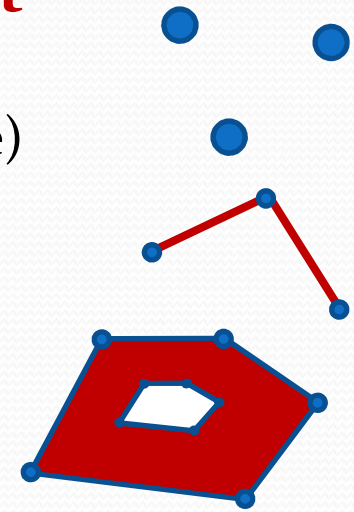
Quoi de neuf dans PostGIS 2.0?
Le raster bien sûr!
Mais beaucoup plus!

Pierre Racine
Professionnel de recherche
Centre d'étude de la forêt
Département des sciences du bois et de la forêt, Université Laval, Québec

PostgreSQL/PostGIS

- **SGBD spatiales Open Source de classe professionnelle**
- **Engin géométriques TRÈS robuste et puissant**
- **Tous les avantages d'un SGBD**
  - Normalisation de la structure des données (redondance)
  - Plusieurs requêtes (usagers) en parallèle
  - Sécurité (accès, transactions, backup)
  - TRÈS GRANDE capacité de stockage (>> shapefile)
- **Stockage de points, lignes, polygones dans des colonnes de type GEOMETRY**
- **Requêtes simples à formuler en langage SQL (~~C, C++, Java~~)**



```
>shp2pgsql c:/temp/buffers.shp buffersTable | psql mydb

SELECT ST_Intersection(l.geom, b.geom)
FROM lakesTable l, buffersTable b
WHERE ST_Intersects(l.geom, b.geom)
```

v. 2.0.0

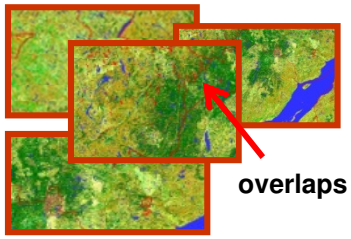
Avril 2012

- **Plus performant que Oracle Spatial, extensible et GRATUIT!**

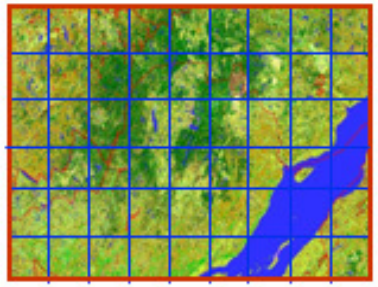
Le type RASTER

- **Principal ajout à la version 2.0.x**
- **Chaque RASTER(ou image ou tuile) est stocké dans un enregistrement dans une colonne de type RASTER**
  - Chaque RASTER (ou tuile) est géoréférencé
  - Une table raster = une couverture raster
  - Tout comme le type GEOMETRY
  - Facile à maîtriser pour les habitués de PostGIS
- **Multibande, un pixeltype par bande, multirésolution, compressé**
- **API SQL très complet**
  - Création, manipulation, traitement et analyse des données raster
  - Tient compte des nodata values
- **Liste des tables RASTER dans la vue raster_columns**
- **Avantages du stockage des rasters dans le SGBD**
  - Accès et manipulation de tous les types de données dans un langage simple et unique (SQL)
  - Opérations très rapides sur de larges jeux de données (tuilé + indexés)

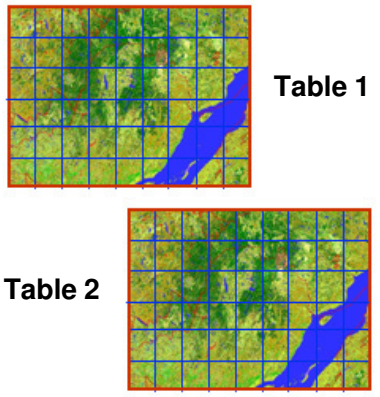
Le type RASTER
Arrangements possibles



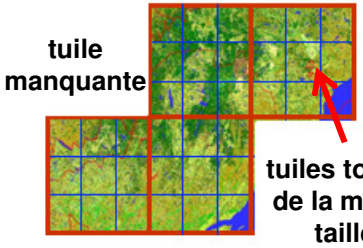
a) Entrepôt d'images satellite ou aériennes) (4 images, 4 enr.)



