

Место Postgres/PostGIS в экосистеме открытого ПО

Дмитрий Барышников

e-mail: dmitry.baryshnikov@nextgis.ru, info@nextgis.ru, web: www.nextgis.ru

План

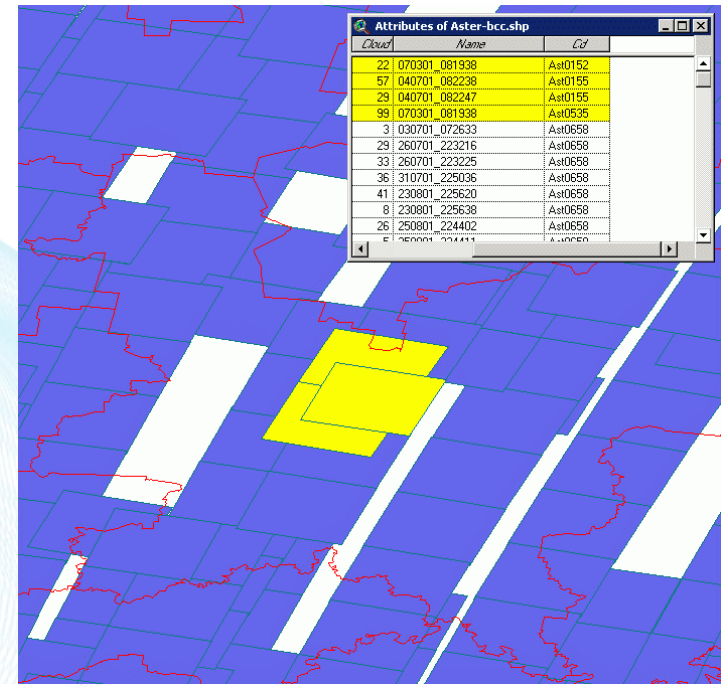
1. Что такое PostGIS
2. Какие функции
3. Лицензия
4. Примеры использования
5. Как с этим всем работают из софта
6. Сообщество.
7. Стандарты и протоколы, совместимость.

Понятие ГИС

геоинформационная система; ГИС:

Информационная система,
оперирующая пространственными
данными. [ГОСТ Р 52438-2005]

Одной из ключевых особенностей
геоданных является единство
геометрической и атрибутивной
информации.



Что такое PostGIS

- **PostGIS** это расширение СУБД PostgreSQL предназначенное для хранения и манипулирования в базе географическими данными.
- Текущая версия **2.1.5** – релиз от 18 декабря прошлого года

