eBook Gratuit

APPRENEZ postgis

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#postgis

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: postgis

It is an unofficial and free postgis ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official postgis.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec Postgis

Remarques

PostGIS est un ensemble d'extensions pour la base de données PostgreSQL. Avec PostGIS, vous pouvez stocker des données géospatiales et effectuer des requêtes spatiales sur une base de données postgres.

Contrairement aux types de données par défaut d'une base de données postgres normale, les données spatiales doivent être gérées différemment. Les requêtes que vous pouvez effectuer sur une base de données spatiale sont généralement définies par des boîtes englobantes en 2 ou 3 dimensions. Pour stocker, indexer et gérer ces types de données, postGIS utilise un concept appelé R-Trees , qui ne fait pas partie de l'installation postgres par défaut.

Avec une base de données postGIS, vous pouvez:

- stocker des données spatiales
- effectuer des requêtes spatiales pour récupérer et extraire des informations (points, zones).
- gérer les informations spatiales et les métadonnées sur les tables (comme: système de référence coorinate utilisé).
- convertir des géométries d'un système de coordonnées à un autre
- comparer des géométries et extraire des propriétés (comme: longueur d'arête d'une route ou d'un bâtiment)
- générer de nouvelles géométries à partir d’autres.

Versions

Dernières fonctionnalités	Documentation officielle	Date de sortie
2.3.0	HTML , PDF	2016-09-26
2.2.0	HTML , PDF	2015-10-07
2.1.0	HTML , PDF	2013-08-17
2.0.0	HTML , PDF	2012-04-03
1.5.0		2010-02-04
1.4.0		2009-07-24
1.3.0		2007-08-09
1.2.0		2006-12-08
1.1.0		2005-12-21

Dernières fonctionnalités	Documentation officielle	Date de sortie
1.0.0		2005-04-19

Examples

Installation via le gestionnaire de paquets

Cambre:

Un paquet pacman officiel est disponible. Installez le paquet en tant que root, en utilisant:

```
pacman -S postgis
```

OpenSuse:

Pour utiliser les référentiels openSuse pour les applications géospatiales, activez le référentiel géographique en tant que root:

```
zypper ar http://download.opensuse.org/repositories/Application:/Geo/openSUSE_[RELEASE]/ GEO
```

où *[RELEASE]* est le numéro de version officiel de votre distribution Suse. Après cela, vous pouvez installer postgis avec:

```
zypper install postgis
```

Installation à partir de la source (avec Postgres 9.1 ou supérieur)

Ce guide est explicitement destiné à PostgreSQL 9.1 ou supérieur sur les machines Linux. Il utilise la fonctionnalité d'extensions postgres qui améliorera considérablement l'importation d'extensions à une installation postgres existante. Si vous devez travailler avec une ancienne version de postgres, veuillez vous référer aux documentations officielles .

Résoudre les dépendances

PostGIS est un projet complexe qui comporte un certain nombre de dépendances. Pour procéder à la configuration manuelle et à la procédure de construction, vous devrez résoudre ces dépendances et installer les packages suivants manuellement ou via les gestionnaires de packages.

Exigences minimales

- PostgreSQL 9.1 ou supérieur. Il est important que vous installiez la base de données, y compris les en-têtes de serveur, qui se trouvent généralement dans les paquetages *dev* du gestionnaire de paquets de votre référentiel.
- Le compilateur GNU C gcc .
- GNU make . Pour terminer le processus de construction.

- Proj4 . Une bibliothèque de projection, pour coordonner les transformations.
- GEOS . Une bibliothèque de géométries, qui implémente des descriptions de fonctions et des géométries simples. La version 3.5 ou supérieure est recommandée pour utiliser des fonctions plus récentes telles que *ST_ClipByBox2D* et *ST_Subdivide* .
- GDAL , version 1.9 ou supérieure. Une bibliothèque qui implémente des formats de données abstraits pour les données géospatiales raster et vectorielles.
- LibXML2 , version 2.5 ou supérieure. Un libraray pour travailler avec XML, XSLT et DTD.
- JSON-C , version 0.9 ou supérieure. Une bibliothèque pour créer une sortie au format JSON

Exigences facultatives

- GTK (nécessite GTK + 2.0, 2.8+) pour compiler le shp2pgsql-gui.
- SFCGAL , version 1.1 (ou supérieure) pourrait être utilisé pour fournir des fonctions d'analyse avancée 2D et 3D supplémentaires à PostGIS.
- PCRE . Une bibliothèque pour la correspondance de motifs d'expression régulière à l'aide de la syntaxe Perl 5. Cette bibliothèque est nécessaire si vous souhaitez travailler avec le standardisateur d'adresses .
- CUnit . Un utilitaire de test unitaire, nécessaire pour les tests de régression.
- DocBook (xsltproc) est requis pour générer la documentation.
- DBLatex est requis pour créer la documentation au format PDF.
- ImageMagick est requis pour générer les images utilisées dans la documentation.

