
Fonctionnalités avancées

Nicolas Ribot - Licence GNU FDL - Version 1.2

- ❖ Requêtes spatiales avancées
- ❖ Utilisation avancée des index spatiaux
- ❖ Outils topologiques complexes (ST_RELATE)
- ❖ Validation et nettoyage topologique des données géographiques
- ❖ Rôles, privilèges
- ❖ Configuration
- ❖ Maintenance
- ❖ Sauvegarde
- ❖ Réplication



Requêtes spatiales avancées, index spatiaux

- ❖ Trouver tous les objets situés à une certaine distance d'un point. (st_distance)
- ❖ Refaire la même requête en utilisant l'index spatial et un filtre spatial
- ❖ Comparer les temps de résultats
- ❖ cf. TP avancé 1.



- ❖ Lister toutes les points formant tous les polygones des communes de France:
- ❖ `st_numInteriorRing`, `st_geometryN`, `generate_series`, ...
- ❖ cf. TP avancé 1



- ❖ Reconstruire les départements de France à partir des limites administratives.
- ❖ `st_buildArea` vs `st_polygonize`
- ❖ cf. TP avancé 1



- ❖ Matrice d'intersection (DE-9IM) Dimensionally-extended, 9 intersection matrix

	Interior	Boundary	Exterior
Interior	$\dim(I(a) \cap I(b))$	$\dim(I(a) \cap B(b))$	$\dim(I(a) \cap E(b))$
Boundary	$\dim(B(a) \cap I(b))$	$\dim(B(a) \cap B(b))$	$\dim(B(a) \cap E(b))$
Exterior	$\dim(E(a) \cap I(b))$	$\dim(E(a) \cap B(b))$	$\dim(E(a) \cap E(b))$

Where:

T == {0,1,2}

F == empty set

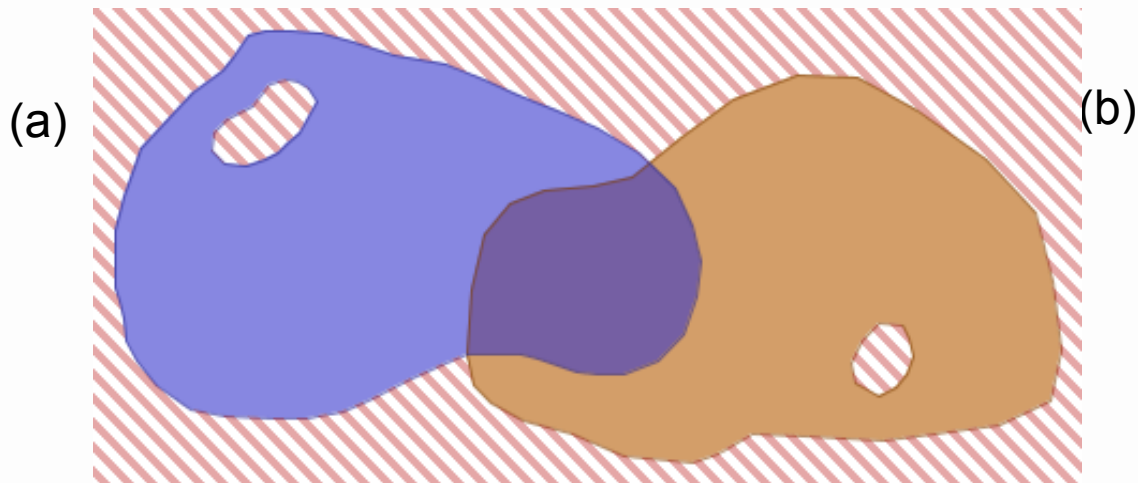
* == don't care

0 == dimensional 0 - point

1 == dimensional 1 - line

2 == dimensional 2 - area

Relations spatiales et mesures: st_relate



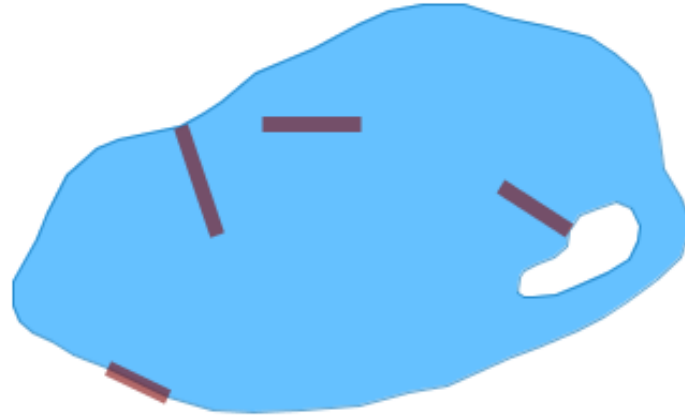
Interior Boundary Exterior

Interior	2	1	2
Boundary	1	0	1
Exterior	2	1	2

ST_Relate(a, b) = '212101212'