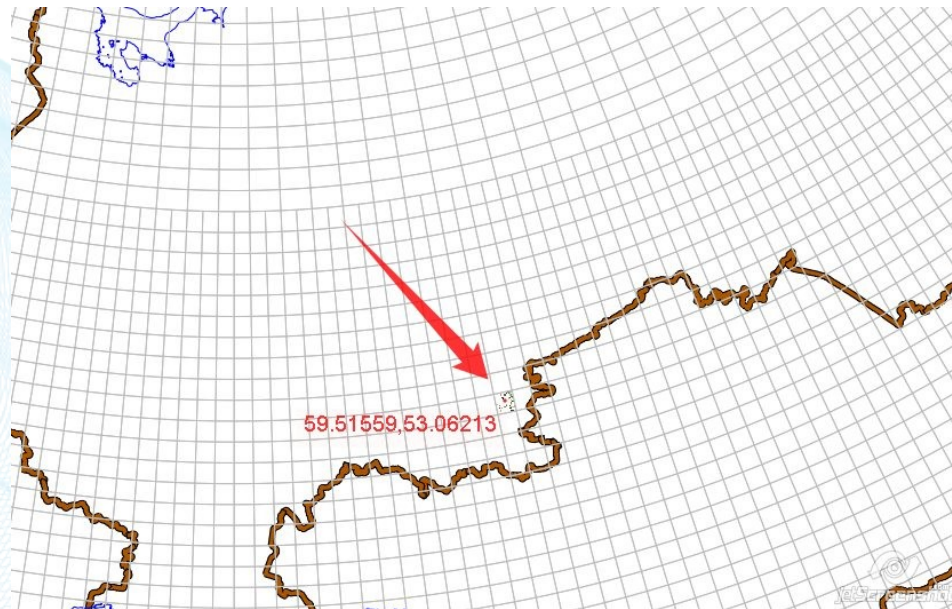
Система координат

- X: 37,6177777778 • X: 4187591,867
- Y: 55,7516666667 • Y: 7509137,51517



Система координат

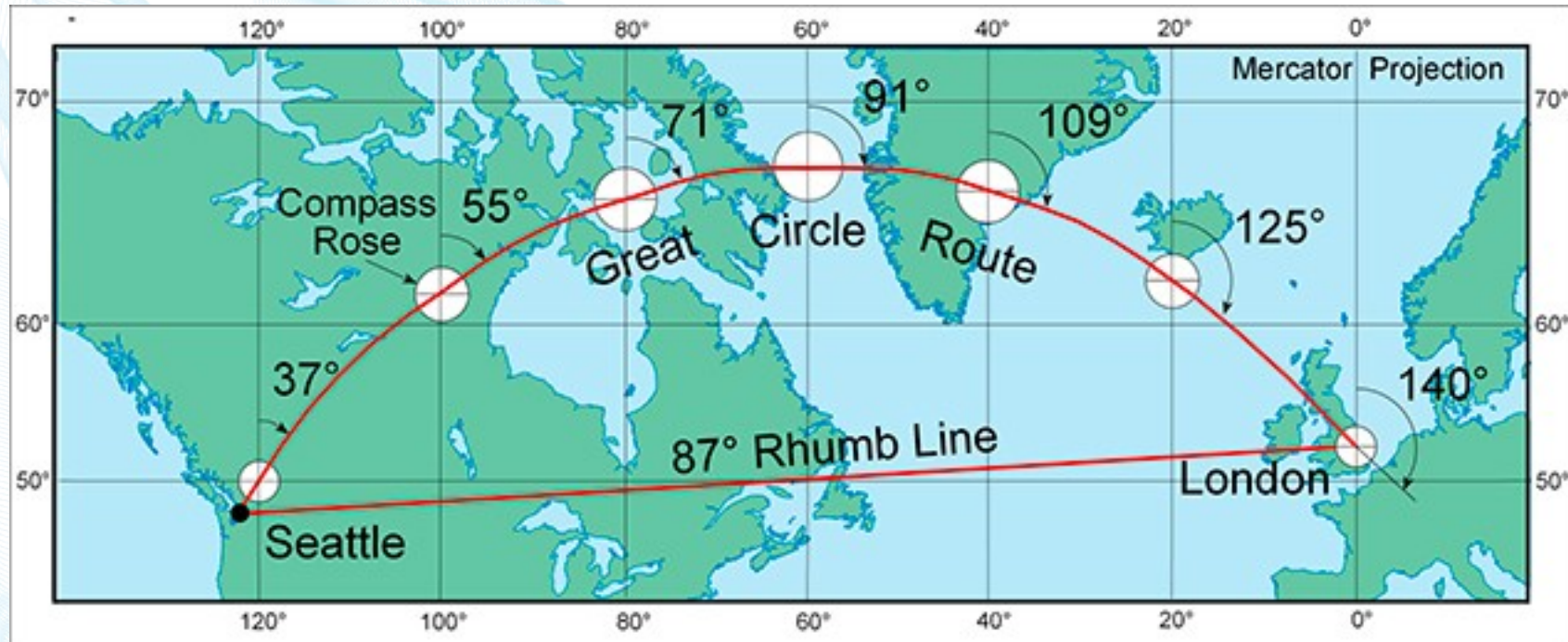
Система координат - это способ определять положение точки в пространстве относительно выбранных осей



Отсканированное изображение,
каждый элемент имеет X,Y

Географически привязанное изображение, каждый
элемент имеет широту, долготу, высоту

Искажение длин и масштабов



Функции PostGIS - геометрии

- Функции управления геометриями (создание и удаление полей с географическими данными, запросы версии расширения и библиотек, скриптов)
- Конструкторы геометрий.
- Функции доступа к геометриям и их свойствам.
- Функции редактирования геометрий.
- Функции форматированного вывода геометрий.
- Операторы над охватами (2D и 3D bbox).

Функции PostGIS - геометрии

- Пространственные отношения и измерения (расстояние на сфере и плоскости, проверка пересечения, примыкания, площадь, периметр, угол и др.)
- Расширенные 2D и 3D функции (площади 3D геометрий, пересечение в 3D пространстве и др.).
- Операции над геометриями (буфер, пересечение, преобразование типов, упрощение, триангуляция и др.)
- Работа с линейной системой координат
- Дополнительные функции работы с геометриями

Функции PostGIS - растры

- Функции управления растрами (создание и удаление свойств растра, отметка/снятие отметки о свойстве поля таблицы - является пирамидным слоем растра или нет, запросы версии библиотек, скриптов)
- Конструкторы растров.
- Функции доступа к растрам и их свойствам.
- Функции доступа к каналам растра и их свойствам.
- Функции работы с пикселями растра
- Функции редактирования растра (определение СК, поворот, масштабирование, сдвиг и др.)

Функции PostGIS - растры

- Функции редактирования каналов растра (установка и запрос значения «нет данных»).
- Статистика растра (гистограмма, суммарная статистика)
- Функции вывода растров (например в JPEG, PNG, TIFF)
- Обработка растров (отмывка рельефа, освещенность, векторизация, обрезка и др.)
- Операции над охватом растров (пересекаются, включают и др.)
- Пространственные связи растров и каналов растров (включает, перекрывается, пересекает, примыкает и др.)

