Introduction à l'analyse spatiale dans PostGIS en langage SQL

Journée SIG
19 nov. 2014
Centre GéoStat
Université Laval

cef
Centre d'étude de la forêt

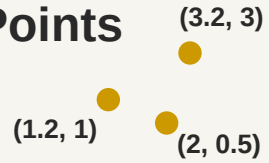
Pierre Racine
Professionnel de recherche

Représentation numérique du monde géographiques

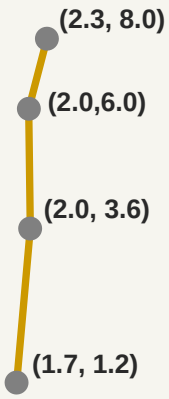
- Les objets du monde géographique sont représentés par les points, des lignes, des polygones et des rasters

Vectorielle

Points



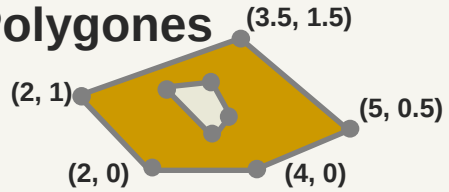
Lignes



Attributes of meso_puertos

FID	Shape *	NOMBRE	COD_MUN	UBICACION
0	Point	LIMON	0701	LIMON,LIMON
1	Point	MOIN	0701	LIMON,LIMON
2	Point	PUNTARENAS	0601	PUNTARENAS, PUNTAREN
3	Point	CALDERA	0602	ESPARZA, PUNTARENAS
4	Point	GOLFITO	0607	GOLFITO, PUNTARENAS
5	Point	QUEPOS	0606	AGUIRRE, PUNTARENAS
6	Point	JIMENEZ	0607	GOLFITO, PUNTARENAS
7	Point	ARMUELLES	0402	BARU, CHIRIQUI
8	Point	PEDREGAL	0406	DAVID, CHIRIQUI
9	Point	MUTIS	0906	MONTIJO, VERAGUAS
10	Point	VACAMONTE	0801	ARRAIJAN, PANAMA
11	Point	BALBOA	0808	PANAMA, PANAMA
12	Point	OBALDIA	1001	KUNAYALA, COMARCA KU
13	Point	COLON	0301	COLON, COLON
14	Point	S/N	0304	PORTOBELLO, COLON
15	Point	S/N	0103	CHIRIQUI GRANDE, BOCAS
16	Point	S/N	N/D	CORN ISLAND, RAAS

Polygones



Matricielle (rasters, images)

38	44	34	38	100	47	80	95	96	78	46
42	44	33	49	123	139	134	133	106	46	32
49	57	43	39	84	135	105	116	109	65	36
63	60	48	52	93	99	85	108	123	104	47
73	76	48	41	42	109	89	101	131	134	87
50	65	59	51	48	63	83	70	58	43	41
45	45	48	49	38	42	39	39	45	52	50
45	49	48	44	44	47	51	34	43	52	60
40	40	46	50	43	40	44	52	51	54	64
40	37	33	47	43	36	40	51	50	49	59
39	41	37	43	46	35	37	39	46	48	55

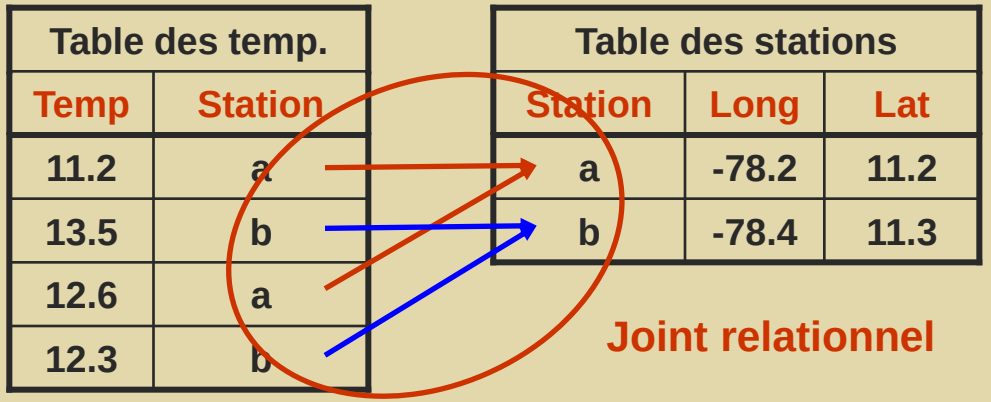
Tables de données VS bases de données

- **Table de données**
  - Excel, Google SpreadSheet ou CSV
- **Base de données**
  - Système **complet et complexe** de gestion de données (**SGBD**)
  - Multi-usager, client-serveur, sécurité et backup, typage et règles, transaction.
  - Plusieurs tables **mises en relation** (élimination des redondances)

