.



Z toho je vidět, že například v městské zástavbě by bylo dobré v některých místech doplnit některé silnice, ale zase ne příliš mnoho, aby to příliš nezpomalilo výpočet.