

PostGIS

Jan Ježek

Západočeská Univerzita v Plzni

jezekjan@kma.zcu.cz

Podpořeno z projektu FRVŠ 584/2011.

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady
- 4 Příklady
- 5 PostGIS raster

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady
- 4 Příklady
- 5 PostGIS raster

PostGIS

- PostGIS - rozšíření databáze PostgreSQL o prostorové datové typy
- (PostgreSQL má prostor. dat. typ, ale neposkytují dostatek funkcionality pro GIS)
- Rozšíření databáze o prostorové funkce
- Rozšíření o prostorový index

PostGIS

- PostGIS - rozšíření databáze PostgreSQL o prostorové datové typy
- (PostgreSQL má prostor. dat. typ, ale neposkytují dostatek funkcionality pro GIS)
- Rozšíření databáze o prostorové funkce
- Rozšíření o prostorový index

`http://www.google.cz/trends`

Používané knihovny

- Proj <http://trac.osgeo.org/proj/>
- Geos <http://trac.osgeo.org/geos/>

Instalace

- Instalace pomocí SQL scriptů do databáze
- Instalace pomocí šablon

Databáze s PostGIS

- OGC - Spatial features for SQL
 - `geometry_columns`
 - `spatial_ref_sys`
 - `raster_columns`, `raster_overviews`

Databáze s PostGIS

- OGC - Spatial features for SQL
- geometry_columns
- spatial_ref_sys
- raster_columns, raster_overviews

Databáze s PostGIS

- OGC - Spatial features for SQL
- `geometry_columns`
- `spatial_ref_sys`
- `raster_columns`, `raster_overviews`

Vytvoření prostorového prvku

```
select geomfromtext('POINT(0 0 0)',4326)
```

Specifikace OGC WKT

POINT(0 0)

LINESTRING(0 0,1 1,1 2)

POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))

MULTIPOINT(0 0,1 2)

MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))

MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)),

((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))

GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))

PostGIS EWKT

POINT(0 0 0) -- XYZ

SRID=32632;POINT(0 0) -- XY with SRID

POINTM(0 0 0) -- XYM

POINT(0 0 0 0) -- XYZM

SRID=4326;MULTIPOINTM(0 0 0,1 2 1) -- XYM with SRID

MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))

POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,

MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),

(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,

GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4

MULTICURVE((0 0, 5 5), CIRCULARSTRING(4 0, 4 4, 8 4))

POLYHEDRALSURFACE(((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)),

((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),

((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)), ((1 1 0, 1 1 1, 1 0

((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0)), ((0 0 1, 1 0 1, 1 1

Ruční vytváření geometrie

```
create table points (pt geometry, name varchar);  
insert into points values ('POINT(0 0)', 'Origin');  
insert into points values ('POINT(5 0)', 'X Axis');  
insert into points values ('POINT(0 5)', 'Y Axis');
```

```
select name, ST_AsText(pt),  
ST_Distance(pt, 'POINT(5 5)') from points;
```

SQL-MM - křivky

```
CURVEPOLYGON(CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0),(1 1,
```


Datové typy - geometry a geography

```
SELECT ST_Distance(  
geomfromtext('POINT(0 0 0)',4326)::geography,  
geomfromtext('POINT(0 1 0)',4326)::geography)
```

|SPATIAL_REF_SYS| a souřadnicové systémy

- Referenční souřadnicový systém
- Geodetické datum
- Kartografické zobrazení
- EPSG kódy

|SPATIAL_REF_SYS| a souřadnicové systémy

- Referenční souřadnicový systém
- Geodetické datum
- Kartografické zobrazení
- EPSG kódy

|SPATIAL_REF_SYS| a souřadnicové systémy

- Referenční souřadnicový systém
- Geodetické datum
- Kartografické zobrazení
- EPSG kódy

|SPATIAL_REF_SYS| a souřadnicové systémy

- Referenční souřadnicový systém
- Geodetické datum
- Kartografické zobrazení
- EPSG kódy

Tabulka GEOMETRY_COLUMNS

- Tabulka pro ukládání metadata
- Vytvoření sloupce geom pomocí funkce

```
SELECT AddGeometryColumn('public',  
    'roads_geom', 'geom',  
    423, 'LINESTRING', 2)
```

- Vytvoření a vložení manuálně.

Tabulka GEOMETRY_COLUMNS

- Tabulka pro ukládání metadata
- Vytvoření sloupce geom pomocí funkce

```
SELECT AddGeometryColumn('public',  
    'roads_geom', 'geom',  
    423, 'LINESTRING', 2)
```

- Vytvoření a vložení manuálně.