Функции PostGIS — топология и экстра функции

- Общие операции топологии (создание/удаления полей таблицы для поддержки топологии геометрий, вывод обобщенной информации по полю топологии, валидация)
- Операции ввода и модификации топологических правил
- Нестандартные операции работы с топологией (упрощение, поиск ребер и др.)
- Операции работы с топогеометриями
- Геокодер для работы с системой адресации США (TIGER)

Часто используемые функции

- регистрация таблицы как слоя геоданных **AddGeometryColumn**
- создание геометрии **ST_MakePoint**, **ST_MakePolygon**, **ST_MakeLine**
- вывод геометрии в виде текстового представления **ST_AsText**
- пространственные выборки **ST_Buffer** и **ST_Intersect**
- перепроецирование **ST_Transform**
- получение центраида **ST_Centroid** и охвата **ST_Envelope**
- преобразование в мультигеомтерию **ST_Multi**
- получение длины **ST_Length**
- формирование оболочки **ST_ConvexHull**
- упрощение геометрии **ST_Simplify**

Часто используемые функции

AddGeometryColumn

Создание таблицы с пространственными данными выполняется в два шага:

- создание обычной «непространственной» таблицы

Например: **CREATE TABLE ROADS_GEOM (ID int4, NAME varchar(25))**

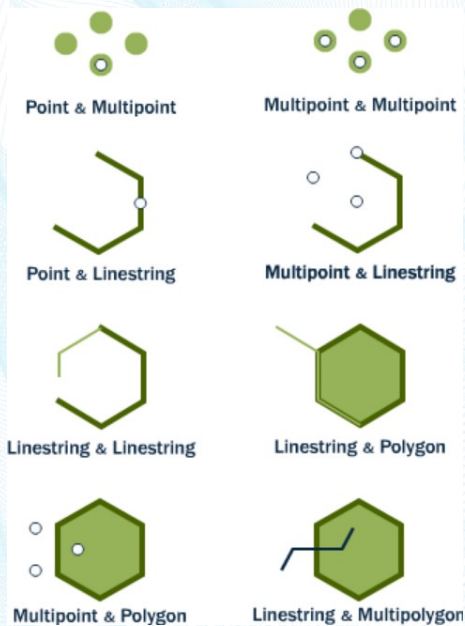
- регистрация таблицы как пространственной (добавление пространственного столбца с помощью функции **AddGeometryColumn**).

Например: **SELECT AddGeometryColumn('public', 'roads_geom', 'geom', 423, 'LINESTRING', 2)**

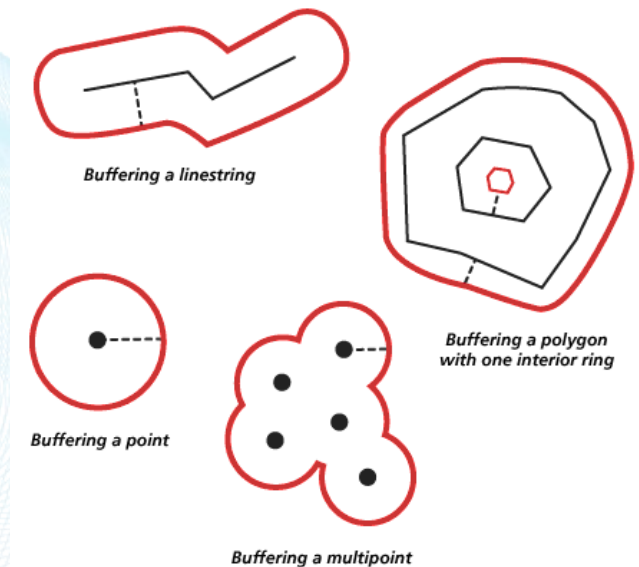
Часто используемые функции

ST_Intersects и ST_Buffer

boolean **ST_Intersects**(geometry
geomA , geometry geomB)



geometry **ST_Buffer**(geometry g1,
float radius_of_buffer, integer
num_seg_quarter_circle)



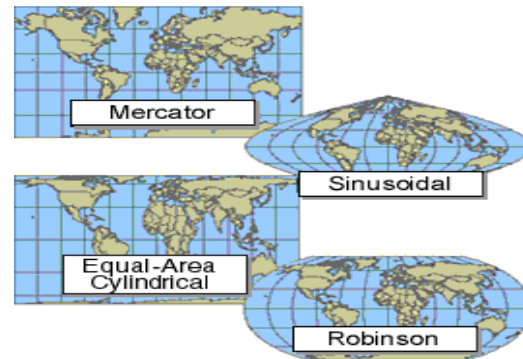
Часто используемые функции

ST_AsText, ST_Transform, ST_Centroid и ST_Envelope

text **ST_AsText**(geometry g1)

Пример вывода функции:
POLYGON(0 0,0 1,1 1,0 0 0)

geometry **ST_Transform**
(geometry g1,
integer srid)



geometry **ST_Centroid**(geometry g1)



geometry **ST_Envelope** (geometry
g1)



Часто используемые функции

ST_Multi, ST_Length, ST_ConvexHull и ST_Simplify

geometry **ST_Multi**(geometry g1)

Пример вывода функции:

```
MULTIPOLYGON(((743238 2967416,743238  
2967450,743265 2967450,743265.625  
2967416, 743238 2967416)))
```

float **ST_Length**(geometry
a_2dlinestring)

Пример вывода функции:

```
st_length: 34309.4563576191
```

geometry **ST_ConvexHull**(geometry
geomA)



geometry **ST_Simplify**(geometry
geomA, float tolerance)