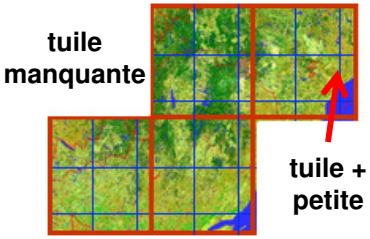
b) Couverture raster rectangulaire régulièrement tuilée (54 tuiles, 54 enr.)



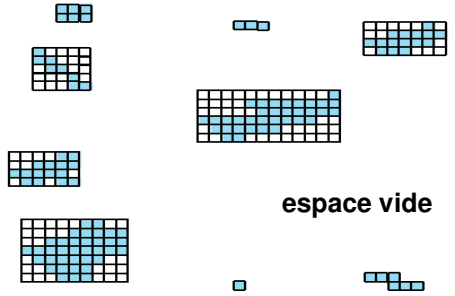
c) Images tuilées (2 tables de 54 tuiles)



d) Couverture raster régulièrement tuilée (36 tuiles, 36 enr.)



e) Couverture raster irrégulièrement tuilée (36 tuiles, 36 enr.)



f) Couverture de géométries rasterizés (9 rasters, 9 enr.)

Le type RASTER Import/Export

- **Importation de rasters en lot**

```
raster2pgsql –t 32x32 –I -C “c:/temp/*.tif” schema.mytable | psql -d testdb
```

  - pas nécessaire de spécifier le SRID
  - l’image sera tuilé 32 pixels par 32 pixels (sur plusieurs enregistrements)
  - un index sera généré
  - les contraintes (permettant de peupler la vue raster_columns) seront ajoutées
  - plusieurs images sont importées en même temps

- **L’export se fait avec GDAL_Translate**

```
gdal_translate -of GTiff PG:"host='localhost' dbname=‘dbname' user='postgres' password=‘xxx' schema='climate' table='maxt' mode=2" D:/temp/maxt.tif
```

  - Mode 1 = ONE_RASTER_PER_ROW
  - Mode 2 = ONE_RASTER_PER_TABLE

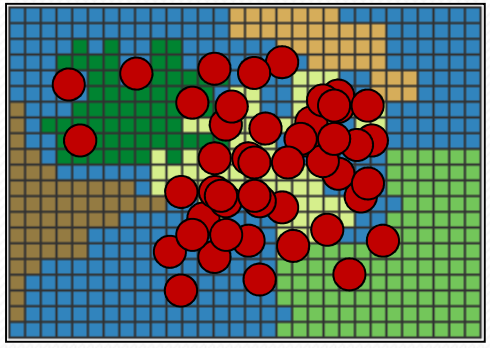
Le type RASTER
Beaucoup de fonctions

- **Accès aux métadonnées**
  - ST_UpperLeftX(), ST_UpperLeftY()
  - ST_Width(), ST_Height()
  - ST_NumBands(), ST_SRID()
  - ST_BandNodataValue(), ST_BandPixelType()
- **Accès aux valeurs**
  - ST_Value(), ST_SetValue()
- **Conversion en GEOMETRY**
  - ST_PixelAsPolygons()
  - ST_DumpAsPolygons()
  - ST_PixelAsPoints(2.1)
  - ST_PixelAsCentroids(2.1)
- **Réchantillonnage**
  - ST_Resample(), ST_Rescale(), ST_SnapToGrid(), ST_Transform()
  - ST_Reclass()
- **Statistiques**
  - ST_Count(), ST_Histogram(), ST_SummaryStats()
- **Conversion en images**
  - ST_AsJPEG(), ST_AsPNG(), ST_AsTIFF(), ST_AsGDALRaster()
- **Opérations géométriques**
  - ST_Clip(raster, geometry)
  - ST_Intersection(raster, geometry)
  - ST_Union(raster set)
- **Map Algebra**
  - Sur un raster
  - Sur deux rasters
  - Avec les pixels voisins sur un raster
  - Avec une expression ou une fonction utilisateur
  - ST_Slope, ST_Aspect, ST_HillShade
- **Relations spatiales**
  - ST_Intersects(), ST_SameAlignment()

Le type RASTER
Opération Overlay Classique

- **Calculer la température moyenne pour chaque polygones d'une couche**

```
SELECT bufID, (gv).geom buffer, (gv).val temp
FROM (SELECT bufID, ST_Intersection(geom, rast) gv
        FROM buffers, temperature
        WHERE ST_Intersects(geom, rast)
```



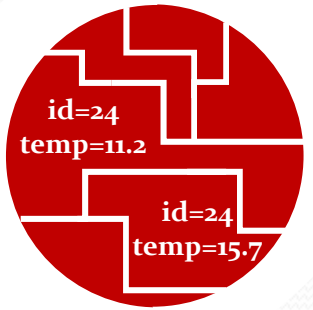
buffers	
geom	pointid
polygon	24
polygon	46
polygon	31
polygon	45
...	...