Table de données

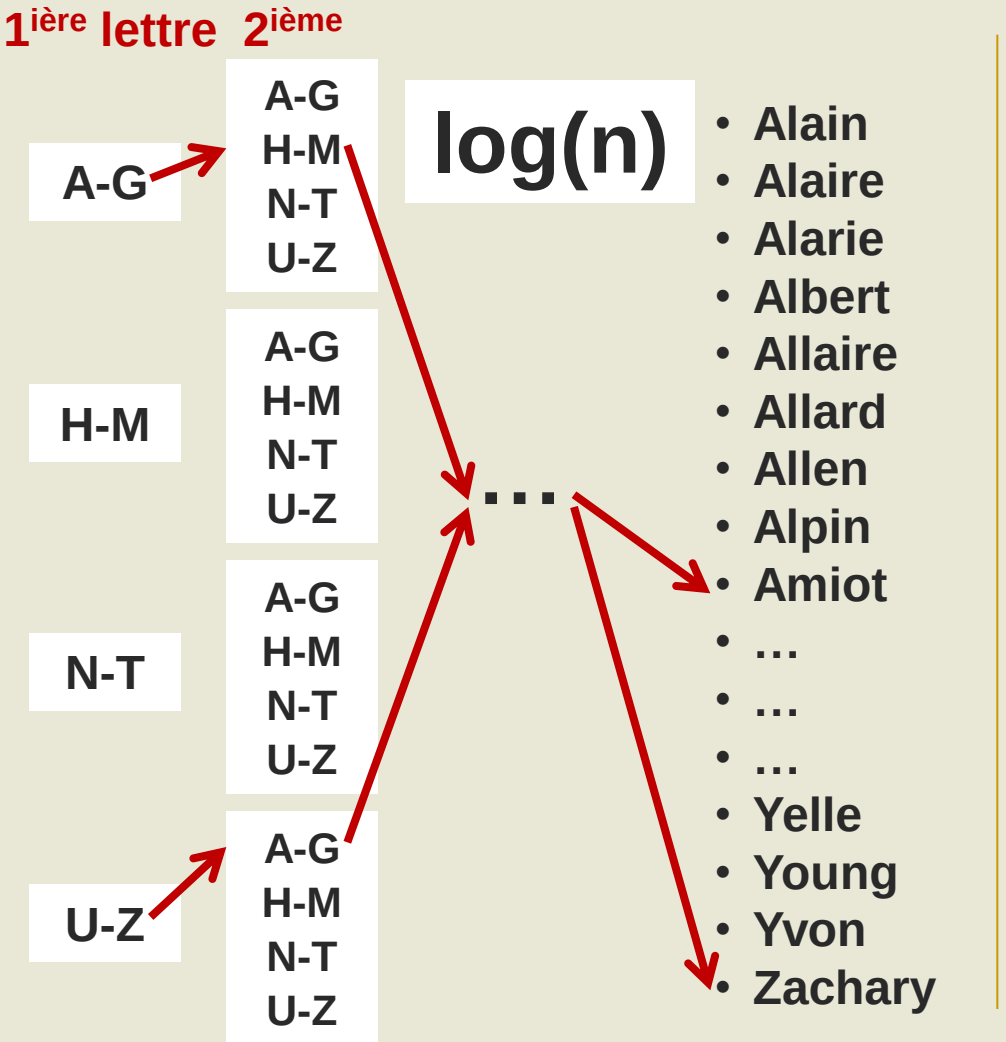
Table des températures			
Temp	Station	Long	Lat
11.2	a	-78.2	11.2
13.5	b	-78.4	11.3
12.6	a	-78.2	11.2
12.3	b	-78.4	11.3