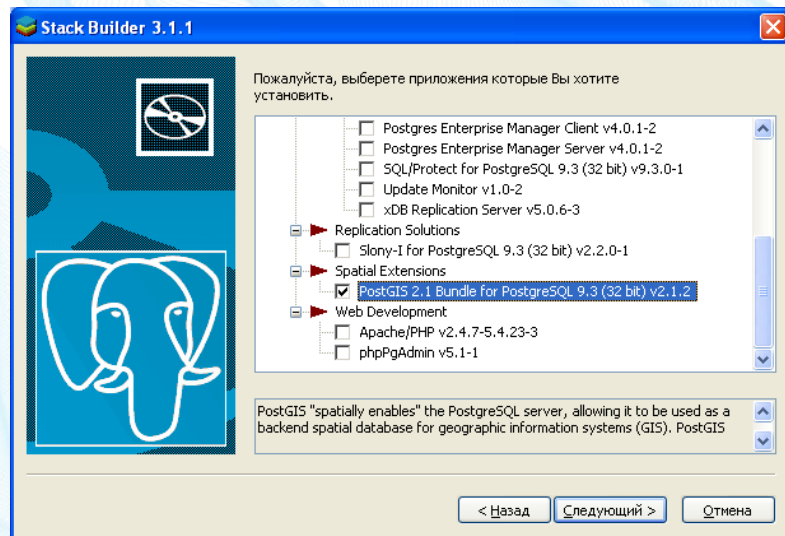
Лицензия

PostGIS распространяется под лицензией **GPL v2**, а также использует следующие библиотеки с открытым исходным кодом:

- GDAL (лицензия MIT совместимая)
- GEOS (лицензия LGPL)
- Proj.4 (лицензия MIT)

Примеры использования установка

Windows



Red Hat / Centos

> `sudo yum install postgis2_93`

Ubuntu/Debian

> `sudo apt-get install -y postgis
postgresql-9.3-postgis-2.1`

CREATE EXTENSION postgis;

Примеры использования пересчет координат

The screenshot shows a PostgreSQL query editor window titled "Query - geo_db из postgres@localhost:5432 *". The window has two tabs: "Редактор SQL" (active) and "Графический конструктор запросов". Below the tabs is a "Предыдущие запросы" (Previous queries) section with a dropdown menu and buttons "Удалить" (Delete) and "Удалить всё" (Delete all). The main SQL editor contains the following query:

```
1 SELECT ST_AsText(ST_Transform(ST_SetSRID(ST_MakePoint(37.617777778, 55.751666667), 4326), 3857)) AS geom;
```

Below the editor is the "Панель вывода" (Output panel) with tabs: "Вывод данных" (Data output), "Построить план выполнения" (Build execution plan), "Сообщения" (Messages), and "История" (History). The "Вывод данных" tab is active, showing a table with the following data:

	geom text
1	POINT(4187591.86699922 7509137.51517409)

At the bottom of the window, the status bar shows "OK.", "Unix Строка 1, Колонка 107, С...", "1 строка. 164 ms".

Примеры использования пространственная выборка

The screenshot shows a PostgreSQL SQL Editor window. The title bar includes menu items: Файл, Правка, Запрос, Избранное, Макрос, Вид, ?. The toolbar contains icons for file operations, editing, and execution. The connection dropdown shows 'geo_db из postgres@localhost:5432'. The 'Редактор SQL' tab is active, displaying the following SQL query:

```
1 SELECT NAME, ST_AsText(wkb_geometry) FROM thematic.productions WHERE ST_Intersects(wkb_geometry, (SELECT wkb_geometry FROM thematic.regions WHERE NAME LIKE 'Московская область')::geometry);
```

Below the query editor is the 'Панель вывода' (Output Panel). It has tabs for 'Вывод данных' (Data Output), 'Построить план выполнения' (Build Execution Plan), 'Сообщения' (Messages), and 'История' (History). The 'Вывод данных' tab is selected, showing a table with 12 rows and 2 columns: 'name' (character varying(243)) and 'st_astext' (text). The data represents road segments intersecting the Moscow region.

	name character varying(243)	st_astext text
1	«Урал» Москва - Рязань - Пенза - Самара - Уфа - Челябинск	POINT(37.9733105620105 55.6124400290512)
2	«Щелковское шоссе» Москва - Щелково - автомобильная дорога А-107 «Московское малое к	POINT(38.0798919030113 55.9033063379808)
3	«Крым» Москва - Тула - Орел - Курск - Белгород - граница с Украиной	POINT(37.6140364616493 55.430425692221)
4	МБК Горьковско - Егорьевский перегон	POINT(38.8792620453375 55.5608596859336)
5	«Крым» Москва - Тула - Орел - Курск - Белгород - граница с Украиной	POINT(37.566275001277 55.1299678867583)
6	«Урал» Москва - Рязань - Пенза - Самара - Уфа - Челябинск	POINT(38.7138988814187 55.1495778872977)
7	«Урал» Москва - Рязань - Пенза - Самара - Уфа - Челябинск	POINT(39.1498299419795 54.8932660173484)
8	МБК Горьковско - Егорьевский перегон	POINT(38.9641928767528 55.8484791783794)
9	«Урал» Москва - Рязань - Пенза - Самара - Уфа - Челябинск	POINT(38.4297069018775 55.3459449805862)
10	«Россия» Москва - Тверь - Великий Новгород - Санкт-Петербург	POINT(37.1316356002288 56.0722553130764)
11	«Россия» Москва - Тверь - Великий Новгород - Санкт-Петербург	POINT(36.7035666152897 56.3509286065765)
12	«Урал» Москва - Рязань - Пенза - Самара - Уфа - Челябинск	POINT(37.9307385819237 55.6337261395194)

At the bottom of the window, the status bar shows 'OK.', 'Unix Строка 1, Колонка 190, С...', '15 строк.', and '908 ms'.