∩

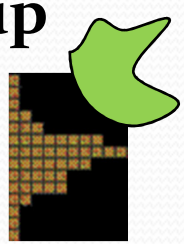
temperature
raster
raster
raster
raster
...

=

result		
geom	pointID	temp
polygon	24	11.2
polygon	53	13.4
polygon	24	15.7
polygon	23	14.2
...	...	...

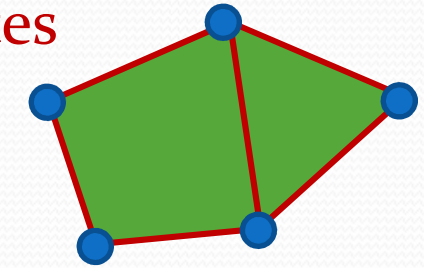


- **Les résultats doivent être aggrégés par buffer après coup**
- **Les analyses tiennent compte des valeurs nodata**
- **Jetez un coup d'oeil au tutoriel dans la page de PostGIS Raster!**



Le type TOPOLOGY

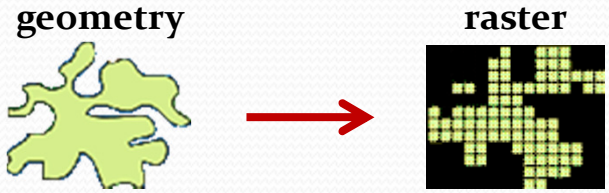
- **Structure de données topologique**
  - Propriétés (ou relations) topologiques explicites basé sur des nœuds, des arêtes et des faces
  - Semblable aux formats coverage et geodatabase de ESRI, Open Street Map
  - Ensemble de fonctions implantant un standard (SQL/MM)
  - Existait déjà mais parachevé dans 2.0



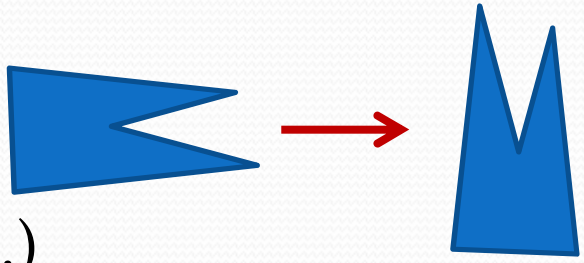
- **Usages**
  - Conserver l'intégrité des relations topologiques lors de l'édition
    - nécessite un logiciel qui fait les bons appels à la BD – » gvSIG
  - Réparer une couche ayant des superpositions ou des trous
  - Simplifier une couche en conservant la topologie

Nouvelles fonctions sur le type GEOMETRY

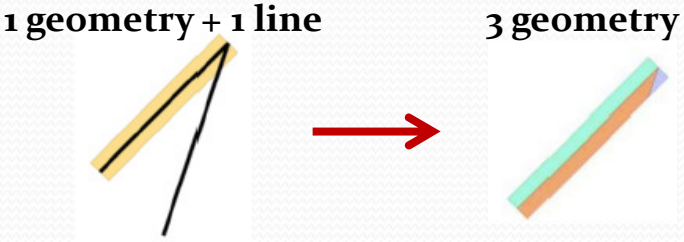
- **ST_AsRaster(rast)**



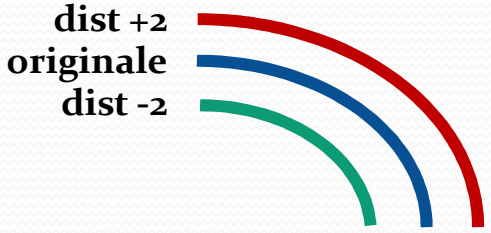
- **ST_MakeValid(geom)**



- **ST_FlipCoordinates(geom)**
  - interverti les X et les Y (long. et lat.)
- **ST_Split(geom1, geom2)**



