Geodatabáze

(Návrh a řízení prostorových databází)

Katedra kartografie a aplikované geoinformatiky, PřF UK

Předmět: „Návrh a řízení prostorových databází“

Cílem předmětu je seznámit se s principy geo-databázových systémů a prakticky se naučit zvládnout základní techniky práce s geografickými daty: manipulaci a analýzu geodat v prostředí relačně-objektového databázového systému PostgreSQL pomocí jazyka SQL. PostGIS je plnohodnotná Open Source alternativa k proprietárním geodatabázím typu Oracle Spatial.

Předpoklady: základy SQL (předmět Principy databází), základy GIS

Obsah

1. CO JE GEODATABÁZE 4

2. OGC STANDARDY PRO GEODATABÁZE 6

2.1. TŘÍDA GEOMETRY 6
2.2. DATOVÝ MODEL PRO JEDNODUCHÉ GEOPRVKY 7
2.3: METODY GEOMETRICKÝCH OBJEKTŮ 8
2.4. PROSTOROVÉ VZTAHY GEOPRVKŮ 9
2.5. METODY PROSTOROVÉ ANALÝZY GEOPRVKŮ 9
2.6. FORMA ZÁPISU PROSTOROVÝCH (GIS) DAT 9
2.3. OGC JEDNODUCHÉ GEOPRVKY PRO SQL 10

3. TOPOLOGIE 12

4. RASTRY 13

5. INDEXY 14

6. POSTGIS 19

6.1. VYTVÁŘENÍ DATABÁZE S PROSTOROVÝMI TABULKAMI 20
6.2. VSTUPNĚ VÝSTUPNÍ PROCEDURY 21
6.3. PROSTOROVÉ FUNKCE A DOTAZY 21
6.3.1. GEOMETRICKÉ KONSTRUKTORY (*GEOMETRY CONSTRUCTORS*) 21
6.3.2. PŘÍSTUP KE GEOMETRII (GEOMETRY ACCESSORS) 22
6.3.3. PROSTOROVÉ EDITORY (GEOMETRY EDITORS) 22
6.3.4. VÝSTUPY PROSTOROVÝCH DAT (GEOMETRY OUTPUTS) 22
6.3.5. PROSTOROVÉ OPERÁTORY (*OPERATORS*) 23
6.3.6. PROSTOROVÉ PREDIKÁTY (*SPATIAL RELATIONSHIPS*) 24
6.3.7. ROZMĚR GEOMETRIE (*MEASUREMENTS*) 24
6.3.8. GEOMETRICKÉ OPERACE (*GEOMETRY PROCESSING*) 24
6.3.9. LINEÁRNÍ REFERENCOVANÍ 24
6.3.10. OSTATNÍ NEZAŘAZENÍ FUNKCE 25
6.3.11. SFCGAL FUNKCE 25
6.4. PROSTOROVÝ INDEX 25
6.5. TOPOLOGIE 26
6.5.1. TOPOLOGICKA SPRAVA VEKTOROVYCH DAT 26
6.6. RASTER V POSTGIS 28
6.6.1. VSTUPNĚ VÝSTUPNÍ PROCEDURY PRO RASTRY 28
6.6.2. METADATA RASTROVÉ VRSTVY 28
6.6.3. KONVERZE 29
6.6.4. ZÁKLADNÍ FUNKCE 29
6.6.5. KONSTRUKTORY 29

1. Co je geodatabáze

Základní vymezení geodatabází.

Databáze:
-> uspořádaná množina informací (dat) na paměťovém médiu.

Ilustrace databáze na knihovním systému

Digitální skladování informací	Fyzické skladování informací
Databáze	Knihovna
Tabulky	Knihy
Obslužný program (sw)	Knihovnice (vyhledává informace)
Indexy (jak vyhledávat)	Rejstříky
Návrh DB	Organizace knihovny
Hardware (server): počet disků	Počet pater knihovny
Hardware (server): diskový prostor	Velikost volných regálů
Hardware (server): počet procesorů	Počet zaměstnanců
Řízení DB: indexace	Řazení podle abecedy / klíčových slov / jména autorů
Aktualizace indexů	Aktualizace katalogů / rejstříků
Komunikační jazyk: SQL v relační DB	Čeština / latina ...
Prostorová data	Mapy, atlasy a glóby

Geodatabáze:
-> je optimalizovaná DB pro
- skladování,
- údržbu (manipulace s prostorovými objekty),
- dotazování prostorových dat (popis polohy nebo tvarů).

Geodatabáze definuje:
1. Datové typy prostorových dat,
2. Metody pro dotazování a zpracování prostorových dat .

Prostorová data: entita definovaná v prostoru, u které je zřejmá identifikace, umístění v prostoru a vztah k okolním entitám.

- vektorová data (bod, linie, polygon)
  - geometrie (prostorová složka)
  - atributy (popisná složka)
  - topologická reprezentace dat (volitelně)
- rastrová data
- metadata (datat o datech)

Kdy ukládat do geodatabáze?

Výhody:

- úlohy vyžadující mnoho manuálních kroků;
- kontrola referenční integrity (“hlídání vstupů”);
- spolupráce více uživatelů;
- transakce (změna provedena celá nebo vůbec).

Nevýhody:

- cena implementace,
- “flexibilita”,
- kompatibilita s GIS sw,
- rychlost přístupu (centrální vs. lokální DB),
- datové struktury,
- uživatelské znalosti / požadavky.

2. OGC standardy pro geodatabáze

http://www.opengeospatial.org

Jednoduché geoprvky *(OGC Simple Features)* je standard OGC který definuje geometrické objekty v databázi (třída geometry), jejich prostorové vztahy, metody prostorové analýzy a formu zápisu prostorových (GIS) objektů.

2.1. Třída Geometry

Specifikace:

- abstraktní rodičovská třída,
- objekty mají prostorové a neprostorové vlastnosti,
- prostorové vlastnosti jsou reprezentovány 2D (3D) geometrickými objekty.