Obtenir les sources

Pour obtenir le code source, téléchargez la dernière archive tar:

```
wget http://postgis.net/stuff/postgis-2.3.2dev.tar.gz
tar -xvzf postgis-2.3.2dev.tar.gz
```

ou utilisez le dépôt officiel SVN:

```
svn checkout http://svn.osgeo.org/postgis/trunk/ postgis-2.3.2dev
```

Configuration

Si vous avez obtenu les sources via SVN, vous pouvez préparer le script config avec:

```
./autogen.sh
```

Pour configurer le processus de construction pour votre machine spécifique, exécutez-le dans le dossier du projet:

```
./configure
```

Il existe plusieurs paramètres facultatifs pour l'étape de configuration. Veuillez vous reporter à la documentation officielle pour obtenir des instructions détaillées. Celles-ci sont généralement facultatives et uniquement pour les serveurs utilisant des installations autres que celles par défaut.

Construire

Une fois l'étape de configuration terminée, un fichier Make sera créé. Pour lancer le processus de génération:

```
make
```

La dernière sortie devrait être:

```
"PostGIS was built successfully. Ready to install."
```

Depuis la version 1.4.0, toutes les fonctions ont des commentaires générés à partir de la documentation. Si vous souhaitez installer ces commentaires dans vos bases de données spatiales ultérieurement, exécutez la commande qui requiert docbook.

```
make comments
```

Installation

Installez toutes les extensions avec:

```
make install
```

Les extensions PostGIS sont créées et installées automatiquement si vous utilisez PostgreSQL 9.1 ou une version ultérieure. Vous pouvez installer les extensions nécessaires manuellement si vous avez une configuration différente.

Dans le dossier du projet:

```
cd extensions
cd postgis
make clean
make
make install

cd ..
cd postgis_topology
make clean
make
make install

cd ..
cd postgis_sfcgal
make clean
make
make install

cd ..
cd address_standardizer
make clean
make
make install
make installcheck
```

```
cd ..
cd postgis_tiger_geocoder
make clean
make
make install
make installcheck
```

Si vous souhaitez installer les extensions manuellement sur un autre ordinateur, copiez les fichiers suivants du dossier `extensions` dans le dossier `PostgreSQL/share/extension` de la cible. Pour chaque extension:

```
scp extensions/[EXTENSION]/sql/*.sql user@target:[POSTGIS_PATH]/share/extension
```

où *[EXTENSION]* est l'extension sélectionnée (postgis, postgis_topology, postgis_sfcgal, address_standardizer, postgis_tiger_geocoder) et *[POSTGIS_PATH]* est le chemin d'installation de PostGIS sur votre machine cible.

Vérification de l'installation

Si vous ne disposez pas d'un service de base de données postgres en cours d'exécution, configurez d'abord votre base de données postgres . Connectez-vous à la base de données en utilisant:

```
su postgres -c psql
```

Pour vérifier que les extensions sont accessibles, exécutez les requêtes suivantes dans une session psql:

```
SELECT name, default_version,installed_version FROM pg_available_extensions WHERE name LIKE
'postgis%' or name LIKE 'address%';
```

La sortie devrait ressembler à ceci:

```
             name              | default_version | installed_version
-------------------------------+-----------------+-------------------
 address_standardizer          | 2.3.2dev        | 2.3.2dev
 address_standardizer_data_us  | 2.3.2dev        | 2.3.2dev
 postgis                       | 2.3.2dev        | 2.3.2dev
 postgis_sfcgal                | 2.3.2dev        |
 postgis_tiger_geocoder        | 2.3.2dev        | 2.3.2dev
 postgis_topology              | 2.3.2dev        |

(6 rows)
```

Pour effectuer un test post-installation approfondi, exécutez la commande suivante dans votre dossier de projet:

```
make check
```

Cela se déroulera à travers différents contrôles et tests utilisant la bibliothèque générée sur une

base de données PostgreSQL réelle.

Mise en place d'une base de données géospatiale

Pour créer une nouvelle base de données vide, exécutez-la en tant qu'utilisateur postgres:

```
createdb [yourdatabase]
```

Connectez-vous à la base de données avec une session psql:

```
psql -d [yourdatabase]
```

Dans la session psql-run:

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

pour créer les extensions géospatiales nécessaires. Une fois cela fait, la base de données est une base de données géospatiale et elle est prête à être utilisée.