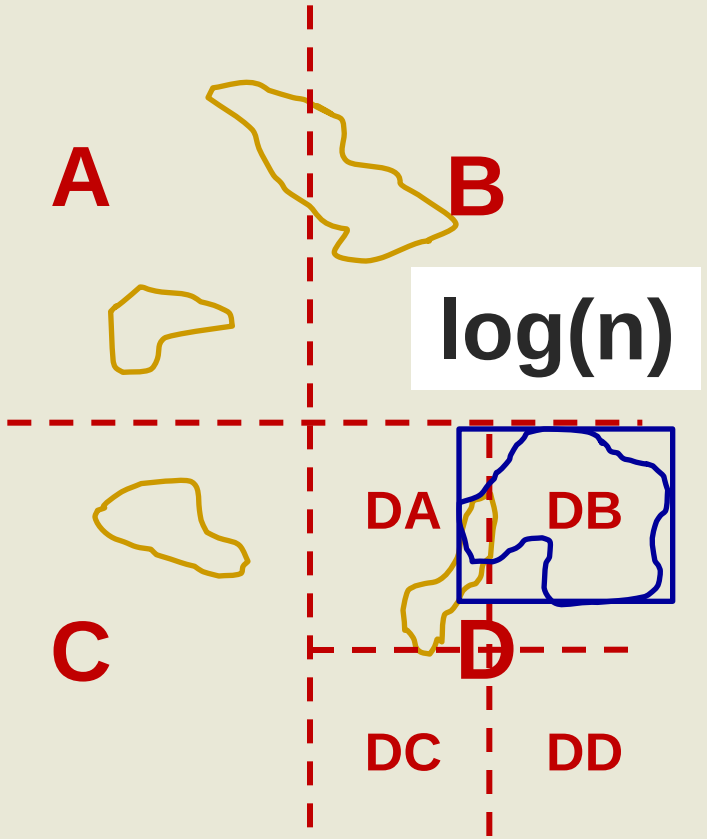
Base de données



Index et index spatial

- Permet de trouver rapidement une entitée parmi des millions





Identification des systèmes de coordonnées

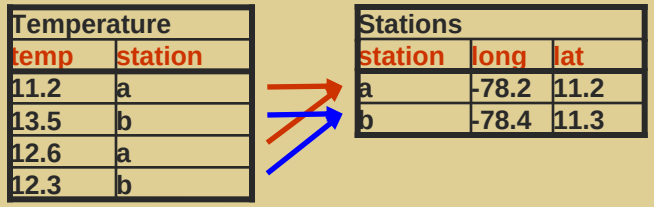
- Chaque SCR a un numéro standard établie par différents organismes
  - **EPSG** - European Petroleum Survey Group
  - **IGNF** - Institut Géographique National de France
  - **OSGEO** - Open Source Geospatial Foundation
- Quelques SCR fréquemment utilisés
  - EPSG: **4326** - WGS 84
  - EPSG: **4269** - NAD83
  - EPSG: **26901** à **26923** - NAD83 / UTM zones 1N à 23N
  - EPSG: **3347** - NAD83 / Statistics Canada Lambert
  - EPSG: **32181** à **32197** - NAD83 / MTM zones 1 à 17
  - EPSG: **32198** - NAD83 / Quebec Lambert
- Stockés dans une table dans une base de données spatiale
  - **spatial_ref_sys** dans PostGIS

Langage SQL - I

Interroger les tables d'une BD relationnelle (requêtes)

- Filtrer les colonnes et les lignes **(SELECT et WHERE)**
- Joindre plusieurs tables entre elles **(FROM)**
- Aggréger des lignes ensemble **(GROUP BY)**
- Trier les lignes dans un certain ordre **(ORDER BY)**

Tables en relation



Temperature	
temp	station
11.2	a
13.5	b
12.6	a
12.3	b

Stations		
station	long	lat
a	-78.2	11.2
b	-78.4	11.3

Résultat

TemperatureParStation			
temp	station	long	lat
11.2	a	-78.2	11.2
12.6	a	-78.2	11.2
13.5	b	-78.4	11.3
12.3	b	-78.4	11.3

```
SELECT Temperature.temp, Stations.station, Stations.long, Stations.lat
FROM Temperature, Stations
WHERE Temperature.station = Stations.station
ORDER BY Stations.station
```

Langage SQL - II

- Chaque nouvelle colonne peut être fonction d'une ou plusieurs colonnes existantes
  - SELECT quantite * prixunitaire AS prixtotal…
  - SELECT abs(max - min) AS range, round(temperature) AS temp…
- Sur une colonne de type TEXT
  - SELECT prenom || " " || nom AS nom,
    left(prenom) || left(nom) AS initiales
- Sur une colonne de type DATE
  - SELECT date_part("month", birth) AS month…
- Fonctions aggrégatives
  - SELECT avg(temp) AS meantemp
    FROM tempTable
    GROUP BY date_part("month", birth)

Fonctions PostGIS spatiales

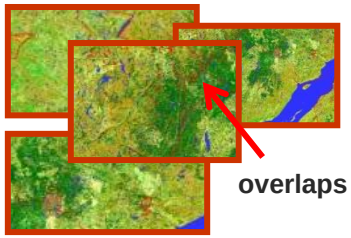
- **Sur une colonne de type GEOMETRY**
  - SELECT ST_Centroid(geom) AS centroid…
  - SELECT ST_Area(ST_Transform(geom, 32181))
  - SELECT ST_Union(geom) AS regadmin
    FROM mrc
    GROUP BY adminname
  - Plus de 200 fonctions sur les GEOMETRY
- **Sur une colonne de type RASTER**
  - SELECT ST_SummaryStats(rast) stats…
  - SELECT ST_DumpAsPolygons(rast) geom…
  - SELECT ST_MapAlgebra(rast) rast…
  - SELECT ST_Clip(ST_Union(rast), geom)…
  - Plus de 100 fonctions sur les RASTER

Chargement des données spatiales

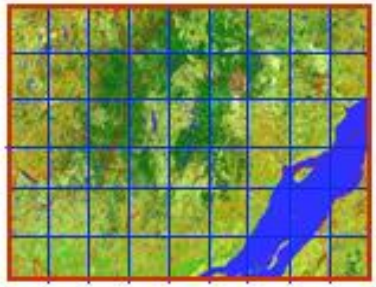
- **Vectoriel (GEOMETRY)**
  - **shp2pgsql -s 42304 -I "shapefile" "targetTable" | psql -U "postgres" -d "database"**
  - **"s": numéro standard du système de coordonnées**
  - **"I": créé automatiquement un index sur les géométries**
  - **"|": pipe le résultat dans psql qui exécute le fichier de commande dans la BD**

- **Matriciel (RASTER)**
  - **raster2pgsql -t 32x32 -I -C "rasterfile" "targetTable" | psql -U "postgres" -d "database"**
  - **"t": tuile le raster sur plusieurs ligne afin de profiter de l'index spatial sur les tuiles**
  - **"C": génère des contraintes permettant de peupler la tables des métadonnées**

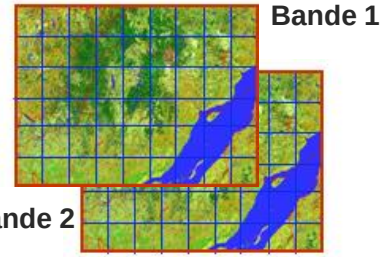
Arrangements RASTER possibles



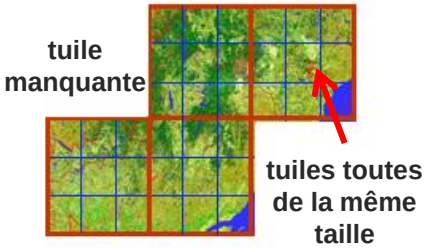
a) Entrepôt d'images satellite ou aériennes) (4 images, 4 enr.)