Geometrické objekty

- Point – bod
- Curve – posloupnost bodů, způsob interpolace
- LineString – linie, lomená čára (lineární interpolace)
- Line – linie s dvěma body
- LineRing – jednoduché a uzavřené linie
    - Jednoduchá křivka – sama sebe neproťıná
    - Uzavřená křivka – společný počáteční a koncový bod
- Polygon – definován vnějšími a vnitřními (tvoří tzv. “díry”) hranicemi (LineRing)
- GeometryCollection – multimnožina geometrických objektů různých typů (může obsahovat duplicitní prvky)
- MultiPoint – množina bodů
- MultiLineString – množina lomených čar
- MultiPolygon – množina polygonů

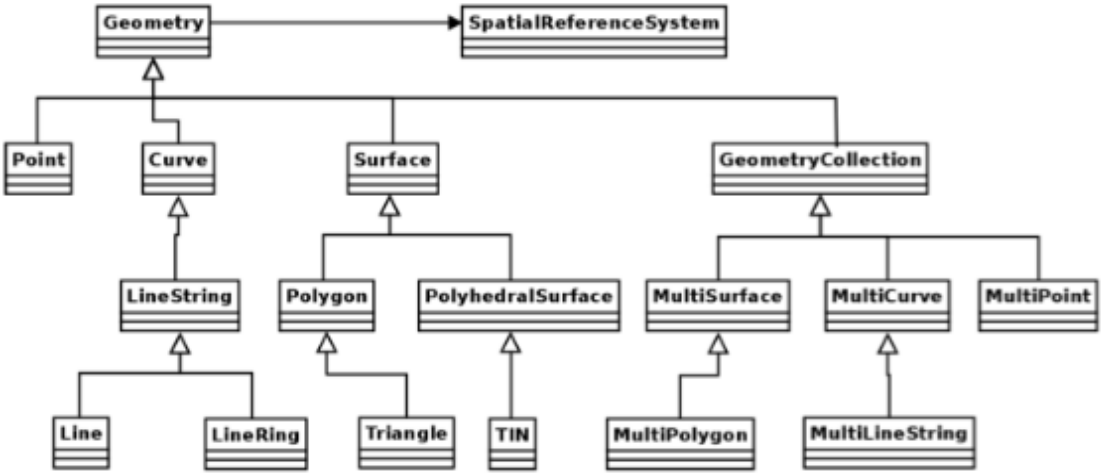
Dimenze objektů:

Geometrické objekty nulté (bod), první (linie) a druhé (polygon) dimenze v 2D/3D/4D jsou uloženy souřadnicovém systému.

- 2D (x, y)
- 3D(x,y,z) – výška
- 3D(x,y,m) – měření
- 4D(x,y,z,m)

2.2. Datový model pro jednoduché geoprvky
(*Simple Features*)

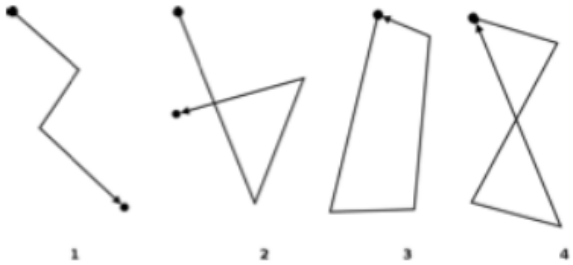
Relační schéma objektů jednoduchých geoprvků.



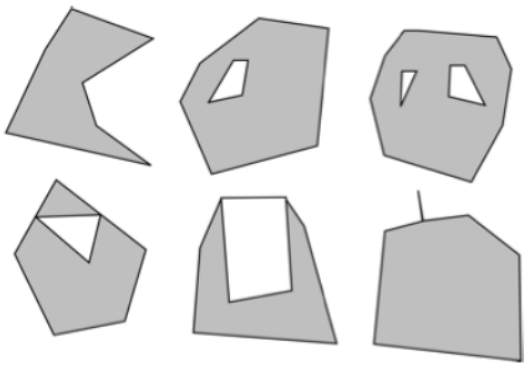
Obrázek X: Datový model jednoduchých geoprvků

<u>Definice jednoduchých geoprvků:</u>

- Bod (Point),
- Křivka (Curve): posloupnost bodu, zpusob interpolace,
  - LineString – linie, lomena cara (linearni interpolace)
  - Line – linie se dvema body
  - LineRing – jednoducha a uzavrena linie
    - Jednoducha krivka – sama sebe neprotına (1) (3)
    - Uzavrena krivka – spolecny pocatecnı a koncovy bod (4)



- Polygon: definovan vnejsimi a vnitrnimi hranicemi, hranice je objekt typu LineRing.



- Množina polygonů (PolyhedralSurface): sdílející společné hranice linie

TIN - Triangulated Irregular Network

Nekonzistentní množina polygonů je vyjádřena jako MultiSurface.

Kolekce geometrii (GeometryCollection): datový typ definující množinu geometrických objektu různých typů (muže obsahovat i duplicitní prvky).

Multi*
MultiPoint
MultiLineString (mnozina lomenych car)
MultiPolygon (mnozina polygonu)

2.3: Metody geometrických objektů

Vrací hodnoty typu geometrie, booleovské hodnoty nebo celé a desetinné číslo.

Point
X(),Y(),Z(),M()

Curve
Lenght(), StartPoint(), EndPoint(), IsClosed(), IsRing()

LineString
NumPoints(), PointN()

MultiCurve
IsClosed(), Length()

Surface/MultiSurface
Area(), Centroid(), PointOnSurface()

Polygon
ExteriorRing(), NumInteriorRing(), InteriorRingN()

PolyhedralSurface/TIN
NumPatches(), PatchN(), BoundingPolygons(), IsClosed()

GeometryCollection

NumGeometries(), GeometryN()

2.4. Prostorové vztahy geoprvků

Založeny na binárních prostorových predikátech (vztahy mezi dvěma prostorovými atributy). Součást modelu DE-9IM (Dimensionally Extended Nine-Intersection Model).

Equals()
Intersects()
Touches()
Crosses()
Within()
Contains()
Overlaps()

2.5. Metody prostorové analýzy geoprvků

Funkce operující nad datovým typem *Geometry*.

Distance()
Buffer()
ConvexHull()
Intersection()
Union()
Difference()

2.6. Forma zápisu prostorových (GIS) dat

OpenGIS WKT a WKB standard

WKT:

- značkovací jazyk WKT (Well Known Text)
- používá se pro
  - popis geometrie objektů
  - definici prostorových referenčních systémů (EPSG:5514)

WKB:

- portovatelný zápis geometrie objektů
- převod dat, výměna mezi SQL/CLI (*Call Level Client*) klientem

Příklady:

- 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))

OpenGIS specifikace vyžaduje aby uložené objekty obsahovaly CRS identifikátor.

