



EBook Gratis

APRENDIZAJE

postgis

Free unaffiliated eBook created from
Stack Overflow contributors.

#postgis

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con postgis.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Instalación a través de Package Manager.....	3
Instalación desde Fuente (con Postgres 9.1 o superior).....	3
Configuración de una base de datos geoespacial.....	7
Un "Hola Mundo" geoespacial.....	7
Creditos.....	11

Acerca de

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Capítulo 1: Empezando con postgis

Observaciones

PostGIS es un conjunto de extensiones para la base de datos PostgreSQL. Con PostGIS puede almacenar datos geoespaciales y realizar consultas espaciales en una base de datos de Postgres.

A diferencia de los tipos de datos predeterminados en una base de datos normal de Postgres, los datos espaciales deben manejarse de manera diferente. Las consultas que puede realizar en una base de datos espacial generalmente se definen mediante cuadros delimitadores en 2 o 3 dimensiones. Para almacenar, indexar y manejar estos tipos de datos, postGIS utiliza un concepto llamado [R-Trees](#), que no forma parte de la instalación predeterminada de postgres.

Con una base de datos postGIS, usted puede:

- almacenar datos espaciales
- Realizar consultas espaciales, para recuperar y extraer información (puntos, áreas).
- gestione la información espacial y los metadatos en tablas (como: coordinate-reference-system).
- convertir geometrías de un sistema de coordenadas a otro
- compare geometrías y extraiga propiedades (como: longitud del borde de una carretera o edificio)
- Generar nuevas geometrías a partir de otros.

Versiones

Últimos lanzamientos de características	Documentacion oficial	Fecha de lanzamiento
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

Últimos lanzamientos de características	Documentación oficial	Fecha de lanzamiento
1.0.0		2005-04-19

Examples

Instalación a través de Package Manager

Arco:

Un paquete oficial de pacman está disponible. Instala el paquete como root, usando:

```
pacman -S postgis
```

OpenSuse:

Para utilizar los repositorios de openSuse para aplicaciones geoespaciales, habilite el Geo-repositorio como root:

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

donde *[RELEASE]* es el número de versión oficial de su distribución Suse. Después de esto, puedes instalar postgis con:

```
zypper install postgis
```

Instalación desde Fuente (con Postgres 9.1 o superior)

Esta guía está explícitamente para PostgreSQL 9.1 o superior en máquinas Linux. Utiliza la función de extensiones de postgres que mejorará en gran medida la importación de extensiones a una instalación de postgres existente. Si tiene que trabajar con una versión anterior de Postgres, consulte las [documentaciones oficiales](#) .

Resolver dependencias

PostGIS es un proyecto complejo que tiene varias dependencias. Para continuar con la configuración manual y el procedimiento de compilación, deberá resolver estas dependencias e instalar los siguientes paquetes manualmente o por medio de los administradores de paquetes.

Requerimientos mínimos

- [PostgreSQL](#) 9.1 o superior. Es importante que instale la base de datos, incluidos los encabezados del servidor, que generalmente se encuentran en los paquetes *dev* del administrador de paquetes de su repositorio.
- El compilador GNU C [gcc](#) .
- GNU [hace](#) . Para completar el proceso de construcción.

- [Proj4](#) . Una biblioteca de proyecciones, para transformaciones de coordenadas.
- [GEOS](#) . Una biblioteca de geometría, que implementa descripciones de características y geometrías simples. Se recomienda la versión 3.5 o superior para usar funciones más nuevas como *ST_ClipByBox2D* y *ST_Subdivide* .
- [GDAL](#) , versión 1.9 o superior. Una biblioteca que implementa formatos de datos abstractos para datos geoespaciales basados en ráster y vectores.
- [LibXML2](#) , versión 2.5 o superior. Un bibliotecario para trabajar con XML, XSLT y DTD.
- [JSON-C](#) , versión 0.9 o superior. Una biblioteca para crear resultados en el formato JSON.

Requisitos opcionales

- [GTK](#) (requiere GTK + 2.0, 2.8+) para compilar shp2pgsql-gui.
- [SFCGAL](#) , versión 1.1 (o superior) podría usarse para proporcionar funciones de análisis avanzado 2D y 3D adicionales a PostGIS.
- [PCRE](#) . Una biblioteca para la coincidencia de patrones de expresiones regulares utilizando la sintaxis de Perl 5. Esta biblioteca es necesaria si desea trabajar con [Address Standardizer](#) .
- [Cunit](#) . Una utilidad de prueba unitaria, necesaria para pruebas de regresión.
- Se requiere [DocBook](#) (xsltproc) para crear la documentación.
- [DBLatex](#) es necesario para construir la documentación en formato PDF.
- Se requiere que [ImageMagick](#) genere las imágenes utilizadas en la documentación.

Obtener las fuentes

Para obtener el código fuente, descargue el último archivo tar:

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

o use el repositorio SVN oficial:

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

Configuración

Si obtuvo las fuentes a través de SVN, puede preparar el script de configuración con:

```
./autogen.sh
```

Para configurar el proceso de compilación para su máquina específica, ejecute en la carpeta del proyecto:

```
./configure
```

Hay varios parámetros opcionales para el paso de configuración. Consulte la [documentación oficial](#) para obtener instrucciones detalladas, esto suele ser opcional y solo para servidores que utilizan instalaciones no predeterminadas.