Примеры использования поиск ближайших геометрий

Панель SQL

```
5 CREATE OR REPLACE FUNCTION ng_getclosepointonroad(IN NAME, IN DOUBLE PRECISION, IN DOUBLE PRECISION)
6 RETURNS TABLE(geometry TEXT, "position" DOUBLE PRECISION, distance DOUBLE PRECISION) AS
7 $BODY$
8 DECLARE ret RECORD;
9 DECLARE pt geometry := ST_SetSRID(ST_MakePoint($2, $3),4326);
10 DECLARE ptout geometry;
11 DECLARE dist DOUBLE PRECISION;
12 DECLARE len DOUBLE PRECISION;
13 DECLARE fraction DOUBLE PRECISION;
14 BEGIN
15 --get closer part, guid and scale
16 EXECUTE 'SELECT wkb_geometry,beg,scale FROM ' || $1 || ' ORDER BY ST_Distance(wkb_geometry, ST_SetSRID(ST_MakePoint(' || $2 || ', ' || $3 || '),4326)) LIMIT 1;' INTO ret;
17
18 --get distance
19 fraction := ST_Line_Locate_Point(ret.wkb_geometry, pt);
20 ptout := ST_Line_Interpolate_Point(ret.wkb_geometry, fraction);
21 len := ST_Distance_Sphere(ptout, pt);
22 SELECT ret.beg + ST_Length(ret.wkb_geometry) * fraction / ret.scale INTO dist;
23
24 RETURN QUERY
25 SELECT ST_AsText(ptout),dist,len;
26
27 END;
28 $BODY$
29 LANGUAGE plpgsql STABLE
30 COST 100
31 ROWS 1000;
32
33 ALTER FUNCTION ng_getclosepointonroad(NAME, DOUBLE PRECISION, DOUBLE PRECISION)
34 OWNER TO postgres;
35
```