Prostorový Referenční Systém (Coordinate Reference System - CRS)
- dataset geodetických parametrů,
- transformace mezi CRS,
- CRS je definované v SRID(),
- CRS objekt je definován jako znakový řetězec nebo platná hodnota PROJ.4

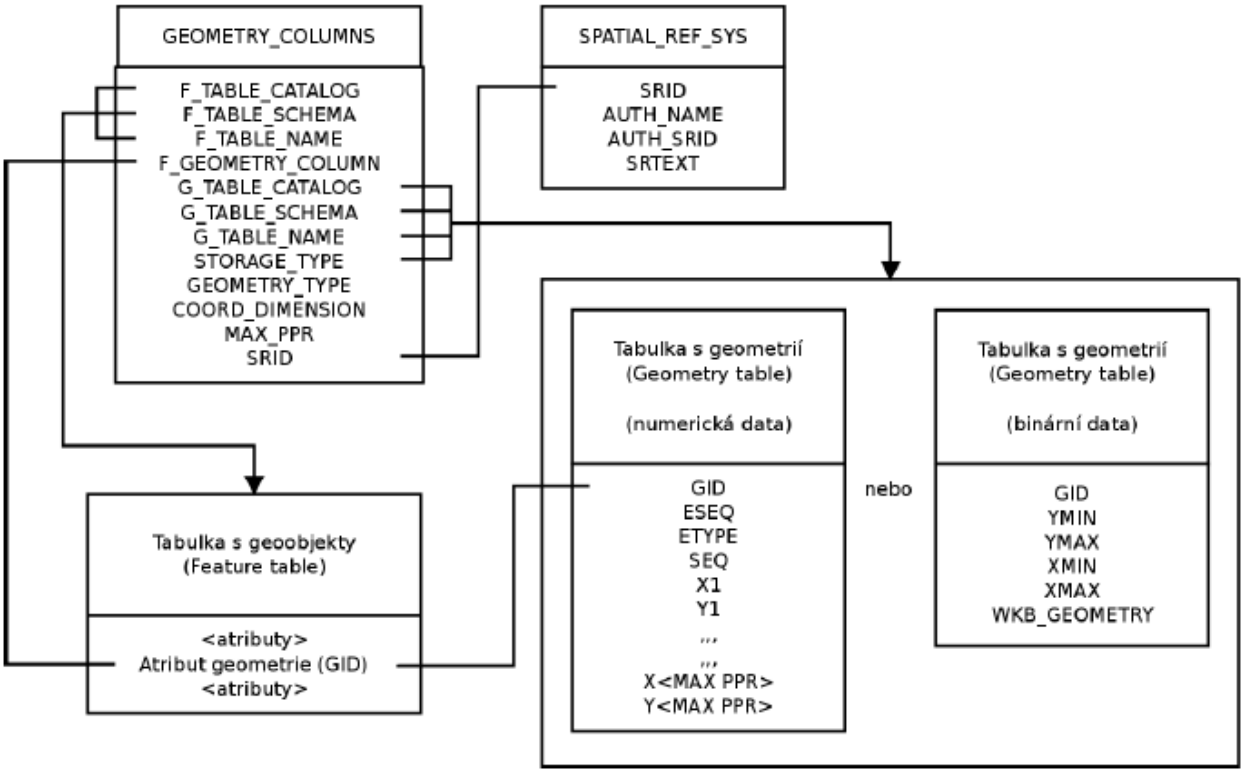
Standard EPSG (http://www.epsg.org/): dataset definicí CRS a souřadnicových transformací. Registr umožňuje přístup k a datům v různých formátech. Je kompatibilní s normou Well-known Text (WKT) ISO 19162.

EPSG: 5514 (S-JTSK)
EPSG: 4326 (WGS 84)
EPSG: 32622 (UTM 33 pás)

2.3. OGC jednoduché geoprvky pro SQL

Podpora pro ukládání, získávání, dotazování a aktualizace geoprvků přes rozhraní SQL/CLI. Tabulka dle specifikace OGC: může být uložena pouze množina prvků stejného geometrického typu. Geoprvek je reprezentován záznamem v tabulce (sloupce: prostorové a popisné informace)

GEOMETRY_COLUMNS – seznam dostupných tabulek s geoprvky
SPATIAL_REF_SYS – definice souřadnicových systémů
Feature table – množina geoprvků jako záznamy v tabulce
Geometry table – geometrie prvků uložena jako standardní numerická či binární data



Obrázek X: SQL schéma jednoduchých geoprvků

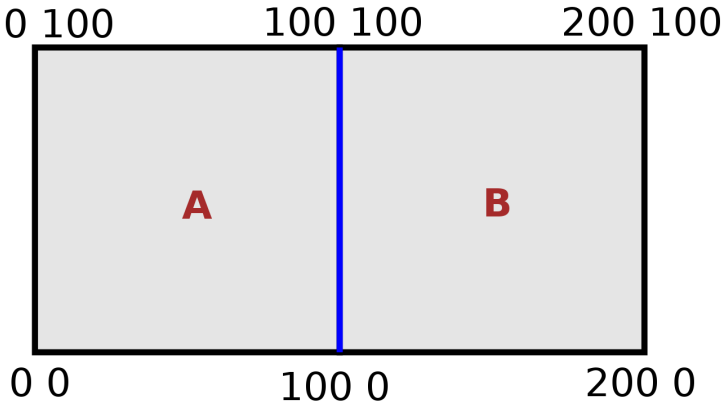
3. Topologie

- vlastnost vektorovych dat;
- lze urcit vzajemne prostorove vztahy;
- popsat charakteristiky dvou vektorovych prvku;

Příklady:
prvek leží v jiném prvku,
prvek se kříži s jiným prvkem,
prvek je nalevo/napravo od prvku,
prvek je shodný s prvkem atd.

Topologicka sprava vektorovych dat
Specifikaci OGC Simple Features for SQL modeluje objekty jako jednoduche geoprvky (simple features), tj. bez topologickeho pohledu na vektorova data, proto je zaveden novy datovy typ Topology.

Topologicka primitiva:
uzly (nodes) — N1, N2
hrany (edges) — E1, E2, E3
steny (faces) — F1, F2



4. Rastry

TBD.

5. Indexy

Slouží k rychlému hledání záznamů v tabulce.

- “rejstřík v knize / knihovně”,
- utříděný seznam klíčů spojený s odkazem na konkrétní místo na pevném disku (datovou stránku)
- minimalizovat porovnávání hodnot.

Pokud DB nemá indexy, jediný způsob, jak najít nějakou informaci, je projít všechny záznamy, dokud požadované údaje nenajdeme.