Construir

Una vez que el paso de configuración haya finalizado con éxito, se creará un makefile. Para iniciar la ejecución del proceso de compilación:

```
make
```

La última salida debe ser:

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

A partir de la versión 1.4.0, todas las funciones tienen comentarios generados a partir de la documentación. Si desea instalar estos comentarios en sus bases de datos espaciales más adelante, ejecute el comando que requiere docbook.

```
make comments
```

Instalación

Instala todas las extensiones con:

```
make install
```

Las extensiones de PostGIS se crean e instalan automáticamente si está utilizando PostgreSQL 9.1 o superior. Puede instalar las extensiones necesarias manualmente, si tiene una configuración diferente.

En la carpeta del proyecto:

```
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 desea instalar las extensiones manualmente en una máquina diferente, copie los siguientes archivos de la carpeta de `extensions` en la carpeta `PostgreSQL/share/extension` del destino. Para cada extensión:

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

donde `[EXTENSIÓN]` es la extensión seleccionada (`postgis`, `postgis_topology`, `postgis_sfcgal`, `address_standardizer`, `postgis_tiger_geocoder`) y `[POSTGIS_PATH]` es la ruta de instalación de PostGIS en su máquina objetivo.

Verificación de la instalación

Si no tiene un servicio de base de datos postgres en ejecución, [configure primero su base de datos postgres](#) . Conéctese a la base de datos, utilizando:

```
su postgres -c psql
```

Para verificar que las extensiones sean accesibles, ejecute las siguientes consultas en una sesión `psql`:

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

La salida debería verse así:

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)

Para realizar una prueba posterior a la instalación en profundidad, ejecute el siguiente comando en su carpeta de proyectos:

```
make check
```

Esto se ejecutará a través de varias comprobaciones y pruebas utilizando la biblioteca generada contra una base de datos PostgreSQL real.

Configuración de una base de datos geoespacial

Para crear una nueva base de datos vacía, ejecute como postgres-usuario:

```
createdb [yourdatabase]
```

Conéctese a la base de datos con una sesión psql:

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

En la ejecución de la sesión psql:

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

Para crear las extensiones geoespaciales necesarias. Una vez hecho esto, la base de datos es una base de datos geoespacial y está lista para usar.

Un "Hola Mundo" geoespacial

En este ejemplo, configuraremos una base de datos geoespacial, importaremos datos de 2 fuentes diferentes y veremos los resultados en una aplicación llamada QGIS. Esta guía está escrita explícitamente para máquinas linux. Si opera en otra plataforma, es posible que algunos comandos o rutas no funcionen como se esperaba.

Para ver los datos importados usaremos una aplicación llamada [QGIS](#) . Si no tiene esta aplicación, [instálela primero](#) , si desea usar otro visor o aplicación geográfica (como ArcGIS), puede omitir la instalación de QGIS.

Nuestras fuentes serán los [Distritos de la Asamblea del Estado de Nueva York](#) y la [Base de datos de mapas de la calle LION de la Ciudad de Nueva York](#) . Por favor descargue los archivos apropiados de las ubicaciones vinculadas. También debe echar un vistazo a la sección Metadatos del material, ya que le brinda información sobre qué sistema de referencia de coordenadas utilizan estos archivos.

Para comenzar, cree una carpeta de trabajo "nycgis", copie los archivos descargados en la ubicación y descomprima los archivos.

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

En la carpeta "nycgis" ahora debería tener 2 carpetas: "nyad_16d", "lion" con varios archivos.

Cuando se trabaja con datos geográficos, es de vital importancia conocer el sistema de referencia de coordenadas (CRS) de sus datos de origen y de sus datos de salida finales. En las secciones

de Metadatos de las ubicaciones vinculadas ([Metadatos: Distritos de ensamblaje](#) , [Metadatos: Base de datos LION](#)), encontrará que el CRS para ambos archivos es EPSG: 2263, un sistema de coordenadas utilizado para hacer referencia al noreste de los Estados Unidos.

Asumamos que queremos usar un CRS diferente en nuestra base de datos. Esto puede tener diferentes motivos, por ejemplo, podríamos querer trabajar con una aplicación geo basada en web en la base de datos. Un CRS común para este tipo de aplicación es WGS: 84 (EPSG: 4326).

Para convertir los sistemas de coordenadas usamos una herramienta llamada `ogr2ogr` que forma parte del paquete GDAL. En la carpeta de trabajo, primero creamos 2 carpetas que representan los datos reprojectados, y luego convertimos nuestros datos.