Query - geo_db из postgres@localhost:5432 *

Редактор SQL | Графический конструктор запросов

Предыдущие запросы

1 SELECT * FROM ng_getclosepointonroad('thematic.parts', 37, 55);

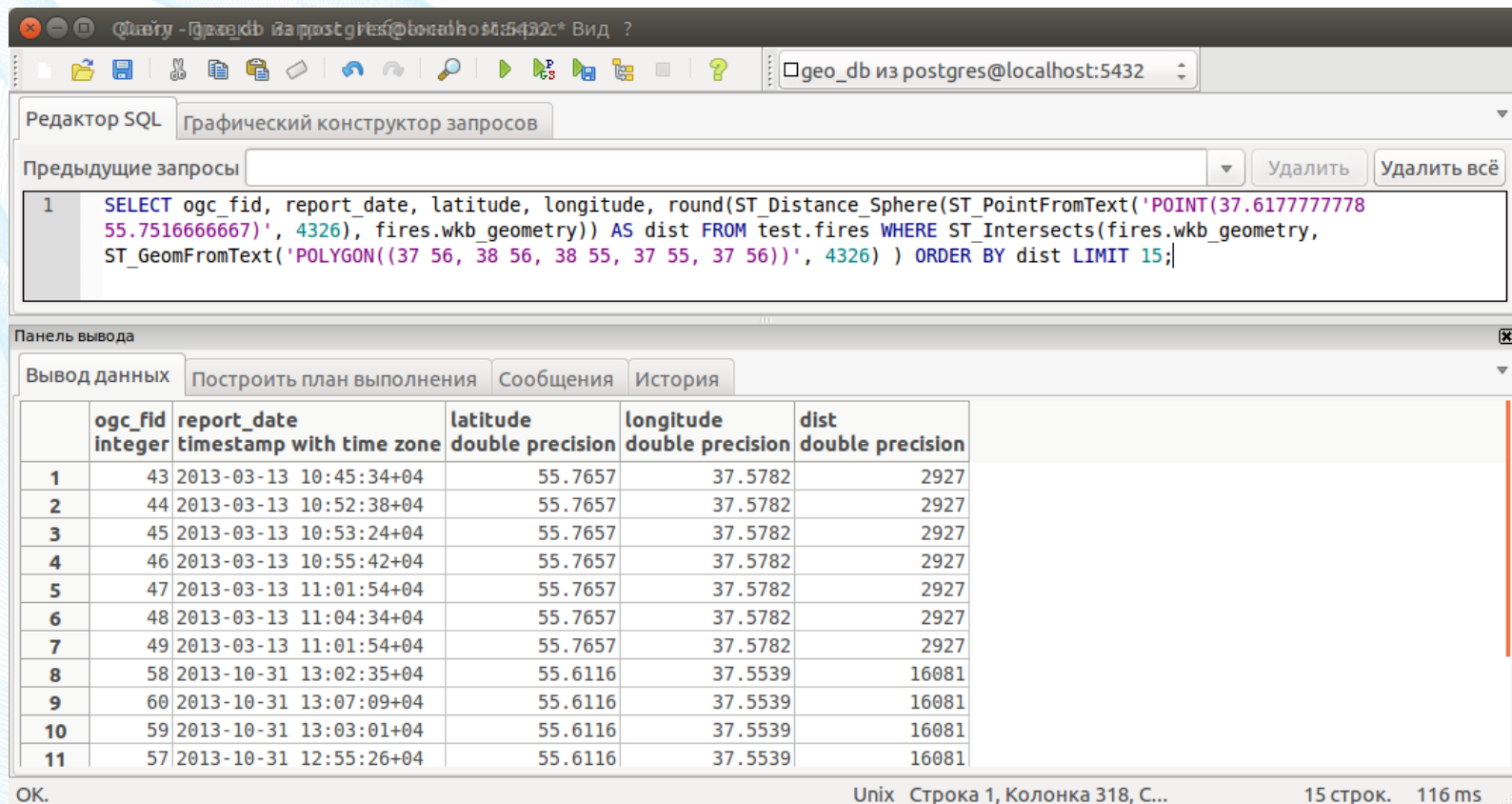
Панель вывода

Вывод данных | Построить план выполнения | Сообщения | История

	geometry text	position double precision	distance double precision
1	POINT(37.0192925999738 55.0610282997165)	27605.9605502065	6896.532331456

OK. Unix Строка 1, Колонка 64, Си... 1 строка. 643 ms

Примеры использования поиск ближайших геометрий



The screenshot shows a PostgreSQL query editor window titled "Query - geo_db из postgres@localhost:5432* Вид ?". The editor has tabs for "Редактор SQL" and "Графический конструктор запросов". Below the tabs is a search bar for previous queries and buttons to delete or delete all. The SQL editor contains a query that finds the nearest fire geometry to a specific point. The results panel at the bottom shows a table with 11 rows of data.

Предыдущие запросы

```
1 SELECT ogc_fid, report_date, latitude, longitude, round(ST_Distance_Sphere(ST_PointFromText('POINT(37.617777778 55.7516666667)', 4326), fires.wkb_geometry)) AS dist FROM test.fires WHERE ST_Intersects(fires.wkb_geometry, ST_GeomFromText('POLYGON((37 56, 38 56, 38 55, 37 55, 37 56))', 4326) ) ORDER BY dist LIMIT 15;
```

Панель вывода

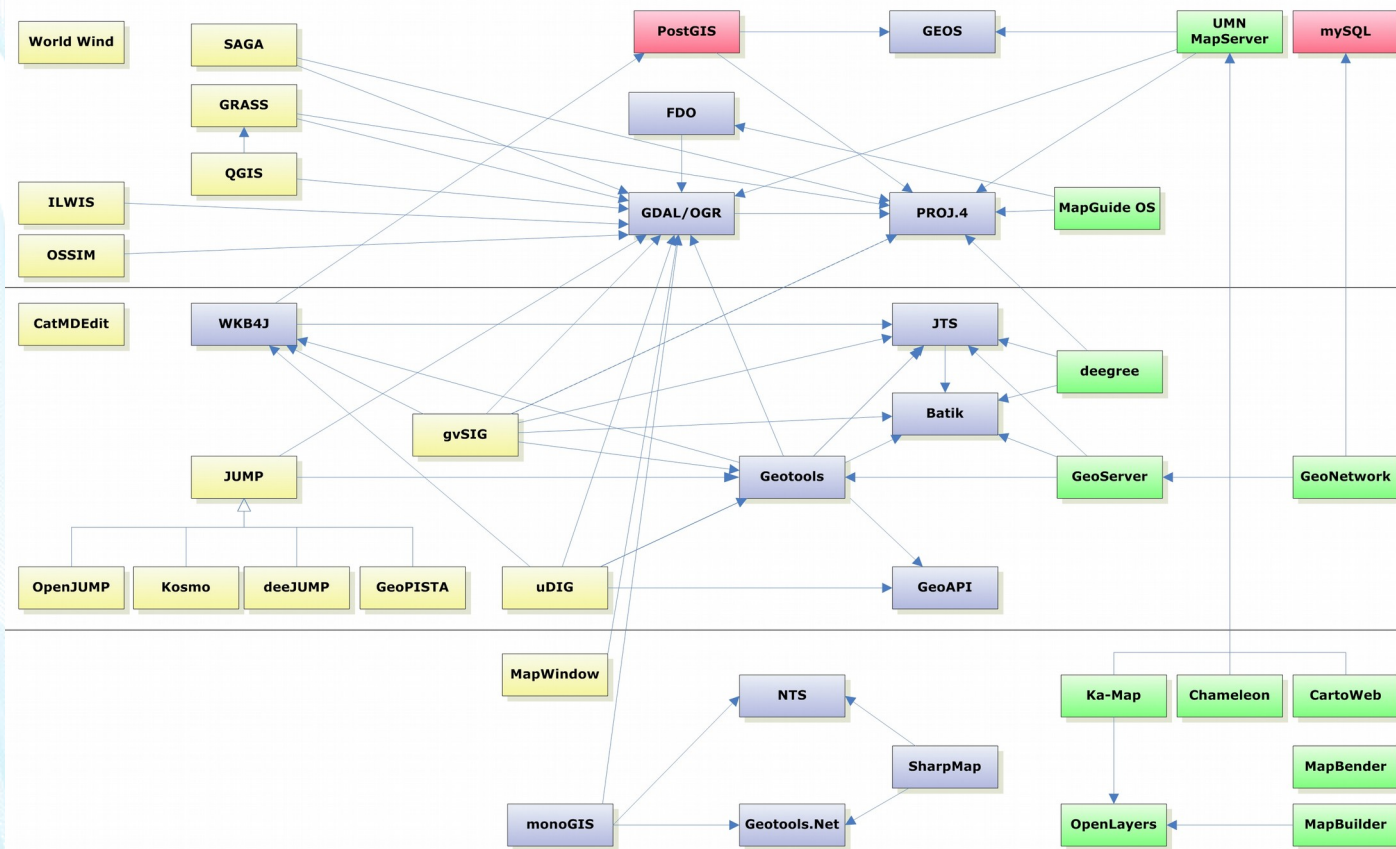
Вывод данных Построить план выполнения Сообщения История