<u>Typy indexů:</u>

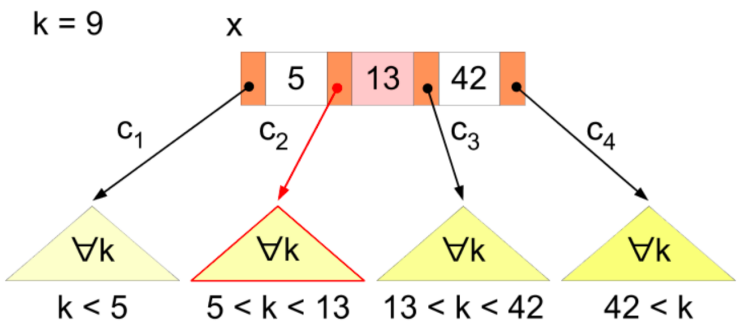
Binární strom:

- orientovaný graf s jedním vrcholem – kořenem,
- uzly mají nejvýš 2 syny;

B-tree:

rozdělují strom na jednotlivé podstromy (c) - dichotomickém větvení,
Pr.: uzel ma 3 potomky → dva klice k1 a k2 ktere budou uzel rozdelovat

- porovnání na každém uzlu: porovnáváme požadovanou hodnotu s hodnotou na uzlu,
- vyhodnocuje: větší / menší
- efektivita: při zdvojnásobení objemu dat přibude jen jedno porovnání navíc
- algoritmus: > a <



<u>B strom algoritmus:</u>

1. Načteme celý uzel
2. Prohledáváme v tomto uzlu zleva doprava všechny klíče, dokud nenalezneme hodnotu klíče, která je stejná nebo větší než požadovaná hodnota

Pokud je hodnota tohoto klíče stejná s hledanou, koncime
Pokud je hodnota tohoto klíče větší a aktuální uzel není listem, zapamatujeme si ukazatel vlevo od tohoto klíče.

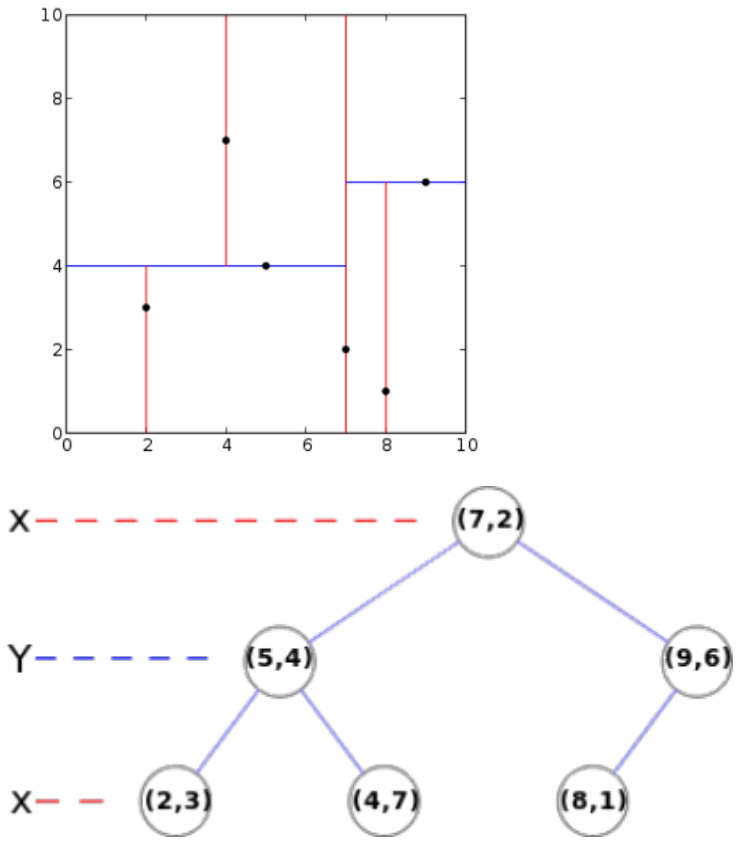
3. V případě, že máme z minulého kroku ukazatel, následujeme ho a pokračujeme krokem jedna.
V případě, že ukazatel není k dispozici, znamená to, že musíme následovat ukazatel, který je nejvíce vpravo (hledaná hodnota je větší než klíč s největší hodnotou v tomto uzlu).

Indexace prostorových dat:
Prostorová data vyžaduji specifické postupy indexovaní, které nejsou běžně implementovány v relačních DB.

Indexové struktury prostorových dat:

a/ KD-stromy
b/ BSP-stromy
c/ R-stromy
d/ Kvadrátové stromy

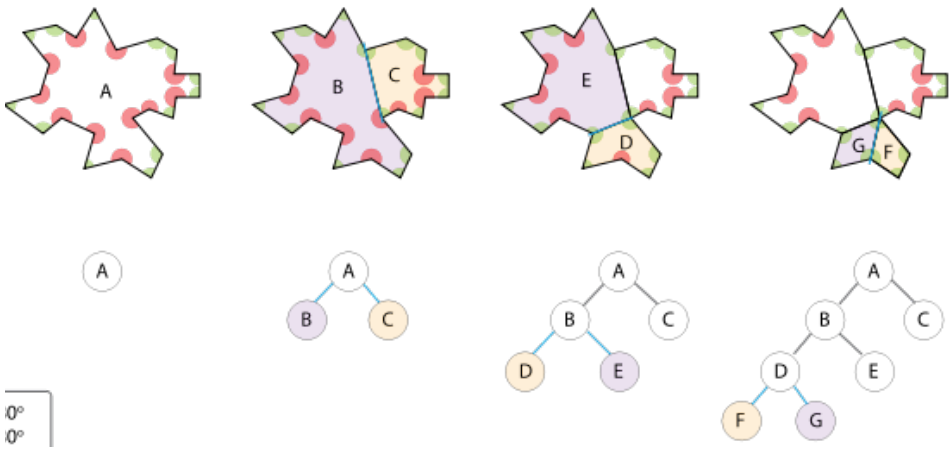
a/ KD-stromy (k-d tree): určené k ukládání bodů (uzlů), specificky pripad binarniho rozdelovani prostoru. Kazdy uzel je k-rozmerny bod.
(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)



Obrázek X: Ilustrace KD-stromu (prostorová a stromová reprezentace)

b/ BSP-stromy: založené na rekurzivní dělení prostoru na dva poloprostory. Binary Space Partitioning

