

Пространственные базы данных в PostgreSQL

Муравьёв С.К.

Национальный исследовательский ядерный университет «МИФИ»
Кафедра №36 «Информационные системы и технологии»

December 17, 2013

Пространственные базы данных

Пространственная база данных — база данных, оптимизированная для хранения и выполнения запросов к данным о пространственных объектах, представленных некоторыми абстракциями: точка, линия, полигон.

Пространственные БД позволяют выполнять аналитические запросы для анализа пространственно-логических отношений объектов (пересекается, касается, содержится в, содержит и пр.)

Некоммерческая организация Консорциум «The Open Geospatial Consortium, Inc» (OGC) разработала и утвердила стандарт OpenGIS, определяющий стандартные типы объектов ГИС, функции для манипуляции ими, и набор таблиц метаданных.

PostGIS

PostGIS - расширение PostgreSQL предназначенное для хранения в базе географических данных в формате OpenGIS.

PostGIS включает поддержку пространственных индексов R-Tree/GiST и функции обработки геоданных.

Дополнительная информация:

- [Официальная страница OGC](#)
- [Официальная страница PostGIS](#)
- [Руководство по PostGIS](#)
- [Установка PostGIS](#)

Установка PostGIS

Установка репозитория ELGIS:

```
$ su
# rpm -Uvh http://elgis.argeo.org/repos/6/
    elgis-release-6-6_0.noarch.rpm
```

Установка репозитория EPEL:

```
# rpm -Uvh http://download.fedoraproject.org/pub/
    epel/6/i386/epel-release-6-8.noarch.rpm
```

Установка PostGIS:

```
# yum install --nogpgcheck postgis2_93
```

Начало работы с PostGIS

Создание пространственной базы данных:

```
# su - postgres
$ psql
postgres=# create database spatial_db;
postgres=# \c spatial_db
spatial_db=# CREATE EXTENSION postgis;
spatial_db=# SELECT PostGIS_version();
2.1 USE_GEOS=1 USE_PROJ=1 USE_STATS=1
```

Создание таблицы с пространственным столбцом:

```
spatial_db=# CREATE TABLE geotable (
  id SERIAL PRIMARY KEY,
  geom GEOMETRY(Point, 26910),
  name VARCHAR(128)
);
```

OpenGIS WKB и WKT

Спецификация OpenGIS определяет два стандартных способа определения пространственных объектов: в форме Well-Known Text (WKT) и в форме Well-Known Binary (WKB).

WKT и WKB включают информацию о типе объекта и координаты, составляющие объект.

Примеры:

```
POINT(0 0)
LINESTRING(0 0,1 1,1 2)
POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
MULTIPOINT(0 0,1 2)
MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 1)),
              ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))
```

Ввод/вывод геоданных в форматах WKB и WKT

Ввод/вывод геоданных в форматах WKB и WKT доступен с использованием следующих интерфейсов:

```
bytea WKB = ST_AsBinary(geometry);  
text WKT = ST_AsText(geometry);  
geometry = ST_GeomFromWKB(bytea WKB, SRID);  
geometry = ST_GeometryFromText(text WKT, SRID);
```

Пример запроса для создания и вставки пространственного объекта OGC:

```
INSERT INTO geotable (id, geom, name)  
VALUES (1, ST_GeomFromText('POINT(-126.4 45.32)', 26910),  
       'A Place');
```

PostGIS EWKB, EWKT

PostGIS расширяет форматы текущего набора OGC, который поддерживает только 2D объекты.

Примеры:

```
POINT(0 0 0) -- XYZ
SRID=32632;POINT(0 0) -- XY with SRID
POINTM(0 0 0) -- XM
POINT(0 0 0 0) -- XYZM
SRID=4326;MULTIPOINTM(0 0 0,1 2 1) -- XYM with SRID
MULTILINESTRING((0 0 0,1 1 0,1 2 1),
                 (2 3 1,3 2 1,5 4 1))
GEOMETRYCOLLECTIONM(POINTM(2 3 9),
                     LINESTRINGM(2 3 4, 3 4 5))
```


Ввод/вывод геоданных в форматах EWKB и EWKT

Ввод/вывод геоданных в форматах EWKB и EWKT доступен с использованием следующих интерфейсов:

```
bytea EWKB = ST_AsEWKB(geometry);  
text EWKT = ST_AsEWKT(geometry);  
geometry = ST_GeomFromEWKB(bytea EWKB);  
geometry = ST_GeomFromEWKT(text EWKT);
```

Пример запроса для создания и вставки пространственного объекта OGC:

```
INSERT INTO geotable (id, geom, name)  
VALUES (1, ST_GeomFromEWKT('SRID=26910;  
POINTM(-126.4 45.32 15)'), 'A Place');
```

Системы координат

Описания систем координат, используемых в пространственной базе данных, хранятся в таблице `spatial_ref_sys`.

```
spatial_db=# \d spatial_ref_sys
```

```
Table "public.ddspatial_ref_sys"
```

Column	Type	Modifiers
-----+-----+-----		
srid	integer	not null
auth_name	character varying(256)	
auth_srid	integer	
srttext	character varying(2048)	
proj4text	character varying(2048)	

```
Indexes:
```

```
"spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)
```

```
Check constraints:
```

```
"spatial_ref_sys_srid_check" CHECK  
(srid > 0 AND srid <= 998999)
```

Системы координат

- `srid` - Уникальный идентификатор системы координат (Spatial Referencing System, SRS) в пределах базы данных.
- `auth_name` - Название стандарта или стандартизирующего организации, являющейся источником информации о данной системе координат. Например, правильным значением `auth_name` будет "EPSG".
- `auth_srid` - Идентификатор системы координат, так как он определяется организацией указанной в `auth_name`. В случае EPSG, это должен быть код проекции EPSG.
- `srtext` - WKT представление системы координат.
- `proj4text` - Определение координат Proj4 для данного `srid`.

