PostGIS v praxi

Michal Páleník

www.palenik.sk/open-data

About me

PostgreSQL user

my sites

- www.oma.sk
- www.freemap.sk - mainly routing
- www.palenik.sk

ako blafovať o GIS

coordinate system

projection

import

SQL commands

lessons learned

other than postGIS

Coordinate system

WGS84,

S-JSTK

Europe is moving away from north america by 2,5 cm per year

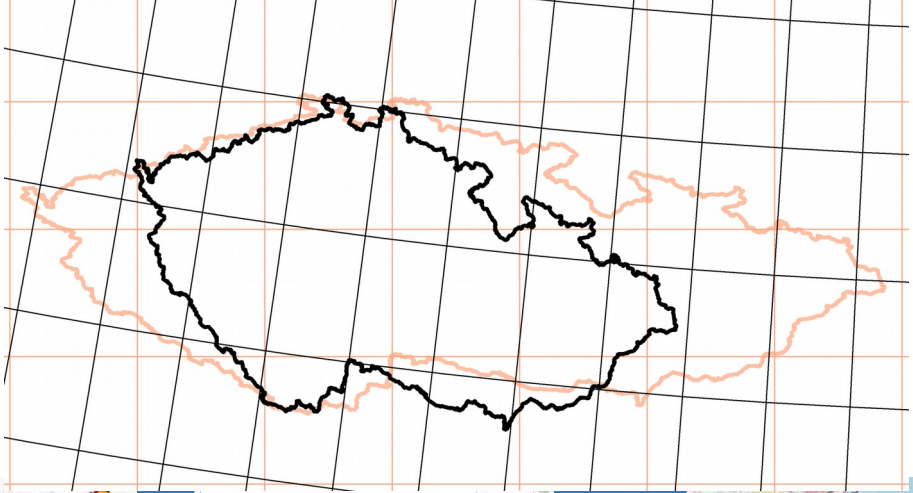
Projection

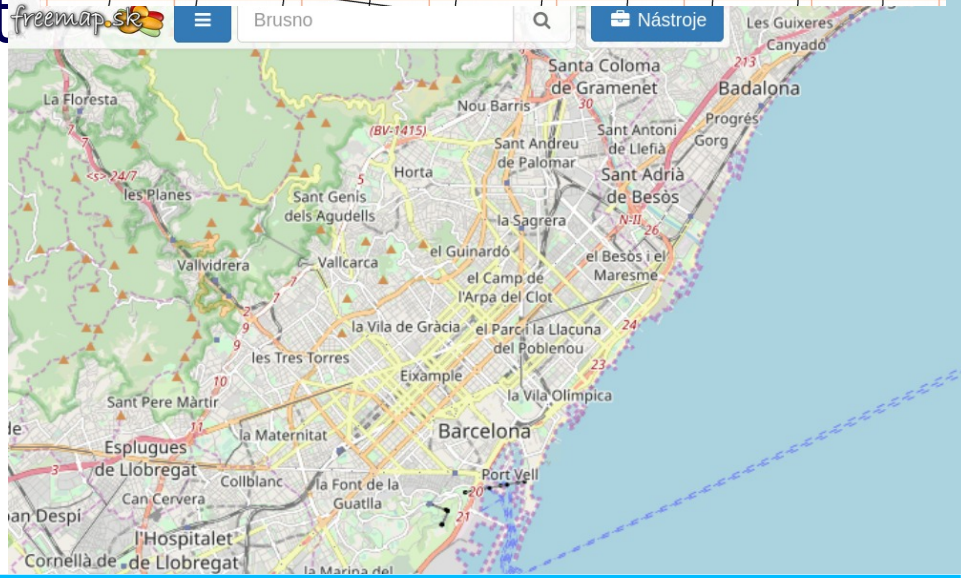
Krovak

Mercator

- northeast is northeast
- area is not the same

- more





Free Data

OpenStreetMap

ODbL licence

https://planet.osm.org/

48 GB PBF

Import

osm2pgsql, imposm, osmosis, osm2pgrouting

install PostGIS

apt install postgresql-10-postgis-2.4

„CREATE EXTENSION postgis;“

OSM structure

nodes, ways, relations

- + tags (name, created_by, ...)
- https://wiki.osm.org/wiki/Map_Features
- not fixed list of tags

nodes – lat,lon + tags

ways – list of ordered nodes + tags

relations – list of ordered nodes/ways/relations with roles + tags

import.style

node,way	amenity	text	polygon
node,way	boundary	text	polygon
node,way	admin_level	text	polygon
way	building	text	polygon
node,way	route	text	linear

import

wget -O /tmp/cr.pbf http://download.geofabrik.de/europe/czech-republic-latest.osm.pbf

osm2pgsql --create --slim --latlong --hstore --hstore-match-only --style import.style --database mapnik --prefix 'p2d2' --multi-geometry /tmp/cr.pbf