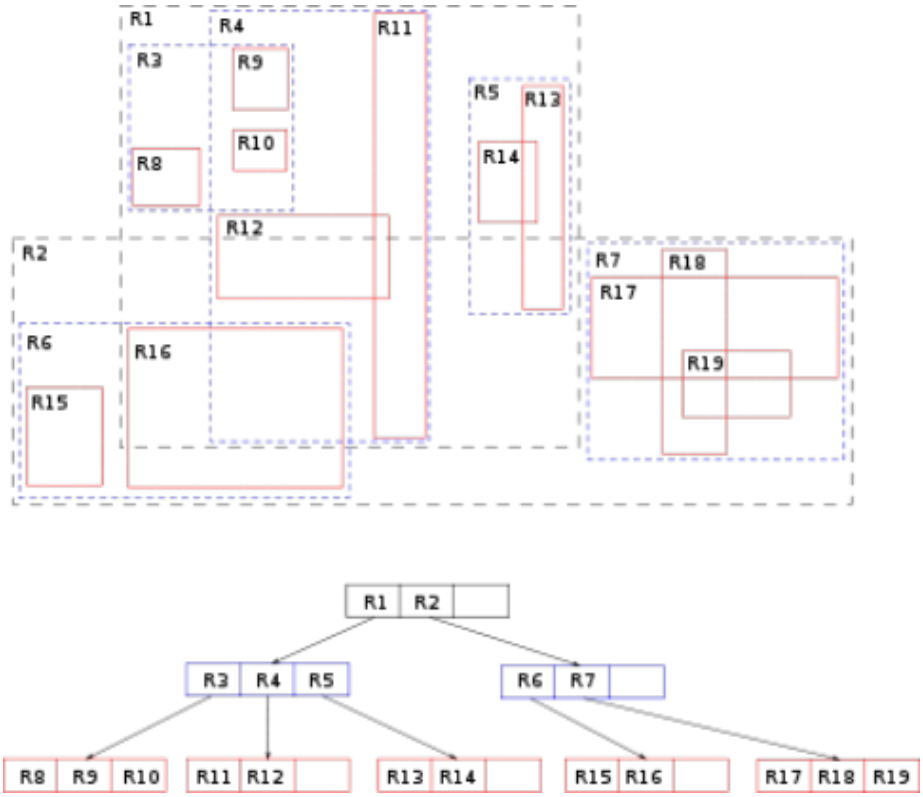
BSP je způsob rozdělení prostoru pomoci binárního stromu. Výsledný strom ve svém kořenu obsahuje rovinu, která všechny objekty v prostoru dělí na 2 skupiny (ležící před a za rovinou). Potomci kořene pak reprezentují vzniklé skupiny, jež jsou opět rekurzivně děleny nově zvolenou rovinou na 2 části. Listy stromů pak obsahují vhodné skupiny objektů



Obrázek X: Ilustrace BSP-stromu

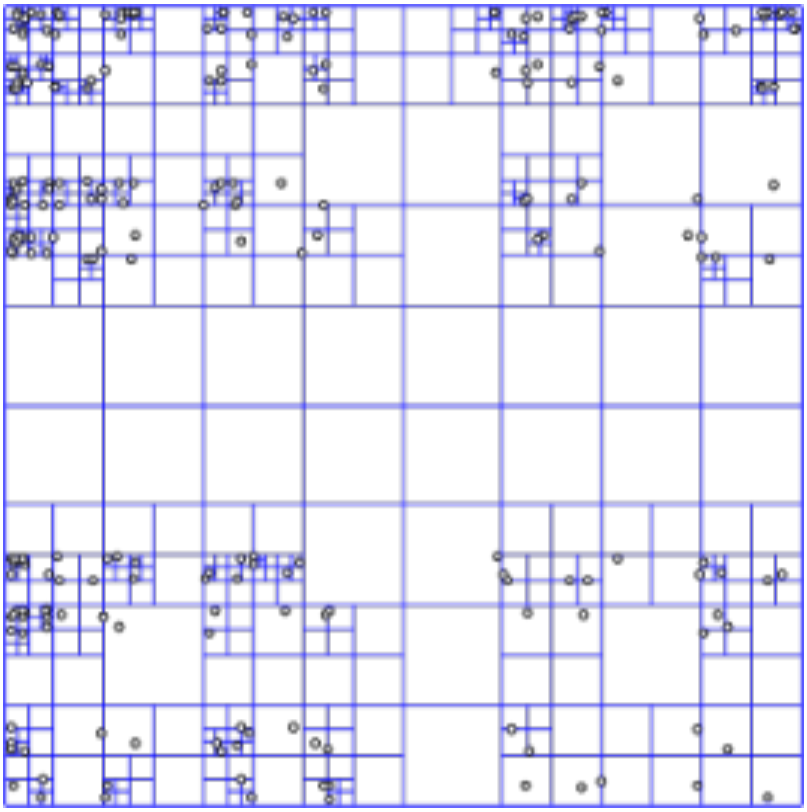
Dvourozměrný případ BSP - pouze pro levé větve stromu.
1. A je kořenem stromu, reprezentuje všechny hrany
2. A je rozdělen na B a C
3. B je rozdělen na D a E.
4. D je rozdělen na F a G, které jsou konvexní, tedy tvoří listy stromu.

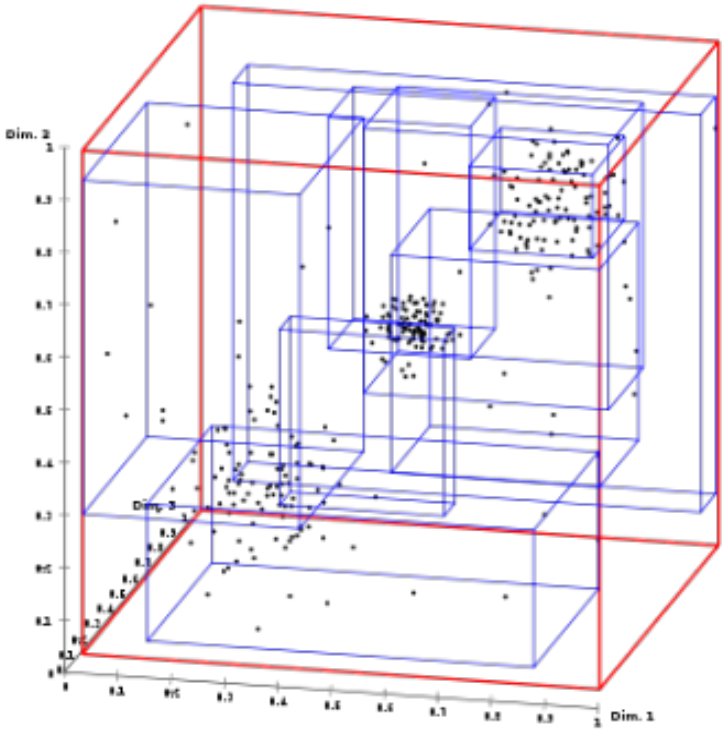
c/ R-stromy: dekompozici prostoru formou minimálních ohraničujících obdélníků, přičemž jednotlivé obdélníky se mohou i vzájemně překrývat. Každý uzel R-stromu má proměnlivý počet záznamů (až do předdefinovaného maxima).