	ogc_fid integer	report_date timestamp with time zone	latitude double precision	longitude double precision	dist double precision
1	43	2013-03-13 10:45:34+04	55.7657	37.5782	2927
2	44	2013-03-13 10:52:38+04	55.7657	37.5782	2927
3	45	2013-03-13 10:53:24+04	55.7657	37.5782	2927
4	46	2013-03-13 10:55:42+04	55.7657	37.5782	2927
5	47	2013-03-13 11:01:54+04	55.7657	37.5782	2927
6	48	2013-03-13 11:04:34+04	55.7657	37.5782	2927
7	49	2013-03-13 11:01:54+04	55.7657	37.5782	2927
8	58	2013-10-31 13:02:35+04	55.6116	37.5539	16081
9	60	2013-10-31 13:07:09+04	55.6116	37.5539	16081
10	59	2013-10-31 13:03:01+04	55.6116	37.5539	16081
11	57	2013-10-31 12:55:26+04	55.6116	37.5539	16081

OK. Unix Строка 1, Колонка 318, С... 15 строк. 116 ms

Работа из прикладного ПО

экосистема ПО ГИС с открытым кодом



Настольные ГИС

Базы пространственных
данных

Веб ГИС

Программные
библиотеки

Работа из прикладного ПО программные библиотеки (GDAL)

- **GDAL** - Geospatial Data Abstraction Library
- проект с **открытым исходным кодом** (лицензия **X/MIT**)
- GDAL предоставляет **единую обобщённую модель данных** для всех поддерживаемых форматов
- GDAL поддерживает **80 векторных** и **135 растровых** форматов
- Последняя версия библиотеки GDAL/OGR - **1.11.1** вышла в сентябре прошлого года
- Сайт библиотеки <http://gdal.org/>

Работа из прикладного ПО программные библиотеки (GDAL)

```
$ ogrinfo "PGB:host=localhost port=5432 dbname=geo_db
user=postgres password=xxxxxx"
INFO: Open of `PGB:host=localhost port=5432 dbname=geo_db
user=postgres password=xxxxxx'
      using driver `PostgreSQL' successful.
1: test.fedokruga (Multi Polygon)
2: test.ch_s_uchas (Multi Point)
3: test.test_gdal (Point)
4: thematic.parts (Line String)
5: thematic.pk (Point)
6: thematic.dps (Point)
7: thematic productions (Point)
8: thematic.roads (Line String)
9: thematic.facilities (Point)
10: thematic.testco (Multi Line String)
11: productions (Point)
12: test.fires (Point)
```

```
$ ogrinfo "PGB:host=localhost port=5432 dbname=geo_db
user=postgres password=xxxxxx" test.fires
INFO: Open of `PGB:host=localhost port=5432 dbname=geo_db
user=postgres password=xxxxxx'
      using driver `PostgreSQL' successful.
Layer name: test.fires
Geometry: Point
Feature Count: 65
Extent: (-122.084090, 2.757590) - (101.697000, 58.394600)
Layer SRS WKT:
GEOGCS["WGS 84",
  DATUM["WGS_1984",
    SPHEROID["WGS 84",6378137,298.257223563,
      AUTHORITY["EPSG","7030"]],
    AUTHORITY["EPSG","6326"]],
  PRIMEM["Greenwich",0,
    AUTHORITY["EPSG","8901"]],
  UNIT["degree",0.0174532925199433,
    AUTHORITY["EPSG","9122"]],
  AUTHORITY["EPSG","4326"]]
FID Column = ogc_fid
```

Работа из прикладного ПО программные библиотеки (GDAL)

```
1 #include "ogr_api.h"
2 #include "ogrsf_frmts.h"
3
4 int main( int nArgc, char ** papszArgv )
5 {
6     OGRRegisterAll();
7
8     const char *pszDataSource = "PGB:host=localhost port=5432 dbname=geo_db user=postgres password=xxxxxx";
9     GDALDataset *poDS = (GDALDataset*) GDALOpenEx( pszDataSource, GDAL_OF_READONLY | GDAL_OF_VECTOR, NULL, NULL, NULL );
10
11     OGRLayer *poLayer = poDS->GetLayer(0);
12     int nGeomFieldCount = poLayer->GetLayerDefn()->GetGeomFieldCount();
13
14     for(int iGeom = 0; iGeom < nGeomFieldCount; iGeom ++ )
15     {
16         if( iGeom > 0 )
17             printf(", ");
18         OGRGeomFieldDefn* poGfldDefn = poLayer->GetLayerDefn()->GetGeomFieldDefn(iGeom);
19         printf( "%s", OGRGeometryTypeToName( poGfldDefn->GetType() ) );
20     }
21
22     OGRFeature *poFeature = NULL;
23     while( (poFeature = poLayer->GetNextFeature()) != NULL )
24     {
25         poFeature->DumpReadable( NULL, papszOptions );
26         OGRFeature::DestroyFeature( poFeature );
27     }
28
29     GDALClose( (GDALDatasetH)poDS );
30     OGRCleanupAll();
31     return 0;
32 }
```