\d p2d2_point

```
                Table "public.p2d2_point"
  Column    |        Type         | Collation | Nullable | Default
------------+---------------------+-----------+----------+---------
 osm_id     | bigint              |           |          |
 amenity    | text                |           |          |
 boundary   | text                |           |          |
 admin_level | text               |           |          |
 route      | text                |           |          |
 tags       | hstore              |           |          |
 way        | geometry(Point,4326) |          |          |
Indexes:
    "p2d2_point_osm_id_idx" btree (osm_id)
    "p2d2_point_way_idx" gist (way)
```

list of pubs

select * from p2d2_point

where amenity='pub';

... where amenity in ('pub','bar','night_club');

select ... from p2d2_point ... union select ... from p2d2_polygon...;

... select st_centroid(way) as way from p2d2_polygon;

find Prague

```
select tags->'name', amenity, boundary, admin_level
from p2d2_polygon where tags->'name'='Praha';
 ?column? |  amenity   |    boundary    | admin_level
----------+------------+----------------+-------------
 Praha    |            | administrative | 4
 Praha    |            | administrative | 8
 Praha    | restaurant |                |
(3 rows)
```

find Prague

```
select osm_id, tags->'name', round(st_area(geography(way))) as area, amenity, boundary, admin_level from p2d2_polygon where tags->'name'='Praha';
```

```
  osm_id   | ?column? |   area    | amenity   |    boundary    | admin_level
-----------+----------+-----------+-----------+----------------+-------------
  -435541 | Praha    | 496267633 |           | administrative | 4
  -439840 | Praha    | 496267633 |           | administrative | 8
 229015602 | Praha    |       667 | restaurant |                |
```

https://www.openstreetmap.org/relation/435541

list of pubs in Prague

```
select osm_id, tags->'name' as name, st_distance(prague, way) from (select way as prague from p2d2_polygon where osm_id=-435541) as prague, p2d2_point where amenity='pub' order by st_distance(prague, way);
```

```
osm_id   |                name                |    st_distance
-----------+--------------------------------------------+--------------------
  895771091 | Satanka                                   |                 0
 1276242990 | Podkováň                                  |                 0
 2605824558 |                                           |                 0
 5692990062 | Výčep Fialka                              |                 0
 2258947412 | U Zdi                                     |                 0
 6211307175 | Vysočanská hospůdka                       |                 0
...
662841126 | U pěti šišek (Na Hřišti)                   |  0.248208919961008
 3742323860 | Zájezdní hostinec                         |  0.250725585008008
 3735982182 | Pivovarská restaurace Suchomasty          |  0.252897378397426
 3727647110 | Kovárna                                   |  0.254191911979901 ????
```

list of pubs in Prague

```
select osm_id, tags->'name' as name, st_distance(geography(prague), geography(way)) from (select way as prague from p2d2_polygon where osm_id=-435541) as prague, p2d2_point where amenity='pub' order by st_distance(prague, way);
```

```
 osm_id   |                 name                  |  st_distance
-----------+---------------------------------------+---------------
...
 662841126 | U pěti šišek (Na Hřišti)              |  18019
 3742323860 | Zájezdní hostinec                    |  27631
 3735982182 | Pivovarská restaurace Suchomasty     |  20599
 3727647110 | Kovárna                              |  28005
```