Obrázek X: Ilustrace R-stromu

d/ Kvadratove stromy (Quadtree): založené na postupném dělení prostoru na kvadranty


Obrázek X: Ilustrace kvadrátového stromu



Obrázek X: Ilustrace 3-D R-stromu

GiST (Generalized Search Tree) index
http://www.sai.msu.su/~megera/postgres/gist/

- používá obálku (bounding box) prvků → forma generalizace geoprvků,
- operace jsou jednoduché,
- uložená data malá,
- je možné implementovat R-tree pomocí GiST,
- řeší překryv a vzdálenost prvků.

6. PostGIS

PostGIS je rozšíření objektově-relačního databázového systému PostgreSQL pro podporu geografických objektů. PostGIS implementuje specifikaci Simple Features konsorcia Open Geospatial Consortium.

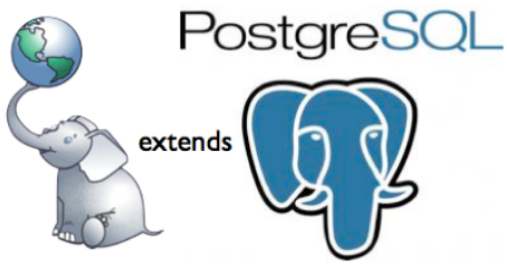
Kromě podpory práce s vektorovými daty v podobě jednoduchých geoprvků (simple features) podporuje PostGIS 2.0 i vektorová data v topologickém formátu a bezešvý přístup k uložení rastrových dat a jejich využití v prostorovém SQL v souvislosti s daty vektorovými.

PostGIS je multiplatformní a plně funkční na platformách jako GNU/Linux, MS Windows či Mac OSX.

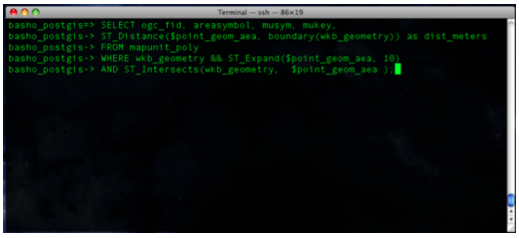
<u>Poč PostreSQL:</u>

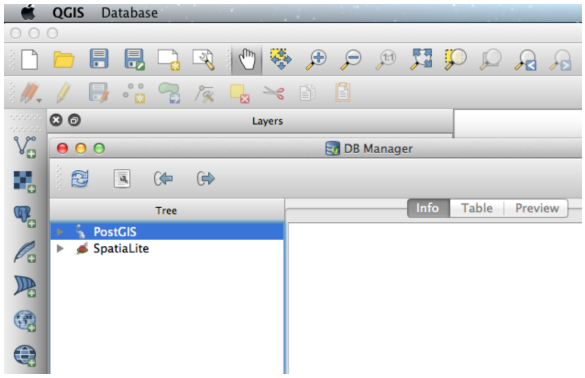
- OS (svobodný software),
- škálovatelnost bez omezení,
- funkčností splňuje požadavky „podnikové“/*enterprise* DB (*hot backup, replication, partitioning, triggers, constraints, foreign keys, user functions, PL/PGSQL, PL/Perl, PL/TCL, PL/Java, PL/R*),
- existuje podpora,
- ORDBMS (object-rational database management system),
- Podporovaná průmyslové standardy (ANSI/ISO SQL, OGC, SQL/MM)

- extenze pro GIS (PostGIS),
- prostorová indexace,
- kompatibilní se standardem OGC SFSQL,
- obohacena o další OS projekty (Proj4, GEOS, GDAL, CGAL/SFCGAL),
- specifická funkčnost: GeoJSON a KML ve webových aplikacích,
- funkce geoprocesingu (150+ operaci s vektory a rastry),
- podpora topologie,
- 3D.



Rozhraní PostGIS
CLI: konzolvý klient psql
GUI: PgAdmin / Qgis





PostGIS OGC jednoduché geoprvky pro SQL
PostGIS podporuje všechny prvky a objekty dle specifikace OGC Simple Feature SQL:

- POINT, MULTIPOINT,
- LINE, MULTILINE,
- POLYGON, MULTIPOLYGON,
- GEOMERTRY COLLECTIONS.

Datové typy
Gemetry
Geograhy
Topology
Raster

6.1. Vytváření databáze s prostorovými tabulkami

- pomocí skriptů,
- extenze (rozšíření)
- zkopírováním již existující prostorové databáze.

Prostorovou databázi můžeme vytvořit pomocí tzv. rozšířeni (extenze) PostGIS.

CLI:
```
createdb mojegdb –h server

psql –d mojegdb –h server --

CREATE EXTENSION postgis;
```

CREATE TABLE ... CONSTRAINT ... PRIMARY KEY ... ;

6.2. Vstupně výstupní procedury

Nástroje pro import a export dat pro PostGIS:

- vkládání GIS objektů pomocí WKT (EWKT)

  --
  INSTER INTO ...
  'SRID=4326;POINT(14.424505675 50.06887390)'
  --

- načítání GIS dat ze souborů:
  - GUI: Qgis, Spit, PgAdmin, shp2pgsql
  - CLI: shp2pgsql, ogr2ogr

  -- Vytvořit dávku ve formátu sql
  shp2pgsql body.shp public.body > sidla.sql
  --

PostGIS EWKB, EWKT
Extended Well Known Text

Příklady:

SRID=32633;POINT(0 0) -- XY s SRID

ST_GeomFromEWKT('SRID=4326;POINT(14.4245 50.0688)'))

6.3. Prostorové funkce a dotazy

Funkce operující nad datovým typem *Geometry*

6.3.1. Geometrické konstruktory (*Geometry Constructors*)

Funkce vkládající a vytvářející geometrii, převádí formáty.