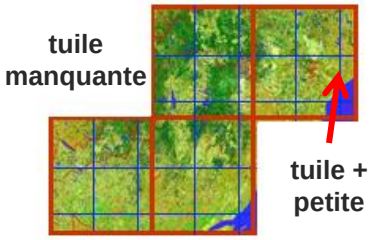
b) Couverture raster rectangulaire régulièrement tuilée (54 tuiles, 54 enr.)



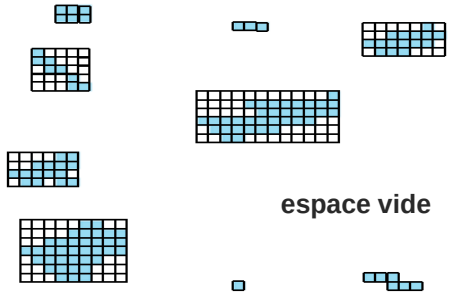
c) Image multi-bandes tuilées (1 table de 108 tuiles)



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.)

Tables de métadonnées

- Permet à certaines applications de lister rapidement les tables contenant des colonnes de type GEOMETRY ou RASTER avec leurs propriétés principales.
- GEOMETRY
  - schéma, nom de la table, nom de la colonne GEOMETRY
  - SRID
  - type de la GEOMETRY (Point, MultiLinestring, MultiPolygon, Geometry)
- RASTER
  - schéma, nom de la table, nom de la colonne RASTER
  - SRID
  - taille et type des pixels, taille des tuiles, nodata value, in-out
  - bien alignées, disposée régulièrement, extent

Autre structures spatiales
Autre APIs

- **Autres APIs**
  - **pgRouting – Permet de faire de l'analyse dans un réseau linéaire. Chemin le plus court, planification d'itinéraires.**
  - **Geocoding – Associer des adresses à des point dans un réseau de routes.**

- **Autres structures de données**
  - **3D – Support presque complet pour les objets en vrai 3D**
  - **Point Cloud – Import et interrogation de nuage de point. Peu de fonction d'analyse pour l'instant. Conversion en point préalable.**
  - **Topology – Format topologique comparable au format ESRI Coverage et geodatabase. Permet de…**
    - **Conserver l'intégrité des relations topologiques lors de l'édition**
    - **Réparer une couche ayant des superpositions ou des trous**
    - **Simplifier une couche en conservant la topologie**

Pourquoi utiliser une base de données spatiale?

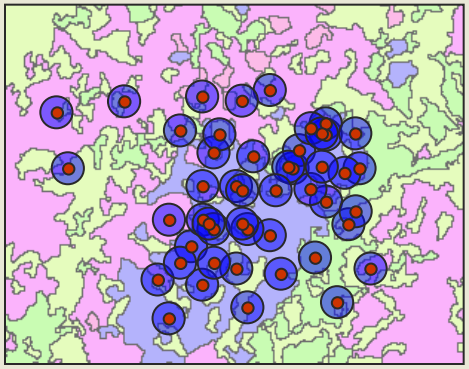
- **Gestion de données à la SGBD**
  - Tous **les avantages** d'un SGBD pour la gestion de l'info spatial.
  - Sécurité, multi-usager, intégrité, volume (complexité).
- **Paradigme unique**
  - **Un seul langage** (SQL) plus simple que Java, Python, C pour exploiter les données tabulaires, vectorielle, matricielle, nuage de point, 3D, topology…
  - L'API sur le type **RASTER** est très **similaire** à celui sur le type **GEOMETRY**.
  - Excellente **interopérabilité** avec une panoplie d'autre logiciels.
- **Performance et robustesse pour l'analyse**
  - Toutes les requêtes utilisent les **indexes** (normal et spatial)
  - Fonctions géométriques très robustes **tester par des milliers d'utilisateurs** depuis plusieurs années.
    - Plus robuste et souvent plus rapide que ArcGIS
    - Plus rapide que R
    - Supporte de plus gros volumes de données

Pourquoi stocker le raster dans la BD?

- **Interaction raster/vecteur**
  - Nos données vecteur sont normalement **déjà** dans la BD…
- **Performance**
  - Traitement sur rasters **tuilés plus rapide** que sur rasters non tuilés.
- **Volume**
  - Possibilité de travailler avec des couverture de l'ordre du **TB**.
- **Possibilité de garder le raster stocké dans le système de fichier**
  - Et de l'exploiter d'une manière **transparente** en SQL.
  - Seulement les **métadonnées** (extent, SRID, pixel type, nodata) sont stocké dans la BD. Les valeurs sont lus de manière **transparente** à partir des fichiers référencés.