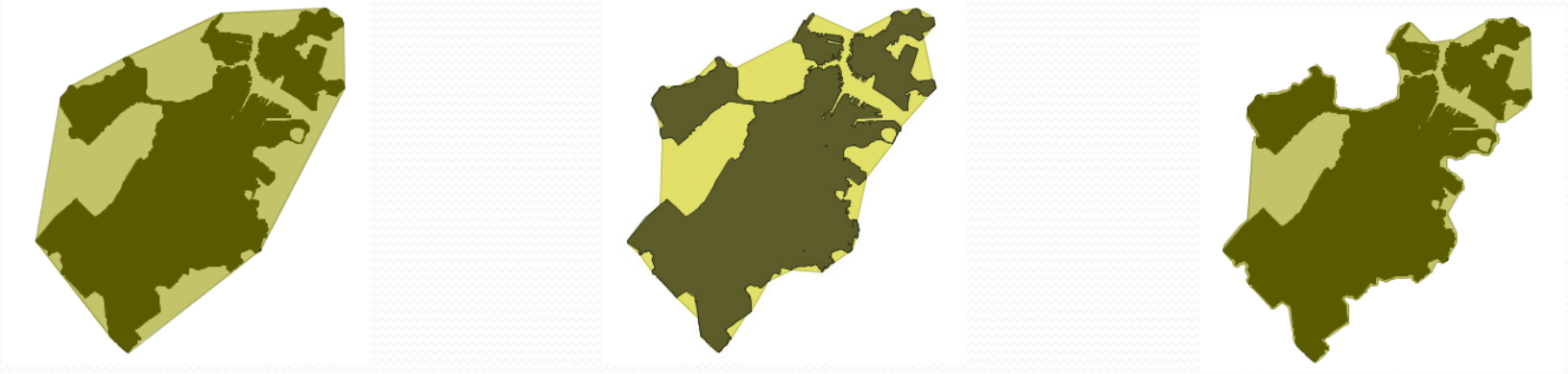
- **ST_OffsetCurve(geom, dist)**



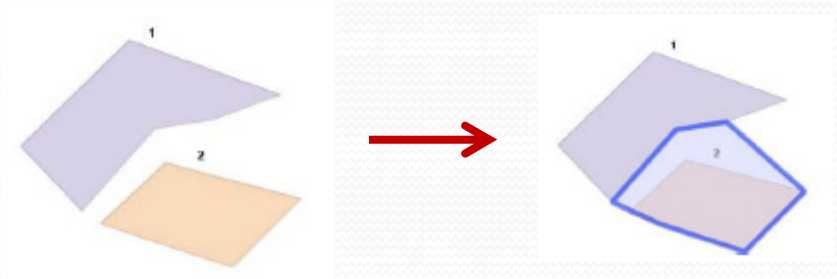
Nouvelles fonctions sur le type GEOMETRY

- **ST_ConcaveHull(geom, pourcent, withholes=FALSE)**

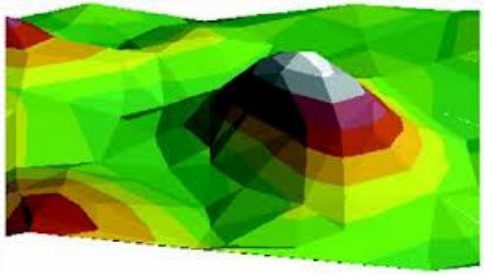
100% **99%** **70%**

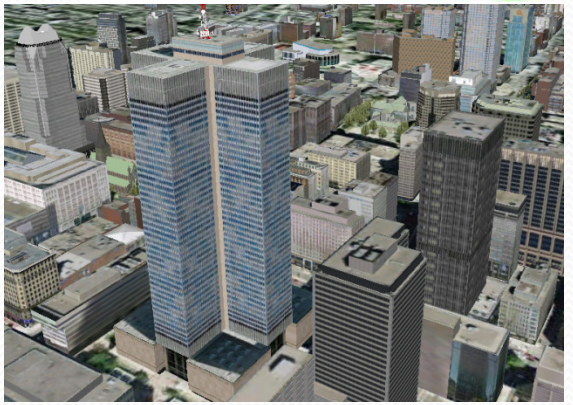


- **ST_Snap(geom1, geom2, distance)**



Les types GEOMETRY 3D





- **Support complet pour Z**
- **Nouveau types de GEOMETRY**
  TRIANGLE, TIN et POLYHEDRALSURFACE
  - ST_3DDistance, ST_3DIntersects, ST_3DLength
  - ST_3DDWithin, ST_3DClosestPoint…
- **Opérateur pour les bounding box 3D: &&&**
- **Nécessite indexage spécial**
  - CREATE INDEX idx ON tbl
    USING GIST (col gist_geometry_ops_nd)
- **Import/Export**
  - shp2pgsql, pgsql2shp et ogr2ogr peuvent importer/exporter les Z
  - Un importeur pour le format 3D CityGML existe
- **Conversion en SQL**
  - Les géométries avec Z peuvent être convertis en KML, GML, X3D et GeoJSON
  - Les TIN et les PS peuvent être convertis en X3d et GML