geometry **ST_GeomFromText**(text *WKT*); -- vrací hodnotu ST_Geometry z WKT

-- Příklad
SELECT ST_GeomFromText('POINT(-71.064544 42.28787)');
--

ST_GeomFromEWKT()
ST_MakePoint()
ST_LineFromMultiPoint()

<u>Převody (WKT/WKB):</u>

ST_AsBinary(geometrie) → WKB
ST_AsText(geometrie) → WKT
ST_AsEWKB(geometry) → EWKB
ST_AsEWKT(geometry) → EWKT

6.3.2. Přístup ke geometrii (Geometry Accessors)

Funkce přístupu ke geometrii.

geometry **ST_Envelope**(geometry *g1*); -- vrátí obálku (bbox)

-- Příklad:
SELECT ST_AsText(ST_Envelope('LINESTRING(0 0, 1 3)'::geometry));
--

ST_Boundary() … vytvoří uzavřenou geometrii
ST_SRID() …
ST_XMin() … minimum bbox
ST_Xmax() …
SELECT ST_Xmin()...
ST_NPoints() … počet bodů (vertexů) v geometrii

6.3.3. Prostorové editory (Geometry Editors)

Funkce pro modifikaci existující geometrie.

geometry **ST_Transform**(geometry *g1*, integer *srid*); -- vrati geometrii se souřadnicemi transformovane do jine zvolene prostorove reference

-- Přiklad:
SELECT ST_AsText(ST_Transform(ST_GeomFromText('POLYGON((743238 2967416,743238 2967450,743265 2967450,743265.625 2967416,743238 2967416))',2249),4326)) As wgs_geom;
--

wgs_geom

POLYGON((-71.1776848522251 42.3902896512902,-71.1776843766326 42.3903829478009, -71.1775844305465 42.3903826677917, 71.177582592723142.3902893647987, -71.177684852225142.3902896512902));

ST_AddPoint()
ST_RemovePoint()
ST_Rotate() … rotace geometrie

6.3.4. Výstupy prostorových dat (Geometry Outputs)

Funkce umožňující převod geometrie do jiné reprezentace.

text **ST_AsGML**(geometry *geom*, integer *maxdecimaldigits=15*, integer *options=0*); --vrátí GML (Geography Markup Language) element vstupní geometrie

```
-- Příklad:
SELECT ST_AsGML(ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0
0))',4326));
--

st_asgml
--------
<gml:Polygon
srsName="EPSG:4326"><gml:outerBoundaryIs><gml:LinearRing><gml:coord
inates>0,0 0,1 1,1 1,0
0,0</gml:coordinates></gml:LinearRing></gml:outerBoundaryIs></gml:Polygo
n>
```

ST_AsSVG() … Scalar Vector Graphics
ST_AsGeoJSON() … Geometry Javascript Object Notation

6.3.5. Prostorové operátory (*Operators*)

Operátory řeší vzájemnou polohu minimálních ohraničujících obdélníků (MMO) prvků. Jsou rychlé.

boolean **&&**(geometry *A* , geometry *B*); -- vrací hodnotu ano jestliže se MMO 2D objektů překrývají

```
-- Příklad:
SELECT tbl1.column1, tbl2.column1, tbl1.column2 && tbl2.column2 AS
overlaps
FROM ( VALUES
        (1, 'LINESTRING(0 0, 3 3)'::geometry),
        (2, 'LINESTRING(0 1, 0 5)'::geometry)) AS tbl1,
        ( VALUES
        (3, 'LINESTRING(1 2, 4 6)'::geometry)) AS tbl2;
--

 column1 | column1 | overlaps
---------+---------+------------------
       1 |       3 |        t
       2 |       3 |        f
```

&& ... překryv minimálních ohraničujících obdélníku MMO
<#> … vzdálenost minimálních ohranicujících obdélníku
<-> … vzdálenost centroidu

A && B ... A a B se prekryvaji
A & < B ... A prekryva nebo lezı vlevo od B

6.3.6. Prostorové predikáty (*Spatial Relationships*)
Funkce pro určení vzájemné polohy dvou geometrií.

boolean **ST_Intersects**(geometry *geomA* , geometry *geomB*); -- vrací ano jestliže se geometrie A a B prostorově překrývají (2D)

```
-- Příklad:
SELECT ST_Intersects('POINT(0 0)'::geometry, 'LINESTRING ( 0 0, 0 2 )'::geometry);
--

st_intersects
---------------
 t
```

Topologický model DE-9IM (Dimensionally Extended Nine-Intersection Model)

ST_Touches(), ST_Contains(), ST_Intersects()
ST_Equals(), ST_Disjoint(), ST_Intersects()
ST_Relate() … devítiprvková matice

6.3.7. Rozměr geometrie (*Measurements*)
Funkce pro výpočet geometrických měr.

float **ST_Area**(geometry *g1*); -- vrací hodnotu plochy jestliže je funkce aplikovaná na polygon, v jednotkách SRID

ST_Area() … plocha
ST_Perimetr() … obvod
ST_Length() … délka

6.3.8. Geometrické operace (*Geometry Processing*)
Operace s geometrickými objekty.

`geometry` **`ST_ConvexHull`**`(`geometry *geomA*`)`; -- vrací konvexní obálku geometrie

ST_Union() … sjednocení
ST_Difference() … rozdíl
ST_ExteriorRing() … obvodová hranice
ST_Dump() ... rozdelí multipart geometrii
ST_Polygonize() … zaplochování
ST_ConvexHull() … konvexní obal
ST_Translate() … posun
ST_Buffer() … obalová zóna

6.3.9. Lineární referencovaní
Funkce pro dopočet hodnot mezi uloženými body geometrií.

ST_InterpolatePoint() … vrátí bod interpolovaný z linie.
ST_AddMeasure() … vrátí body lineárně interpolované mezi začátkem a koncem

6.3.10. Ostatní nezařazení funkce

ST_SetSRID … nastaví SRID (ID sořadnicového systému)
FindSRID() …
PostGIS_AddBBox()
PostGIS_DropBBox()

6.3.11. SFCGAL funkce

Knihovna pro pokročilé 2D a 3D funkce.

ST_Orientation()
ST_3DIntersection()
ST_3DDifference()
ST_Volume()

6.4. Prostorový index

PostgreSQL podporuje tři typy indexů: B-trees (nevhodné pro prostorová data), R-Trees (není robustní v PostgreSQL) a GiST.

PostGIS využívá pro indexaci GIS dat R-strom implementovaný nad GiST (Generalized Search Tree).