Загрузка геоданных в формате ESRI Shapefile

Утилита shp2pgsql преобразует файлы в формате ESRI Shape в SQL, в вид пригодный для вставки в PostGIS/PostgreSQL.

Вызов утилиты shp2pgsql:

```
shp2pgsql -I -s <SRID> <PATH/TO/SHAPEFILE>  
          <SCHEMA>.<DBTABLE> | psql -d <DATABASE>
```

Установка osm2pgsql

Утилита osm2pgsql позволяет загружать в PostGIS/PostgreSQL файлы в формате OpenStreetMap.

Копирование исходных кодов и сборка osm2pgsql:

```
# yum install -y --nogpgcheck libtool postgresql93-devel
  bzip2-devel geos-devel proj-devel protobuf-c-devel
  lua-devel
# su - postgres
$ git clone
  https://github.com/openstreetmap/osm2pgsql.git ~/git
$ cd ~/git/osm2pgsql/
$ ./autogen.sh
$ ./configure
$ make
# su
# make install
```

Загрузка геоданных в формате OpenStreetMap

Загрузка геоданных с помощью osm2pgsql:

```
$ osm2pgsql -U postgres -W -H 127.0.0.1 -s -c  
-S ~/git/osm2pgsql/default.style map.osm -d spatial_db
```

Проверка загруженных данных:

```
spatial_db=# SELECT f_table_name, srid, type  
              FROM geometry_columns;
```

f_table_name	srid	type
mytable	26910	POINT
planet_osm_point	900913	POINT
planet_osm_roads	900913	LINESTRING
planet_osm_polygon	900913	GEOMETRY
planet_osm_line	900913	LINESTRING

(5 rows)

```
spatial_db=# SELECT name, ST_AsText(way) AS geom FROM  
              planet_osm_polygon WHERE osm_id=-2784645;
```

Построение индексов

Индексы делают возможным использование пространственной базы данных для больших наборов данных. Без индексации, любой поиск приводил бы к "последовательному сканированию" каждой записи в базе данных. Индексация организует данные в поисковое дерево, по которому можно быстро перемещаться, чтобы быстро найти конкретную запись.

Индексы GiST (Generalized Search Trees - обобщенные деревья поиска) разделяют данные на "объекты по одну сторону" ("things to one side"), "пересекающиеся объекты" ("things which overlap"), "объекты внутри" ("things which are inside") и могут быть использованы для многих типов данных, включая данные ГИС.

Синтаксис запроса для создания GiST-индекса:

```
CREATE INDEX indexname ON tablename USING GIST (geom);
```

Использование индексов

Временное отключение индекса:

```
spatial_db=# UPDATE pg_index SET indisvalid = false  
WHERE indexrelid = 'planet_osm_polygon_index'::regclass;
```

Поиск объектов в радиусе 100 метров от заданной точки:

```
spatial_db=# \timing  
spatial_db=# SELECT osm_id, name FROM planet_osm_polygon  
WHERE ST_DWithin(way, 'SRID=900913;  
POINT(4192794 7489017)', 100.0);  
...  
Time: 120.523 ms
```

Включение индекса:

```
spatial_db=# UPDATE pg_index SET indisvalid = true  
WHERE indexrelid = 'planet_osm_polygon_index'::regclass;
```

Время поиска объектов с использованием индекса:

```
Time: 32.376 ms
```


Функции OpenGIS

Стандарт OpenGIS определяет большое количество функций для работы с геоданными:

- Функции управления
- Функции геометрической связи
- Функции обработки геометрии
- Геометрические способы доступа
- Геометрические конструкторы
- Функции измерения
- Геометрический вывод
- Геометрические редакторы
- Линейные ссылки
- и другие

Подробнее этих функций можно найти в официальном [руководстве](#)

Функции управления

Тип	Описание
AddGeometryColumn	Добавляет столбец геометрии в существующую таблицу
DropGeometryColumn	Удаляет столбец геометрии указанной таблицы.
ST_SetSRID	Устанавливает SRID для геометрии в определенное целочисленное значение.

- Создать новую таблицу со столбцом для хранения геометрии.
- Создать GiST index для таблицы
- Вставить в таблицу точечные, линейные и площадные объекты.
- Определить расстояние между объектами.
- Определить длину линейных объектов.
- Определить площадь площадных объектов.
- Определить наличие пересечения между объектами.
- Построить объединение/пересечение нескольких объектов.
- Найти 5 ближайших станций метро от МИФИ