Requêtes types
Intersections vecteur/vecteur

- Permet d'extraire, pour une série de points, de lignes ou de polygones, les valeurs sous-jacentes provenant d'une autre couche.
- Ex. dans quels types de couverture se retrouvent une série de polygones



observ	
geom	obsid
polygon	24
polygon	31
polygon	45
…	…

∩

cover	
geom	ctype
polygon	4
polygon	3
polygon	5
polygon	2
…	…

=

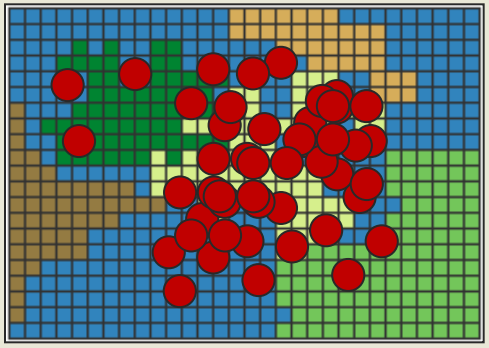
result			
geom	obsid	ctype	area
polygon	24	4	10.34
polygon	53	3	11.23
polygon	24	5	14.23
polygon	23	2	9.45
…	…	…	…

```
SELECT obsid, ctype, ST_Area(geom) area, geom
FROM (SELECT ST_Intersection(o.geom, c.geom) geom, obsid, ctype
        FROM observations o, couvert c
        WHERE ST_Intersects(o.geom, c.geom)) foo;
```

- On procède généralement ensuite à une analyse statistique classique afin de déterminer s'il existe une corrélation entre les types identifiés et les superficies mesurées.

Requêtes types
Intersections vecteur/raster

- Permet d'**extraire**, pour une série de points, de lignes ou de polygones, les valeurs sous-jacentes provenant d'une couche raster.
- Ex. calculer la température moyenne pour chaque polygones d'une couche



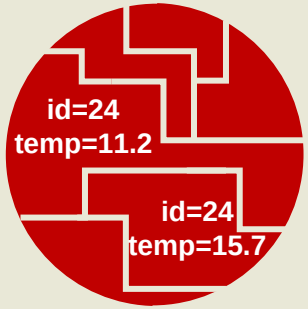
buffers	
geom	pointid
polygon	24
polygon	46
polygon	31
polygon	45
…	…

∩

temperature
raster
raster
raster
raster
raster
…

=

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



Mode vecteur (les pixels sont découpés)

```
SELECT bufid,
(ST_AreaWeightedSummaryStats(gv)).*
FROM (SELECT ST_Intersection(rast, geom) gv
        FROM temperature, buffer
        WHERE ST_Intersects(rast, geom)) foo
GROUP BY bufid;
```

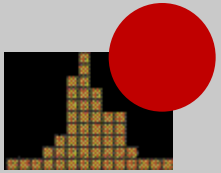
- lorsque taille polygones beaucoup < taille des pixels
- couche vectorielle = lignes
- moins rapide, plus précis

Mode raster (les pixels ne sont pas découpés)

```
SELECT bufid,
(ST_SummaryStatsAgg(ST_Clip(rast, geom, true))).*
FROM temperature, buffer
WHERE ST_Intersects(rast, geom)
GROUP BY bufid;
```

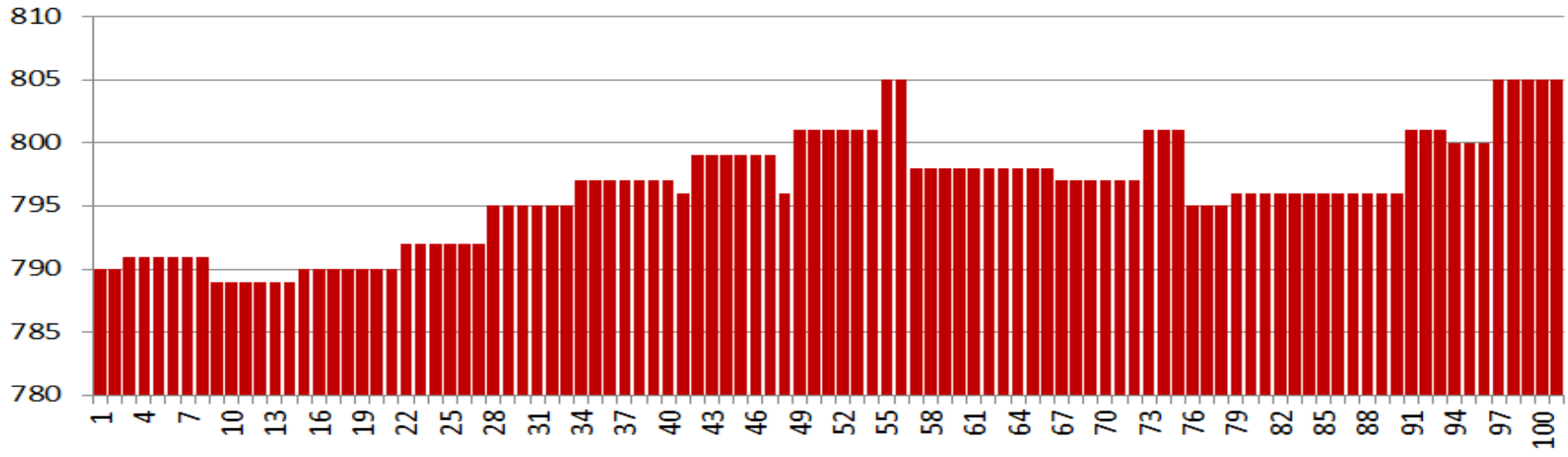
- lorsque taille polygones beaucoup > grande taille des pixels
- seulement couches de polygones
- plus rapide, moins précis