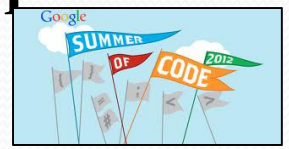
Index K-Nearest Neighbor (KNN)

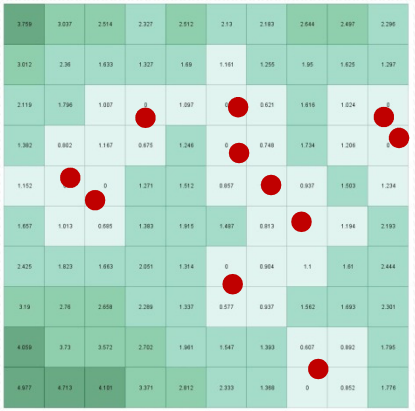
- **Facilite et optimise la recherche des voisins les plus proches**
- **Deux nouveaux opérateurs**
  - <-> : distance entre centroïdes
  - <#> : distance entre bounding box

```
SELECT geom
FROM your_geom_table
ORDER BY geom <-> st_setsrid(st_makepoint(-90,40),4326)
LIMIT 3;
```

  - Résultats exacts si on cherche des points
  - Résultats approximatifs si on cherche des géométries

- **Exemple: Utilisé pour généré efficacement un raster de distance au point le plus proche parmi des millions de points**





Métatables dynamiques

- **La table GEOMETRY_COLUMNS est remplacée par une vue dynamique**
  - Générée à partir d'un « modifieur » du type GEOMETRY (typmod) lors de l'import ou de la création de la table

```
CREATE TABLE geotable (
    geom GEOMETRY(PointZ, 4326)
)
```

  - AddGeometryColumn()
- **Idem pour RASTER_COLUMNS**
  - Générée à partir des multiple contraintes pouvant être appliquées sur les tables raster
  - SRID, scale X et Y, width et height, same alignment, regular blocking, number of band, pixel types, nodata values, out db, extent
  - Option –C à l'import avec raster2pgsql
  - AddRasterConstraints()
- **Deux vues toujours à jour quoi qu'il arrive**

Autres améliorations

- **Installation plus facile**
  - Au lieu d'exécuter postgis.sql, spatial_ref_sys.sql, rtpostgis.sql et topology.sql

```
CREATE EXTENSION postgis;
CREATE EXTENSION postgis_topology;
```

  - Déplacement: ALTER EXTENSION postgis SET SCHEMA postgis;
  - Déinstallation: DROP EXTENSION postgis;
  - Mises à jour plus faciles

- **Import/Export de plusieurs fichiers/tables en même temps dans l'interface graphique du loader (GUI)**

- **Un nouvel agrégateur de blogs: Planet PostGIS**

La mauvaise nouvelle…

- **La structure physique des géométries ayant changée pour accommoder**
  - Plus de types de géométrie
  - Des bounding box 3D
  - L'ajout d'un numéro de version
  - Un meilleur alignement des données physiques

- **Il est nécessaire de faire un hard upgrade de ses données PostGIS**

```
>pg_dump -Fc -b -t yourtable yourbd > C:/temp/backup.dump
>
>pg_restore -d yourbd C:/temp/backup.dump
```

PostGIS 2.1.0 (printemps 2013?)

- **Intégration de pgRouting (pledge en cours, Boston GIS)**
- **SELECT ST_DelaunayTriangles(ST_Collect(geom))**
  FROM your_geometry_table
- **ST_AddBand(rast, ARRAY[]) de plusieurs bandes**
- **ST_Union(raster) plus rapide et multi-bandes**
- **Plus d'opérateur relationnels entre RASTERs**
  - ST_Contains, ST_ContainsProperly, ST_Covers, ST_CoveredBy, ST_Disjoint, ST_Overlaps, ST_Touches, ST_Within, ST_DWithin, ST_DFullyWithin
- **ST_Distance(), ST_Dwithin() et ST_Intersects() plus rapides pour le type GEOGRAPHY**
- **Meilleur support (distance, intersection) pour les géométries de type CURVE**
- **Intégration de la librairie CGAL pour un meilleur traitement des géométries 3D**
- **Builds automatiques pour Windows et Debian**

Questions

Merci!

Geospatial Elucubrations (blog)

@geoelucubration

cef Centre d'étude de la forêt

UNIVERSITÉ LAVAL