```
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
```

Tenga en cuenta que solo usamos el archivo llamado "a0000000d.gdbtable" de la Base de Datos LION para nuestros propósitos. La sintaxis del `ogr2ogr` es la siguiente:

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

Ahora tenemos 2 shapefiles, proyectados en el CRS correcto. Para utilizar los datos en nuestra base de datos, debemos convertir los shapefiles a sql-statements. Para esto utilizamos una herramienta llamada `shp2pgsql`. En el directorio de trabajo ejecute los siguientes comandos:

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

Los archivos `nyad_4326.sql` y `ny_streets_4326.sql` ahora están listos para usar en postgres. Para continuar, e importar los datos, cree una base de datos habilitada espacialmente.

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

En la sesión de psql, ejecute:

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

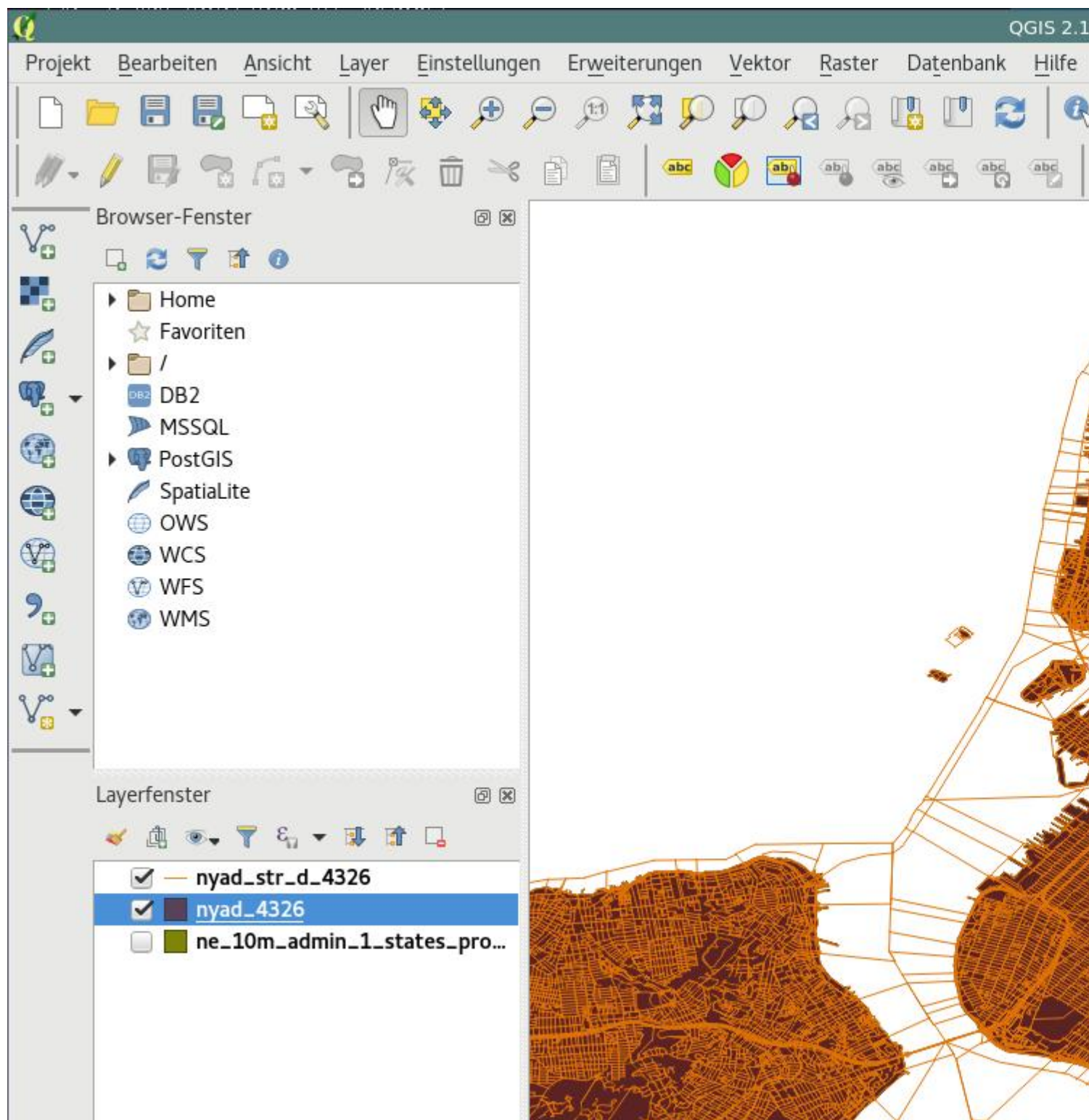
Salga de las sesiones de usuario psql y postgres con (CTRL + D). E importa los archivos con:

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

La base de datos `nycgis` tiene ahora 2 tablas en las cuales las fuentes reproyectadas fueron importadas exitosamente.

Para verificar esto: abre QGIS

1. usar *Capa > Agregar Capa > Capa PostGIS*
2. conectarse a su base de datos
3. selecciona tus mesas
4. (opcional) establecer el estilo de las capas recién creadas.



Et voilà: ahora tiene una base de datos habilitada espacialmente, con geodatos importados y reproyectados.

Lea **Empezando con postgis en línea**: <https://riptutorial.com/es/postgis/topic/7380/empezando-con-postgis>

Creditos

S. No	Capítulos	Contributors
1	Empezando con postgis	Community , maze-le