Les intersects tiennent compte des **nodata**



Requêtes types
Elevation profile

```
CREATE TABLE elevation_profile AS
WITH points AS (
   SELECT id, ST_LineInterpolatePoint(geom, id/100.0) geom
   FROM generate_series(0, 100) id, (SELECT (ST_Dump(geom)).geom
   FROM routes_fm_utm7 WHERE id = 2058) foo
)
SELECT id, geom, ST_Value(rast, ST_Transform(geom, 4326)) elev
FROM points, e_elevation_fm_10x10_wgs84
WHERE ST_Intersects(rast, ST_Transform(geom, 4326));
```



Requêtes types – Proximité I

- **Déterminer les N géométries les plus proches d'un ensemble de géométries**
- **Les N POINTs de 1 POINT (méthode classique)**

```
SELECT pointB.geom, pointB.id,
        ST_Distance(pointA.geom, pointB.geom) dist
FROM pointtableA pointA, pointtableB pointB
WHERE pointA.id = 999 AND ST_DWithin(pointA.geom, pointB.geom, 100)
ORDER BY dist
LIMIT 3;
```

Très difficile, voire impossible à déterminer!

- **Les N POINTs de 1 POINT (méthode KNN avec l'opérateur <->)**

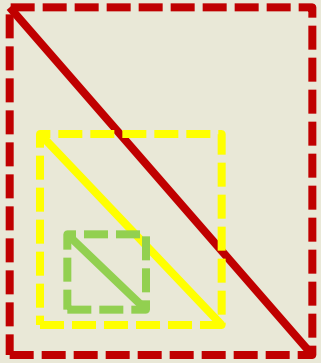
```
SELECT pointB.geom, pointB.id, ST_Distance(pointA.geom, pointB.geom) dist
FROM pointtableA pointA, pointtableB pointB
WHERE pointA.id = 999
ORDER BY (SELECT geom FROM pointtableA WHERE id = 999) <->
pointB.geom
LIMIT 3;
```

Requêtes types – Proximité II

- Les N POINTs de tous les POINTs (KNN)

```
SELECT pointA.id, pointA.geom, pointB.id id2,
        ST_Distance(pointA.geom, pointB.geom) dist
FROM pointtableA pointA, pointtableB pointB
WHERE pointB.id = ANY ((SELECT array(
                        SELECT id FROM pointtableB
                        ORDER BY geom <-> pointA.geom
                        LIMIT 3
                        ))::integer[])
ORDER BY pointA.id, dist;
```

- Les N GEOMETRY de toutes les GEOMETRY (KNN)
  - Problème: l'opérateur <#> n'opère que sur les extents
  - Un extent d'une géométrie plus éloignée peut toujours être plus proche que la géométrie la plus proche.
  - Solution…



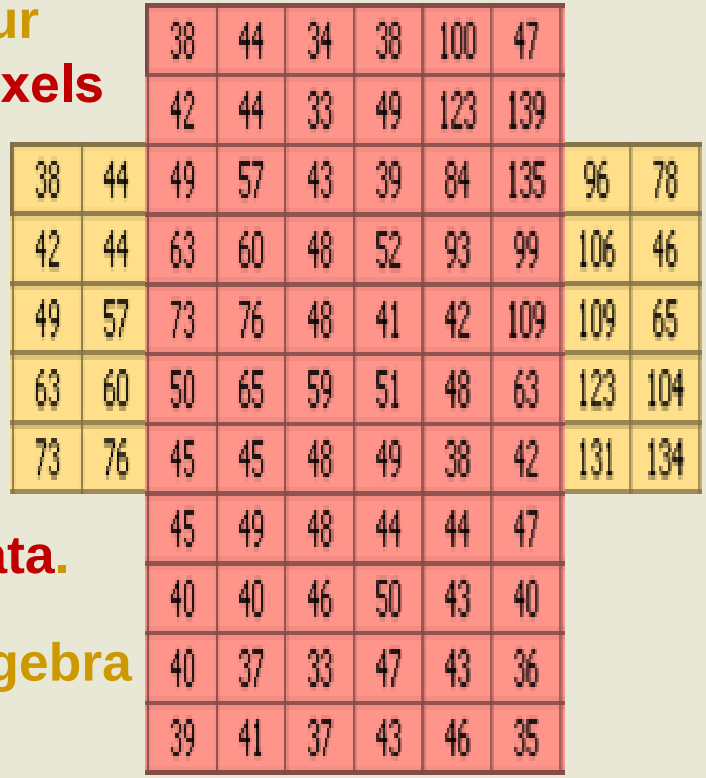
Requêtes types – Proximité IV

- Les N GEOMETRY de toutes les GEOMETRY (KNN)

```
WITH first100 AS (
  SELECT pointA.id, pointA.geom, pointB.id id2,
              ST_Distance(pointA.geom, pointB.geom) dist
  FROM pointtableA pointA, pointtableB pointB
  WHERE pointB.id = ANY ((SELECT array(
                                  SELECT id FROM pointtableB
                                  ORDER BY geom <#> pointA.geom
                                  LIMIT 30))::integer[])
  ORDER BY pointA.id, dist
) , ordered AS (
SELECT *, ROW_NUMBER() OVER (PARTITION BY id ORDER BY dist) rownum
FROM first100
)
SELECT * FROM ordered WHERE rownum < 4;
```