Relations spatiales : st_relate

- ❖ Exemple: Trouver tous les pontons inclus dans le lac, ne touchant pas les berges



Quels prédicats utiliser ? ST_Within? ST_Contains? ST_Touches?

```
SELECT a.id
FROM docks a, lakes b
WHERE a.geom && b.geom
AND ST_Relate(a.geom, b.geom, 'TFFTFF212');
```


Validation et nettoyage topologique des données géographiques

- ❖ `ST_IsValid`, `ST_IsValidReason` pour tester la validité des géométries
- ❖ `ST_Buffer(geometry, 0.0)` peut parfois corriger des (Multi) Polygones invalides, ex:
- ❖ Update matable set `geom = st_buffer(geom, 0.0) where not st_isValid(geom);`

- ❖ Représente la notion d'**utilisateur** de la base, ou de **groupe d'utilisateurs**, suivant la définition du rôle
- ❖ Indépendant des utilisateurs du système d'exploitation
- ❖ Permet le contrôle de l'accès aux objets de la base
- ❖ Création d'un rôle: CREATE rôle ...
- ❖ Modification d'un rôle: ALTER ROLE ...
- ❖ Suppression: DROP ROLE ...

- ❖ **LOGIN**: Le rôle peut se connecter à la base. Equivalent de **USER**
- ❖ **SUPERUSER**: Le super utilisateur contourne tous les contrôles. A utiliser avec précaution: créer un autre rôle non superuser mais administrateur
- ❖ **CREATEDB**: Le rôle peut créer des bases de données
- ❖ **CREATEROLE**: Le rôle peut créer de nouveaux rôles
- ❖ **PASSWORD**: Précise le mot de passe du rôle. N'est utilisé qu'avec les méthodes d'authentification **PASSSSWORD**, **CRYPT**, **MD5** (cf. `pg_hba.conf`)
- ❖ Modification d'un rôle: **ALTER** rôle ...

- ❖ Par défaut, tout objet (table, vue, index, ...) d'une base appartient à l'utilisateur l'ayant crée
- ❖ Seul cet utilisateur peut manipuler cet objet
- ❖ Des privilèges peuvent être donnés pour permettre l'accès aux objets
- ❖ Privilèges : Actions qu'un rôle a le droit d'exécuter
- ❖ SELECT, INSERT, UPDATE, DELETE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE, USAGE
- ❖ **GRANT** permet de donner des privilèges
- ❖ **REVOKE** supprime des privilèges



- ❖ Les groupes permettent de gérer plus facilement les privilèges
- ❖ Pour créer un groupe d'utilisateur:
- ❖ Créer un rôle (sans LOGIN ou avec):
- ❖ `CREATE ROLE name;`
- ❖ Ajouter ou supprimer des utilisateur a ce groupe:
- ❖ `GRANT group_role TO role1, ... ;`
- ❖ `REVOKE group_role FROM role1, ... ;`
- ❖ Le rôle PUBLIC ne peut pas être ajouté à un rôle
- ❖ Les privilèges d'un rôle peuvent être hérités ou non aux membres du rôle

- ❖ Le fichier postgresql.conf contient toutes les variables de configuration de la base
- ❖ La plupart de ces variables sont surchargeables au niveau d'une base, d'un utilisateur, d'une connexion.
- ❖ Conseil de réglages:

<http://postgis.refrations.net/pipermail/postgis-users/2007-October/017309.html>

<http://postgis.refrations.net/pipermail/postgis-users/2006-March/011539.html>

<http://www.postgresql.org/docs/8.2/interactive/kernel-resources.html>

- ❖ Vacuum: récupérer les espaces vides dans les tables
- ❖ Analyse: collecte et met à jour les statistiques sur les tables (index, usage)
- ❖ A lancer régulièrement sur les tables subissant des UPDATE et DELETE fréquents
- ❖ Peut être lancé automatiquement avec AUTOVACUUM



- ❖ Sauvegarder les bases de données !!
- ❖ pg_dump: utilitaire de sauvegarde d'une base
- ❖ Génère du SQL pouvant être exécuter sur une nouvelle base
- ❖ Dispose de nombreuses options (schéma seul, données seules, choix des objets, etc)
- ❖ Restauration du dump: `psql -f dump.bck postgres`
- ❖ pg_dumpall pour sauvegarder toutes les bases d'une instance
- ❖ Option de compression pour les gros volumes
- ❖ pg_restore pour restaurer une sauvegarde compressée
- ❖ PostgreSQL 9.0: réplication à chaud 😊

- ❖ Duplique une BD vers d'autres BD
- ❖ Mode maitre/esclave
- ❖ Replication en cascade:
 - ❖ Un esclave peut se répliquer à son tour
- ❖ Plusieurs modes de réplication disponibles:
 - ❖ Streaming
 - ❖ Synchronous
 - ❖ Log Shipping
 - ❖ ...
- ❖ Assez facile à mettre en place
- ❖ Se couple a pgPool (pooling, heartbeat, load balancing)