geometry vs. geography

geography: WGS84 and distances in meters

geometry: more functions, faster

- but strange distances

list of pubs in Prague

```
explain select osm_id, tags->'name' as name, st_distance(geography(prague), geography(way)) from (select way as prague from p2d2_polygon where osm_id=-435541) as prague,
p2d2_point where amenity='pub' order by st_distance(prague, way);
                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Sort  (cost=7122.22..7132.29 rows=4029 width=56)
   Sort Key: (st_distance(p2d2_polygon.way, p2d2_point.way))
   ->  Nested Loop  (cost=0.29..6880.95 rows=4029 width=56)
         ->  Index Scan using p2d2_polygon_osm_id_idx on p2d2_polygon  (cost=0.29..8.31 rows=1 width=2464)
               Index Cond: (osm_id = '-435541'::integer)
         ->  Seq Scan on p2d2_point  (cost=0.00..5543.07 rows=4029 width=242)
               Filter: (amenity = 'pub'::text)

explain select osm_id, tags->'name' as name from (select way as prague from p2d2_polygon where osm_id=-435541) as prague, p2d2_point where amenity='pub'
and st_contains(prague, way);
                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Nested Loop  (cost=5.44..446.36 rows=1 width=40)
   ->  Index Scan using p2d2_polygon_osm_id_idx on p2d2_polygon  (cost=0.29..8.31 rows=1 width=2464)
         Index Cond: (osm_id = '-435541'::integer)
   ->  Bitmap Heap Scan on p2d2_point  (cost=5.15..438.03 rows=1 width=242)
         Recheck Cond: (p2d2_polygon.way ~ way)
         Filter: ((amenity = 'pub'::text) AND _st_contains(p2d2_polygon.way, way))
         ->  Bitmap Index Scan on p2d2_point_way_idx  (cost=0.00..5.15 rows=116 width=0)
               Index Cond: (p2d2_polygon.way ~ way)
```

pub 354988465 „500“ is closed!!!

```
select osm_id, amenity, tags->'name', round(st_distance(mypub, geography(way))) as vzd from p2d2_point, (select geography(way) as mypub from p2d2_point where osm_id=354988465) as mypub where amenity is not null order by st_distance(mypub, geography(way)) limit 20;
```

osm_id	amenity	?column?	vzd
354988465	restaurant	Restaurant 500 (pětistovka)	0
354988470	recycling		34
619170327	toilets	Veřejné WC	75
3413954378	pharmacy	Hradčanská lékárenská společnost	81
4534896182	bicycle_parking		89
3413954377	bank	Česká spořitelna	92
4690862415	shelter		106
5713651736	restaurant	Veganland	111
3778159289	recycling		112
6631111325	vending_machine		114
309290158	cafe	Costa Coffee	115

500 is closed!!!

```
select osm_id, amenity, tags->'name', tags->'opening_hours' as opening_hours, round(st_distance(mypub, geography(way))) as vzd, degrees(st_azimuth(mypub, geography(way))) as azimuth from p2d2_point, (select geography(way) as mypub from p2d2_point where osm_id=354988465) as mypub where amenity in ('pub','restaurant','bar') and st_dwithin(mypub, geography(way), 500) order by st_distance(mypub, geography(way)) limit 10;
```

osm_id	amenity	?column?	opening_hours	vzd	azimuth
354988465	restaurant	Restaurant 500 (pětistovka)		0	
5713651736	restaurant	Veganland	Mo-Sa 10:30-21:00	111	-65.2499571149107
331960839	restaurant	Mash Hana		157	60.6542133671608
309290156	restaurant	Na Hradčanské		174	-83.0821047661996
307338919	restaurant	Sokolovna	Mo,Tu 10:00-23:00; We-Fr 10:00-24:00; Sa,Su 11:00-23:00	186	-20.4087586432402
360039727	pub	Krkonošská hospůdka		206	18.02128027214
2657339378	restaurant	Yam yam		279	-39.6975318875035
3156994661	restaurant	à table!		296	-38.1917875435709
354984792	restaurant	Indian by Nature II		299	-43.5924578972126
360039737	restaurant	Na Rozhraní	Mo-Fr 11:00-23:30, Sa 11:30-23:30, Su 12:00-22:00	299	-5.8246485869541

pubs close to tram 22

https://www.openstreetmap.org/relation/8060768

```
select tags->'name' as name,
round(st_distance(geography(way), tram)) as dist

from p2d2_point,

(select geography(way) as tram from p2d2_line
where osm_id=-8060768) as tram

where st_dwithin(geography(way), tram, 200) and
geography(way) && tram and amenity='pub';
```

pubs in Prague by segregation

```
select pubs1.osm_id, first(pubs1.tags->'name') as name, count(*) as
pocet, array_agg(pubs2.osm_id) as other

from (select * from (select way as prague from p2d2_polygon
where osm_id=-435541) as prague, p2d2_point where
amenity='pub' and st_contains(prague, way)) as pubs1

 left outer join (select * from (select way as prague from
p2d2_polygon where osm_id=-435541) as prague, p2d2_point
where amenity='pub' and st_contains(prague, way)) as pubs2

on pubs1.osm_id != pubs2.osm_id and
st_dwithin(geography(pubs1.way), geography(pubs2.way), 1000)
group by pubs1.osm_id order by count(*) desc;
```

least segregated:

4248794095	Klub Buben	40
4248794094	Sport Pub	39
2408146791	Hostinec Slavíkova	39
355008175	Krčma U Parašutistů	39
4184471426	Rocky O' Reillys Irish Pub	38
296601430	Belzepub	38
1832498512	U Kacíře	38
331960300	U Žaludů	38
2420223836	Na Schůdku	38
331960299	Milá tchýně	38
355008194	Vínko Charlie	37
3457334001	Trafika BAR	37
4197006861	Branická Formanka	37
4326615089	Vape House Prague	36
552405058	Hostinec u Rotundy	36

most segregated

309103062 | U žraloka | 1 | {309103011}
2614956474 | Palpost | 1 | {NULL}
6357245380 | Hospoda U lesa | 1 | {NULL}
309103011 | Bar & Vinárna | 1 | {309103062}
7042015085 | Restaurace Pivovar Řeporyje | 1 | {2167490020}
1143014989 | Na Korunce | 1 | {NULL}
1109692879 | Jako doma | 1 | {NULL}
4438779796 | Hurdálkova chýše jedová | 1 | {1712556993}
4323521297 | Spartak | 1 | {NULL}
1008212347 | | 1 | {NULL}
4257560092 | Hospoda Konrad | 1 | {NULL}
296804168 | | 1 | {NULL}
1403144336 | U Kubíčků | 1 | {1736152126}
4515217790 | U Huberta | 1 | {NULL}
1409474794 | Na Hřišti | 1 | {NULL}
4586922093 | Hospůdka U Bobra | 1 | {NULL}
4587023791 | Sportovní hospoda | 1 | {589970268}
1427338969 | | 1 | **{NULL} ?? is this it?**
2709963362 | U Herčíků | 1 | {3237974418

most segregated

```
select pubs1.osm_id, first(pubs1.tags->'name') as name,
round(min(st_distance(geography(pubs1.way),
geography(pubs2.way)))) as min_dist
from (select * from (select way as prague from p2d2_polygon
where osm_id=-435541) as prague, p2d2_point where
amenity='pub' and st_contains(prague, way)) as pubs1
 left outer join (select * from (select way as prague from
p2d2_polygon where osm_id=-435541) as prague, p2d2_point
where amenity='pub' and st_contains(prague, way)) as pubs2
on pubs1.osm_id != pubs2.osm_id group by pubs1.osm_id order by
min_dist desc;
```

most segregated

osm_id	name	min_dist
5702007631	U Beránků	2689
4323521297	Spartak	2030
1143014989	Na Korunce	1905
296155831		1799
2378848820	Útulna	1542
1008212347		1504
5181640829	Комо	1452
2858014134	U Čížků	1416
2614956474	Palpost	1356

display maps

TMS with OpenLayers or Leaflet or ...

- images (jpeg,png) in special directory structure
  - zoom/x/y.jpeg
- created by mapnik, ...

z/x/y to envelope

```
CREATE OR REPLACE FUNCTION www.bbox(zoom integer, x integer, y integer)  RETURNS geometry
 LANGUAGE plpgsql IMMUTABLE PARALLEL SAFE
AS $function$
DECLARE
 n float; E float = 2.7182818284; tt float; lat float; lon float;
BEGIN
  n = 1.0*power(2.0, zoom);
  tt=pi() - (2.0 * pi() * y) / n;
  lat = degrees(atan( (1 - power(E, -2*tt)) / (2 * power(E, -tt)) ));
  lon = 360.0 * x / n - 180; --ok
  tt=pi() - (2.0 * pi() * (1+y)) / n;
  return ST_MakeEnvelope( lon, lat, 360.0 * (1+x) / n - 180, degrees(atan( (1 - power(E, -2*tt)) / (2 * power(E, -tt)) )),
4326);
END;
$function$
```

MVT in $tile=9,285,177

```
select st_asmvt(q, 'pubs', 4096, 'geom') as mvt
from (select
  – st_AsMvtGeom(way, BBox($tile), 4096, 256, true) as geom,
  – tags->‘name‘ as name, amenity
from p2d2_point where way && Bbox($tile)
) as q;
```

php code tile server

```
header("Access-Control-Allow-Origin: *");
header('Content-Type: application/x-protobuf');
header("Content-Disposition: attachment");
$res = pg_query($q); $out = '';
while($r = pg_fetch_assoc($res)) $out .= pg_unescape_bytea($r['mvt']);
echo $out;
```