Un géospatial "Hello World"

Dans cet exemple, nous allons configurer une base de données géospatiale, importer des données provenant de 2 sources différentes et afficher les résultats dans une application appelée QGIS. Ce guide est explicitement écrit pour les machines Linux, si vous opérez sur une autre plate-forme, certaines commandes ou certains chemins risquent de ne pas fonctionner comme prévu.

Afin de visualiser les données importées, nous utiliserons une application appelée QGIS . Si vous ne possédez pas cette application, installez-la d'abord , si vous souhaitez utiliser un autre visualiseur ou une application géo (comme ArcGIS), vous pouvez ignorer l'installation de QGIS.

Nos sources seront les districts de l'Assemblée de l' État de New York et la base de données LION de New York . Veuillez télécharger les fichiers appropriés à partir des emplacements liés. Vous devriez également jeter un coup d'œil à la section Métadonnées du document, car elle vous donne des informations sur le système de référence de coordonnées utilisé par ces fichiers.

Pour commencer, créez un dossier de travail "nycgis", copiez les fichiers téléchargés à l'emplacement et décompressez les archives.

```
mkdir nycgis
cd nycgis
cp ~/Downloads/nyad_16d.zip .
unzip ~/Downloads/nyad_16d.zip
cp ~/Downloads/nylion_16d.zip .
unzip  ~/Downloads/nylion_16d.zip
```

Dans le dossier "nycgis", vous devriez maintenant avoir 2 dossiers: "nyad_16d", "lion" avec plusieurs fichiers.

Lorsque vous travaillez avec des données géographiques, il est essentiel de connaître le système de référence de coordonnées (CRS) de vos données source et de vos données de sortie finales. Dans les sections de métadonnées des emplacements liés ci-dessus (Métadonnées: Districts d'assemblage , Métadonnées: Base de données LION), vous trouverez que le SCR des deux fichiers est EPSG: 2263, un système de coordonnées utilisé pour référencer le nord-est des États-Unis.

Supposons que nous voulons utiliser un CRS différent dans notre base de données. Cela peut avoir différentes raisons, nous pourrions vouloir utiliser une application Web basée sur la base de données par exemple. WGS: 84 (EPSG: 4326) est un CRS commun pour ce type d'application.

Pour convertir les systèmes de coordonnées, nous utilisons un outil appelé `ogr2ogr` qui fait partie du package GDAL. Dans le dossier de travail, nous créons d'abord 2 dossiers représentant les données reprojetées, puis convertissons nos données.

```
mkdir nyad_16d_proj_4326
ogr2ogr -f "ESRI Shapefile" ./nyad_16d_proj_4326/nyad_4326.shp ./nyad_16d/nyad_16d.shp -s_srs
EPSG:2263 -t_srs EPSG:4326

mkdir nylion_16d_proj_4326
ogr2ogr -f "ESRI Shapefile" ./nylion_16d_proj_4326/ny_str_4326.shp
./nylion_16d/lion/lion.gdb/a0000000d.gdbtable -s_srs EPSG:2263 -t_srs EPSG:4326
```

Notez que nous utilisons uniquement le fichier appelé: "a0000000d.gdbtable" de la base de données LION pour nos besoins. La syntaxe de la commande `ogr2ogr` est la suivante:

```
ogr2ogr -f [output-format] [output-file] [input-file] -s_srs [source crs] -t_srs [target crs]
```

Nous avons maintenant 2 fichiers de formes, projetés dans le bon CRS. Pour utiliser les données de notre base de données, nous devons convertir les fichiers de formes en sql-statemens. Pour cela, nous utilisons un outil appelé `shp2pgsql` . Dans le répertoire de travail, exécutez les commandes suivantes:

```
shp2pgsql ./nyad_16d_proj_4326/nyad_4326.shp > nyad_4326.sql
shp2pgsql ./nylion_16d_proj_4326/ny_str_4326.shp > ny_streets_4326.sql
```

Les fichiers `nyad_4326.sql` et `ny_streets_4326.sql` sont maintenant prêts à être utilisés dans postgis. Pour continuer et importer les données, créez une base de données spatialement activée.

```
sudo su - postgres
createdb nycgis
psql -d nycgis
```

Dans la session psql, exécutez:

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

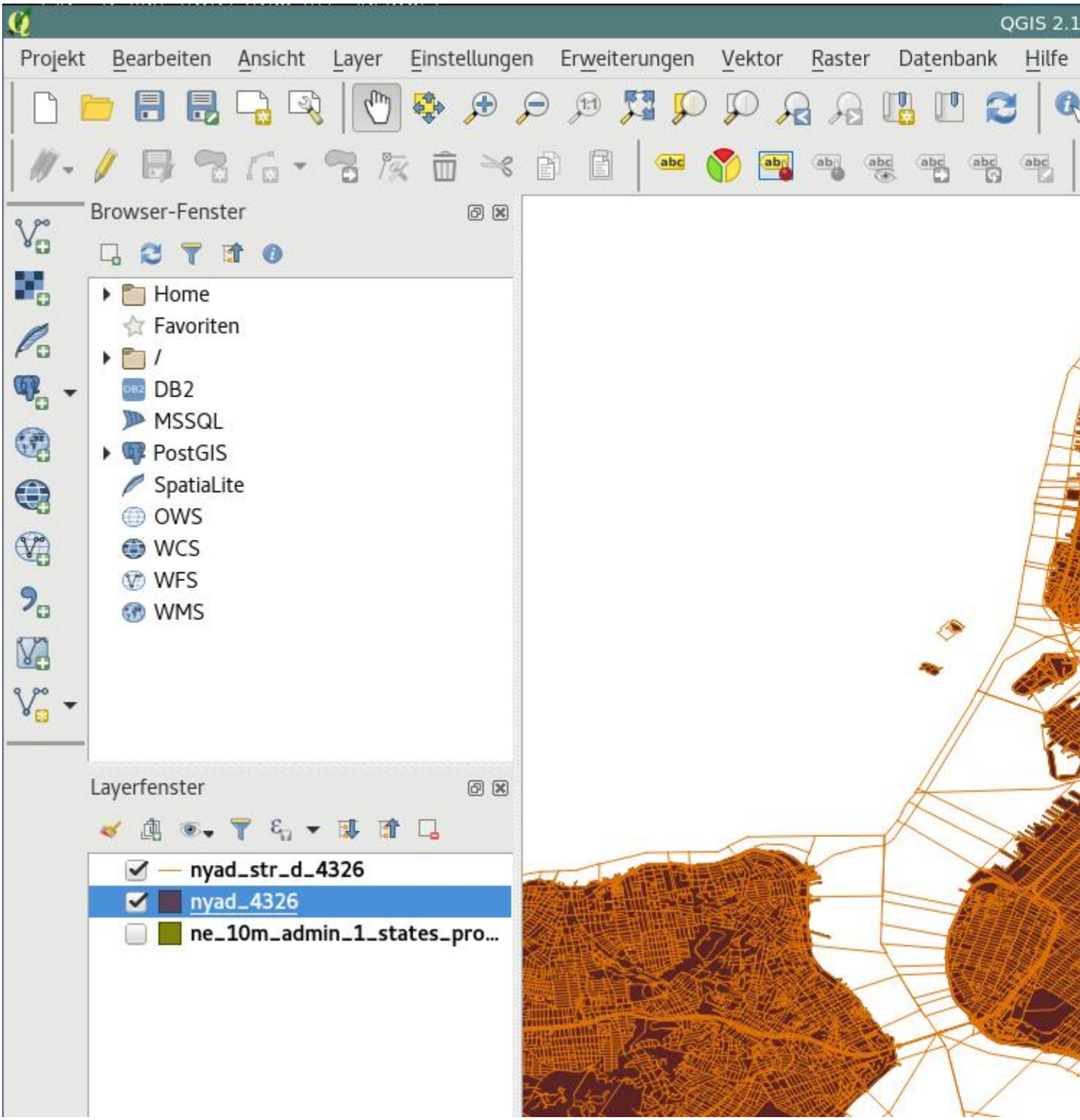
Quittez les sessions psql- et postgres-user avec (CTRL + D). Et importez les fichiers avec:

```
psql -f nyad_4326.sql -d nycgis
psql -f ny_streets_4326.sql -d nycgis
```

La base de données `nycgis` a maintenant 2 tables dans lesquelles les sources reprojetées ont été importées avec succès.

Pour vérifier ceci: ouvrez QGIS

1. utiliser la *couche* > *Ajouter une couche* > *PostGIS-Layer*
2. se connecter à votre base de données
3. sélectionnez vos tables
4. (facultatif) définir le style des nouveaux calques créés.



Et voilà: vous disposez maintenant d'une base de données spatialement active, avec des géodonnées importées et reprogrammées.

Lire Démarrer avec Postgis en ligne: https://riptutorial.com/fr/postgis/topic/7380/demarrer-avec-postgis

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec Postgis	Community, maze-le