Requêtes types – Map Algebra I

- **Opération classique d'analyse avec des rasters**
  - **Le raster en sortie résulte d'une d'expression ou d'une fonction custom SQL évaluée sur chaque pixel ou l'entourage de chaque pixels de un ou deux rasters en entrée.**
  - **L'extent résultant peut être celui du 1er raster, du 2ième raster, de l'intersection ou de l'union des deux rasters.**
  - **On peut contrôler explicitement ce qui se passe quand une des valeurs est nodata.**
  - **Plusieurs fonctions dépendent de MapAlgebra**
    - **ST_Clip(rast, geom)**
    - **ST_Intersection(rast, rast)**
    - **ST-Union(rast)**
    - **ST_Aspect(), ST_Hillshade(), ST_Slope(), ST_Roughness()**



Requêtes types – Map Algebra II

- **Fusionner plusieurs rasters ensemble**

Disjoints

```
SELECT year, ST_Union(rast) rast
FROM tiledrastseries
GROUP BY year;
```

Superposés

```
SELECT year, ST_Union(rast, 'MEAN') rast
FROM tiledrastseries
GROUP BY ST_UpperLeftX(rast);
```

- **Calculer les ombrage à partir d'une couverture d'élévation tuilée**

```
SELECT ST_HillShade(ST_Union(e2.rast), 1, e1.rast, '32BF', 180) rast
FROM elev e1, elev  e2
WHERE ST_Intersects(e1.rast, e2.rast)
GROUP BY e1.rast;
```

- **Reclassification d'un raster 32BF en 8BUI, 255 = nodata**

```
SELECT ST_Reclass(rast,
        ROW(1, '0-500:1-10, (500-10000: 10-254', '8BUI', 255)::reclassarg) rast
FROM rasttable;
```

Requêtes types – Map Algebra II

- **Même reclassification avec ST_MapAlgebra()**

```
SELECT ST_SetBandNodataValue(ST_MapAlgebra(rast, '8BUI',
  'CASE
    WHEN 0 <= [rast] AND [rast] <= 150 THEN round(10 * [rast] / 150.0)
    WHEN 150 < [rast] AND [rast] <= 254 THEN 10 + round(10 * ([rast] - 150)/(254 - 150))
    ELSE 255
   END', 255), 255) rast
FROM hillshade_fm;
```

- **Ajouter la hauteur des arbres à une couche d'élévation**

```
SELECT ST_MapAlgebra(e.rast, fc.rast, '[rast1] + [rast2]', NULL,'INTERSECTION')
FROM elevation e, forestcover fc
WHERE ST_Intersects(e.rast, fc.rast);
```

Requêtes types
Rasterization d'une couche vectorielle I

```
CREATE TABLE ramsafe_welltiled_forestcover_rast AS
WITH forestrast AS (
  SELECT rid, ST_MapAlgebra(
            ST_Union(ST_AsRaster(geom, rast, '32BF', height, -9999)),
            ST_AddBand(ST_MakeEmptyRaster(rast), '32BF'::text, -9999, -9999),
            '[rast1]', '32BF', 'SECOND') rast
  FROM forestcover, elevation
  WHERE ST_Intersects(geom, rast)
  GROUP BY rid, rast
)
SELECT a.rid,
      CASE
        WHEN b.rid IS NULL THEN ST_AddBand(
                        ST_MakeEmptyRaster(a.rast), '32BF'::text, -9999, -9999)
        ELSE b.rast
      END rast
FROM elevation a LEFT OUTER JOIN forestrast b
ON a.rid = b.rid;
```

- Seule la valeur au centroïde du pixel peut être extraite
- Seule la moyenne des valeurs peut être calculée lorsque deux pixel se superposent

Requêtes types
Rasterization d'une couche vectorielle I

- **Avec ST_ExtractToRaster() de PostGIS Addons**

```
CREATE INDEX forestcover_geom_gist ON forestcover USING gist (geom);

CREATE TABLE extracttoraster_forestcover AS
SELECT ST_ExtractToRaster(
                ST_AddBand(
                    ST_MakeEmptyRaster(rast), '32BF'::text, -9999, -9999),
                'public',
                'forestcover',
                'geom',
                'height',
                'MEAN_OF_VALUES_AT_PIXEL_CENTROID'
              ) rast
FROM elevation;
```

Vectorization facile!

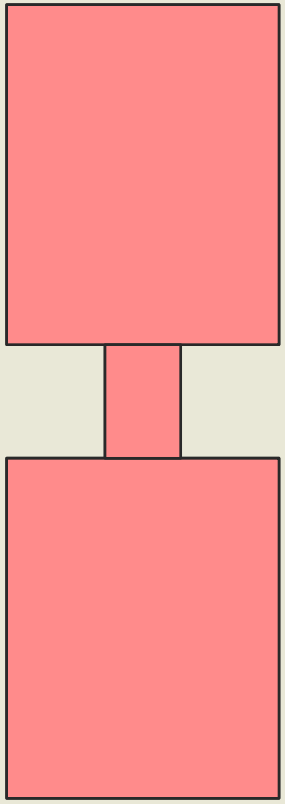
```
SELECT  (ST_DumpASPolygons(rast)).*
FROM rast
```