html code browser

<script src='map.js'></script>

plus mapbox-gl.css and mapbox-gl.js

map.js

```
var simple = {"version": 8, "sources": {
      "osm": { "type": "vector", "attribution":"Zrobil <a href='https://www.palenik.sk/'>Michal Páleník</a> vďaka <a href='https://www.openstreetmap.org'>prispievateľom OpenStreetMap</a>", "minzoom": 5, "maxzoom": 17,
     "tiles": ["https://tiles.epsilon.sk/zaklad/{z}/{x}/{y}.pbf"] }, },
   "glyphs": "https://tiles.epsilon.sk/fonts/{fontstack}/{range}.pbf",
   "layers": [ { "id":"pubs", "type":"circle", "source":"osm", "source-layer": "pubs",
  "paint": { "circle-color": [ "case", ["==", ["get","amenity"], "pub"], "blue", ["==", ["get","amenity"], "restaurant"], "green", "yellow"], "fill-opacity": 0.7, "circle-radius": 4 },
} ]};
var map=new mapboxgl.Map({ container: "map",  style: simple, zoom: 9, center: [21.24, 48.84], minZoom: 5 });
map.addControl(new mapboxgl.NavigationControl()); map.dragRotate.disable();
```

http://project-osrm.org/

real distances

PostGIS used to create better routes

- prefer routes in forests

no function ST_RealDistance(point1,point2,mode)

- call osrm library, return foot/bicycle distance/linestring between points
- without http calls
- (easy to do with http json calls)

lessons learned

SSD, shared_buffers, ...

MVCC:

- one ten-page insert is far faster than 5 updates

indexes

- select .. where st_distance(a,b)<400; --slow
- select .. where st_dwithin(a,b,400);

create tables

create table tmp_schema.my_table as select ...

- 12 hours

check if number of rows is > 10

- or other basic quality control

begin; drop table live.my_table;

alter table tmp_schema.my_table set schema live;

commit;

Some small details

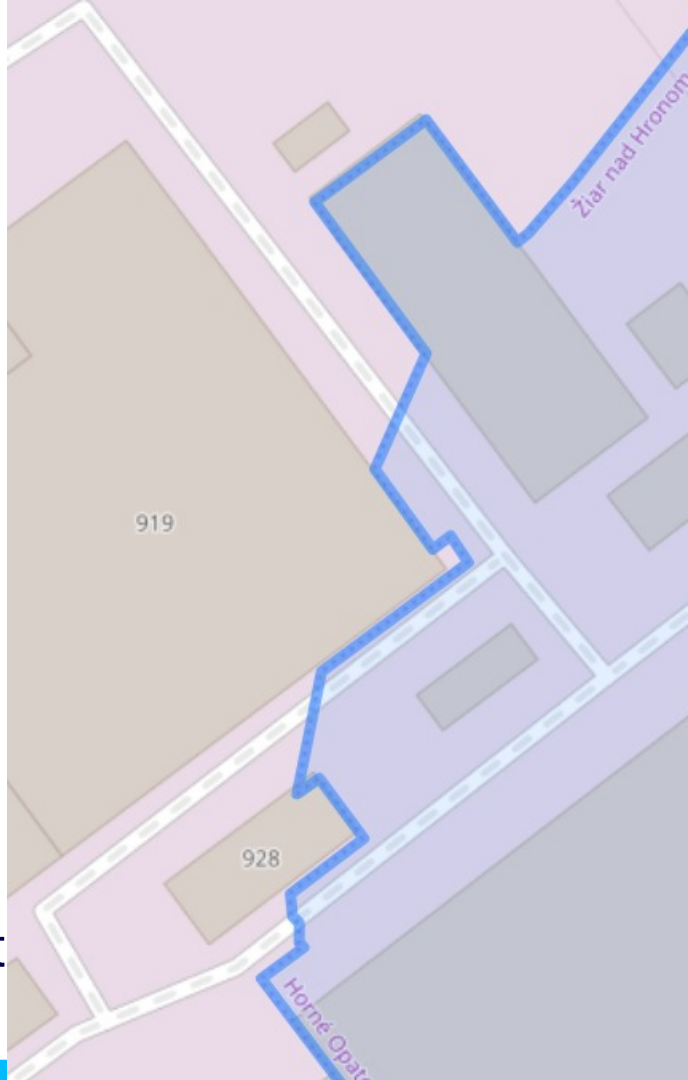
some borders are strange:

combination of

- ST_Buffer
- ST_SimplifyPreserveTopology

I would need:

- st_??(g1,g2,dst1,dst2)
- return true if they are closer than dst1
- return false if more than dst2
- return anything **fast** if between dst1 and dst



&&

... where g1 && g2

- return true if bounding boxes of g1 and g2 overlap
- very, very fast

some very big routes exist

turistika.oma.sk/e8

thank you

Michal Páleník

https://www.palenik.sk/open-data