1. v prvni fazi se vyhledavaji vsechny objekty na zaklade jejich minimalniho ohranicujiciho obdelniku – viz. Operator prekryti “&&”

Poznamka: ST_Intersect(), ST_Contains(), … pouziva tento operator automaticky; _ST_Intersect(), _ST_Contains() nikoliv!

2. v druhe fazi se provede presnejsi dotazovani pouze na podmnozine objektu vybrannych v prvni fazi.

```
-- Příklad
CREATE INDEX [indexname] ON [tablename] USING GIST ( [geometryfield]
);
--
CREATE INDEX idx_geo_buildings ON tbl_buildings USING GIST ( geom );
--
VACCUM ANALYZE tbl_buildings (the_geom) ;
--
```

6.5. Topologie

PostGIS implementuje specifikaci OGC Simple Features for SQL. Modeluje objekty jako jednoduche geoprvky (simple features), tj. bez topologickeho pohledu na vektorova data;

PostGIS Topology (od verze 2.0) implementuje datovy model Topo-Geo z technicke normy SQL (ISO 13249-3:2006).

Topologicka primitiva:

uzly (nodes) — N1, N2
hrany (edges) — E1, E2, E3
steny (faces) — F1, F2

6.5.1. Topologicka sprava vektorovych dat

Schéma topology

Nahrani datovych typu a funkci PostGIS Topology:

```
--
CREATE EXTENSION postgis_topology;

psql -d mojedb -c "CREATE EXTENSION postgis_topology;"
--
```

Tvorba topologickych dat v PostGIS

Každá datová vrstva s topologii je uložena ve zvláštním schématu, nové schéma vytvoříme pomocí funkce CreateTopology. Topologická schéma jsou uložena v tabulce topology.

```
-- nove topologicke schema
CreateTopology('tschema')
--
```

Do schématu vložíme nový atribut, do kterého posléze sestavíme topologii prvků.

```
--
AddTopoGeometryColumn(varchar topology_name, varchar schema_name,
varchar table_name, varchar column_name, varchar feature_type);
--v tabulce vytvoří nový sloupce s názvem topo a datovým typem
TopoGeometry
```

Topologická primitiva sestavíme z jednoduchých prvků pomocí funkce toTopoGeom.

```
--
toTopoGeom(geom,'tschema',tolerance)
--
```

Datovy typ: TopoGeometry reprezentuje geometrii definovanou topologickými primitivy:

- topology_id (id topologického schématu v tabulce topology)
- layer_id (id topologického atributu v tabulce layer)
- id (id topologického primitiva)
- type (geometrický typ jednoduchého prvku):
  1 bod (point)
  2 lomená čára (linestring)
  3 polygon.

Realce 'Node'

node_id integer PRIMARY KEY
containing_face integer REFERENCES Face.face id
geom geometry (Point)

Relace 'Edge'

edge_id integer PRIMARY KEY
start_node integer REFERENCES Node.node_id
end_node integer REFERENCES Node.node_id
next_left_edge integer REFERENCES
next_right edge integer REFERENCES
left_face integer REFERENCES Face.face_id
right_face integer REFERENCES Face.face_id
geom geometry (LineString)

Relace 'Face'

face_id integer PRIMARY KEY
mbr geometry (muze byt prazdna)

Vypis tabulky s topologickými primitivy (v topologickém schématu node, edge a face)

```
-- seznam uzlů
SELECT node_id,containing_face,st_astext(geom) from ...;
-- seznam hran
SELECT edge_id,start_node,end_node,next_left_edge,next_right_edge,
 left_face,right_face,st_astext(geom) from ...;
-- seznam stěn
SELECT face_id,ST_AsText(mbr) from ...;
--

SELECT TopologySummary('topo_tab');
--
```

6.6. Raster v PostGIS

Podporu pro práci s rastrovými daty poskytuje PostGIS až od verze 2.0 a to v rámci rozšíření PostGIS Raster.

Historie: PGRaster - dvoudimenzionalni pole hodnot
- LO: SQL large object
- TOAST: PostgreSQL bytea

PostGIS 2.0+ PostGIS Raster (drive WKTRaster)
Implementace datoveho typu podobneho “geometry” a prislusne funkce

Základní vlastnosti:
Tabulka -> rastrova vrstva (raster coverage)
Zaznam: dlazdice nebo rastrovy objekt

Datovy typ RASTR, rastrova dlazdice:
- velikost bunky (pixel) – prostorove rozliseni
- sirka / vyska
- pocet kanalu (vrstev)
- datovy typ kanalu a hodnota no-data

<u>Sprava rastrovych dat v PostGIS</u>
- IN-DB: uložena přímo v databázi;
- OUT-OF-DB: mimo databázi.

Ulozeni dat v internim formatu: binárně ~WKB
Externe: TIFF, JEPG, … s mozností používat rastrové SQL funkce!

6.6.1. Vstupně výstupní procedury pro rastry

Rutiny:
- raster2pgsql
- pgsql2raster

```
-- Příklad
raster2pgsql -s 5514 dem.tif tab | psql -U user
--
```

-R umožňuje data připojit jako externí zdroj dat
-t (<šířka>x<výška>) … rozdeleni dlazdice

```
--
raster2pgsql -s 5514 dem.tif -F dem > dem.sql
\i clc.sql
--
```

6.6.2. Metadata rastrové vrstvy

Zpřístupňuje informace o rastrových vrstvách (tabulkách) v PostGIS.
- Id projekce
- počet vrstev / kanálů
- datový typ rastru
- šířka / výška, velikost pixelu v ose x, y

- georeference

```
--
SELECT ST_SRID(rast), ST_NumBands(rast),
ST_BandPixelType(rast, 1),
ST_Width(rast), ST_Height(rast), ST_PixelWidth(rast),
ST_PixelHeight(rast), ST_GeoReference(rast)
FROM dem;

SELECT * FROM raster_columns
--
```

6.6.3. Konverze

PostGIS umožňuje konverze dat mezi rastrovou a vektorovou reprezentací.

ST_AsPolygon(raster) → geometry
ST_AsRaster(geometry) → rastr

6.6.4. Základní funkce

Funkce pro modifikaci rastrů a mapovou algebru.

ST_Resample(raster, pixelsize, method) → raster

ST_Clip(raster|geometry, geometry) → typ prvního argumentu

ST_MapAlgebra((raster|geometry, raster), vyraz, "raster"|"geometry")
→ raster/geometry

6.6.5. Konstruktory

Obdoba funkcí pro práci s geometrií.

ST_Intersection()
ST_Union()
ST_Transform()