- **Plusieurs autres méthodes…**
  - **Facile d'en ajouter de nouvelles**

Requêtes types
Nettoyage des superpositions

- Enlever toutes les parties communes aux autres polygones en s'assurant de le faire d'une manière ordonnée.

```
SELECT a.id, ST_DifferenceAgg(a.geom, b.geom) geom
FROM overlappingtable a,
          overlappingtable b
WHERE ST_Equals(a.geom, b.geom) OR
     ((ST_Contains(a.geom, b.geom) OR
       ST_Contains(b.geom, a.geom) OR
       ST_Overlaps(a.geom, b.geom)) AND
      (ST_Area(a.geom) < ST_Area(b.geom) OR
       (ST_Area(a.geom) = ST_Area(b.geom) AND
        ST_AsText(a.geom) < ST_AsText(b.geom))))
GROUP BY a.id;
```



- Nécessite l'installation de PostGIS Addons

Les grands manquants

- Méthodes d’interpolation
- Analyse de raster de coût et chemin le plus court
- Meilleure intégration dans QGIS

Deux projets réalisé avec PostGIS
1) Boreal Data Foundry

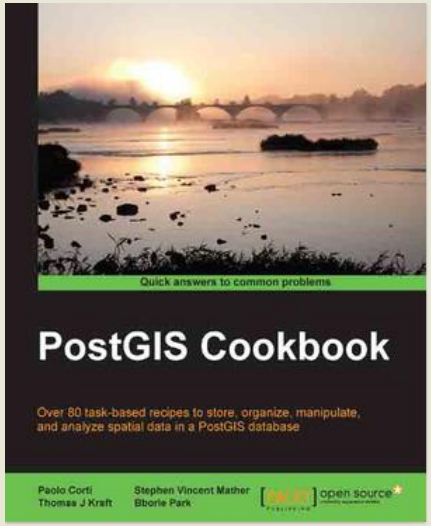
- Standardisation des **26 millions de polygones** de peuplement forestiers provenant d'une **vingtaine d'inventaires provinciaux** différents.
- Une trentaine de couche écologiques couvrant le Canada
  - élévation, température, précipitation, routes, etc…
- Beaucoup d'observations animales
  - **100 000** observations de caribou
  - **76 000** observations d'oiseau
  - **2500** transects d'observation de la sauvagine
- Extraction des quantités associées aux variables écologiques pour chaque observation

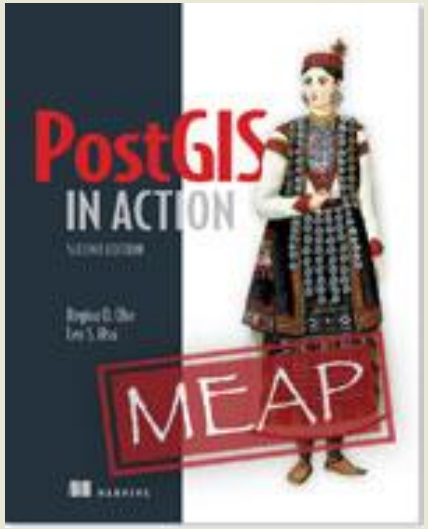
Deux projets réalisé avec PostGIS

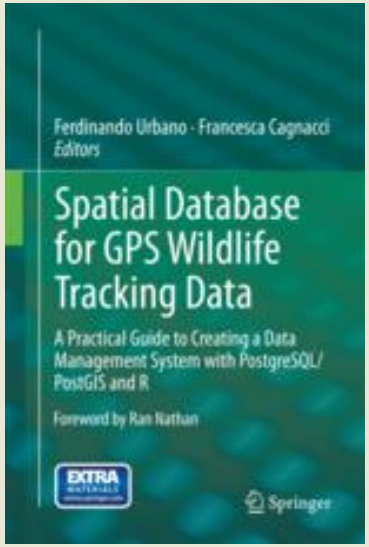
2) Linear network multiscale generalization

- Généralisation **automatique** en SQL d'un réseau linéaire selon l'échelle affichée.
- Les nœuds du réseau et l'information qu'ils contiennent sont **agrégés** lors des zoom arrières.
- Plus difficile que la simple généralisation par regroupement de points
  - Les point regroupés ensemble doivent non seulement être près les uns des autres, mais aussi être **consécutifs** dans le réseau
- Itinéraires de voyageurs
  - Tout autre type d'itinéraire ayant de l'information associé à chaque nœud

Ressources







- Documentation de PostGIS
- Tutoriels en ligne
- Groupe de discussion postgis-users
- GIS Stack Exchange
- PostGIS Planet
- Geospatial Elucubration

Thanks!

http://trac.osgeo.org/postgis/wiki/WKTRaster

Boreal Avian Modelling Project

The Canadian BEACONs Project
Boreal Ecosystems Analysis of Conservation Networks

western boreal conservation initiative
Environment Canada
l'initiative de conservation boréale de l'ouest
Environnement Canada

Fondation canadienne pour l'innovation
Canada Foundation for Innovation

PostgreSQL

PostGIS
Geospatial Objects for PostgreSQL

Cadcorp
WORLD LEADING GIS SOFTWARE

deimos

PARAGON CORPORATION

azavea

MichiganTech
Create the Future

cef
Centre d'étude de la forêt

UNIVERSITÉ LAVAL

España Virtual