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady
- 4 Příklady
- 5 PostGIS raster

Nahrávání GIS Dat - SQL

```
BEGIN;  
INSERT INTO roads (road_id, roads_geom, road_name)  
VALUES (1,ST_GeomFromText('LINESTRING(191232 243118,191108  
INSERT INTO roads (road_id, roads_geom, road_name)  
VALUES (2,ST_GeomFromText('LINESTRING(189141 244158,189265  
INSERT INTO roads (road_id, roads_geom, road_name)  
VALUES (3,ST_GeomFromText('LINESTRING(192783 228138,192612
```

Nahrávání GIS Dat - pomocí utilit

```
shp2pgsql -c -D -s 4269 -i -I  
shaperoads.shp myschema.roadstable > roads.sql  
psql -d roadsdb -f roads.sql
```

Nahrávání GIS Dat - ogr2ogr

Získávání dat pgsql2shp

```
pgsql2shp [<options>] <database> [<schema>.]<table>  
pgsql2shp [<options>] <database> <query>
```

Plugin pro QGIS - SPIT

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady**
- 4 Příklady
- 5 PostGIS raster

Ukázka - ArcCR Všechny řeky, které procházejí vybraným okresem:

```
select  
vod_tok.gid, vod_tok.the_geom, vod_tok.nazev from vod_tok, okres  
where  
okresy.nazev ='Beroun' AND  
vod_tok.the_geom && okresy.the_geom AND  
st_intersects(vod_tok.the_geom, okresy.the_geom);
```

Ukázka - ArcCR Všechny řeky, které procházejí vybraným okresem:

```
select  
vod_tok.gid, vod_tok.the_geom, vod_tok.nazev from vod_tok, okres  
where  
okresy.nazev ='Beroun' AND  
vod_tok.the_geom && okresy.the_geom AND  
st_intersects(vod_tok.the_geom, okresy.the_geom);
```


Ukážká - ArcCR Všechny řeky, které procházejí vybraným okresem:

```
select vod_tok.gid, vod_tok.the_geom, vod_tok.nazev  
from vod_tok, okresy where  
okresy.nazev ='Beroun' AND  
vod_tok.the_geom && okresy.the_geom AND  
st_intersects(vod_tok.the_geom, okresy.the_geom);
```

Ukázka - ArcCR Všechny řeky, které procházejí vybraným okresem:

```
select vod_tok.gid, vod_tok.the_geom, vod_tok.nazev  
from vod_tok, okresy where  
okresy.nazev ='Beroun' AND  
vod_tok.the_geom && okresy.the_geom AND  
st_intersects(vod_tok.the_geom, okresy.the_geom);
```

Ukázka - ArcCR Všechny obce které leží od vybrané řeky v určité vzdálenosti

```
select obce_b.gid, obce_b.the_geom, obce_b.nazob FROM  
vod_tok, obce_b  
where  
vod_tok.nazev ='Berounka' AND  
distance(obce_b.the_geom, vod_tok.the_geom)<0.05 ;
```

Vytvoření funkce

```
CREATE FUNCTION I_DWithin(geometry, geometry, float8)
  RETURNS boolean
  AS '
    SELECT
      $1 && expand($2,$3) AND
      $2 && expand($1,$3) AND
      distance($1, $2) < $3
  ' LANGUAGE 'SQL' IMMUTABLE;
```

```
select obce_b.gid, obce_b.the_geom, obce_b.nazob FROM
vod_tok, obce_b
where
vod_tok.nazev ='Berounka' AND
I_Dwithin(obce_b.the_geom, vod_tok.the_geom, 0.05)
```

Indexování prostorových dat

```
CREATE INDEX [indexname] ON [tablename] USING GIST ( [geomcolumnname]  
VACUUM ANALYZE [table_name] [(column_name)];
```

GIST index:

- Obecná metoda pro různé algoritmy
- Použítou pouze pro

<<, &<, &>, >>,
<<| , &<| ,|&> ,|>>
@> , @ , ~= , &&

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady
- 4 Příklady**
- 5 PostGIS raster

Desktop

- uDig
- QGIS

Server

- Geoserver
- UMNMapserver

Výkonost

- Objemná data v jedné geometrii vs. a počet řádků v tabulce
"SET enable_seqscan TO on;"

Po dotazu.

Data Clustering

- Seskupení dat podle indexu
- cluster thegeomindex ON okresy

Indexace

- Partial index
- Index na více položek
- Index na pro funkci

Denormalizace ...

Výkonost - High Availability, Load Balancing a Replication

Obsah

- 1 PostGIS - úvod
- 2 Nahrávání a získávání GIS dat
- 3 Příklady
- 4 Příklady
- 5 PostGIS raster

Nahrávání

```
raster2pgsql.py -r  
/media/data_/Dokumenty/geodata/rasters/coiba.jpg  
-I -t coiba -o coiba.sql
```


Raster - od verze 2.0.0

- Nový datový typ raster
- Utility pro nahrávání a získávání rastru raster2pgsql

Nahrání do databáze

```
psql -h localhost -U jezek -d test -f coiba.sql
```

Nahrávání - dlaždice

```
raster2pgsql.py -r /media/data_/Dokumenty/geodata/rasters/  
-t pele -o pele.sql
```

Dotaz na data

```
SELECT count(*) As num_rasters, ST_Height(rast) As height,  
ST_Width(rast) As width, ST_SRID(rast) As srid,  
ST_NumBands(rast) As num_bands,  
ST_BandPixelType(rast,1) As btype  
FROM pele  
GROUP BY ST_Height(rast) ,  
ST_Width(rast), ST_SRID(rast),  
ST_NumBands(rast),  
ST_BandPixelType(rast,1);
```

Dotaz na data

```
SELECT astext(ST_Polygon(rast)) FROM pele limit 10
```