Работа из прикладного ПО настольные программы

The image displays two overlapping software windows. The background window is QGIS 2.6.1-Brighton, showing a map with a pink boundary and a toolbar. A dialog box titled "Добавить таблицы PostGIS" (Add PostGIS tables) is open, showing a table list with columns: Schema, Таблица (Table), Поле (Field), Тип данных (Data type), Тип геометрии (Geometry type), and SRID. The table list includes thematic parts, pk, productions, and thematic productions. The foreground window is NextGIS Manager, showing a tree view of a database connection to geo_db.xconn/thematic. The tree view includes a catalog, connections, and a list of tables: dps, facilities, parts, pk, productions, regions, roads, and testco. The table list is displayed in a table with columns: Имя (Name), Тип (Type), and Размер (Size).

QGIS 2.6.1-Brighton - Выгрузки OSM в shp-файлы (gis-lab.info)

Устаревший файл проекта: Этот проект был создан в предыдущих версиях QGIS. При сохранении проекта он будет отмечен, что может привести к невозможности открытия.

Добавить таблицы PostGIS

Соединения: local

Подключиться Создать Изменить Удалить Загрузить Сохранить

Схема	Таблица	Поле	Тип данных	Тип геометрии	SRID
thematic	parts	wkb_geom...	Геометрия	Линия	4326
thematic	parts		Не опреде...	Нет гео...	
thematic	pk	wkb_geom...	Геометрия	Точка	4326
thematic	pk		Не опреде...	Нет гео...	
thematic	productions	wkb_geom...	Геометрия	Точка	4326
thematic	productions		Не опреде...	Нет гео...	

☒ Показать таблицы без геометрии ☐ Keep dialog open

☐ Параметры поиска

Справка Добавить Фильтр Закрыть

NextGIS Manager - [Подключение к БД/geo_db.xconn/thematic]

Путь: Подключение к БД/geo_db.xconn/thematic

Слой:

Вид дерева

- Каталог
 - Подключения
 - Подключение к БД
 - bishop_server_virtual.xconn
 - geo_db.xconn
 - public
 - test
 - thematic
 - IBS
 - NextGIS VPN
 - ore.xconn
 - ore_local.xconn
 - test local db.xconn
 - test.xconn
 - zab.xconn
 - ГИС-Лаб

Содержимое

Имя	Тип	Размер
dps	Класс пространственных объектов Post...	288,0 KB
facilities	Класс пространственных объектов Post...	16,4 MB
parts	Класс пространственных объектов Post...	15,0 MB
pk	Класс пространственных объектов Post...	12,6 MB
productions	Класс пространственных объектов Post...	280,0 KB
regions	Класс пространственных объектов Post...	12,4 MB
roads	Класс пространственных объектов Post...	4,2 MB
testco	Класс пространственных объектов Post...	168,0 KB

Выбрать вид

Работа из прикладного ПО серверные программы



Работа из прикладного ПО серверные программы

Mapnik

```
<Layer name="countries" status="on" srs="+proj=latlong +datum=WGS84">
  <StyleName>countries_style_label</StyleName>
  <Datasource>
    <Parameter name="type">postgis</Parameter>
    <Parameter name="host">localhost</Parameter>
    <Parameter
name="dbname">geodjango_geographic_admin</Parameter>
    <Parameter name="user">postgres</Parameter>
    <Parameter name="password"></Parameter>
    <Parameter name="table">(select ST_Buffer(ST_Centroid(geometry),2)
as geometry, name from world_worldborders) as world</Parameter>
    <Parameter name="estimate_extent">>false</Parameter>
    <Parameter name="extent">-180,-90,180,89.99</Parameter>
  </Datasource>
</Layer>
```

MapServer

```
LAYER
NAME "boundary"
CONNECTIONTYPE postgis
CONNECTION "user=postgres password=xxxx dbname=city"
DATA "the_geom from boundary"
TYPE POLYGON
CLASS
  COLOR 200 120 123
END
```

Сообщество

Расширение **PostGIS** было разработано канадской компанией **Refractions Research Inc** в качестве исследовательского проекта

В проекте планировалось реализовать поддержку важных технологий:

- полная поддержка OpenGL
- расширенная топология
- средства для просмотра и редактирования в GUI
- средства доступа к геопространственным данным посредством Веб

Сейчас проект управляется комитетом (**PostGIS Project Steering Committee, PSC**)

PostGIS является проектом **OSGeo Foundation**

Официальный адрес в интернет **<http://postgis.org/>**



Стандарты и протоколы, совместимость

Объекты ГИС, поддерживаемые PostGIS, являются надмножествами "**Simple Features**", определенных Консорциумом **Open Geospatial Consortium** (OGC)

PostGIS расширяет стандарт поддержкой координат **3DZ**, **3DM** и **4D**.

Спецификация OGC определяет два стандартных способа определения пространственных объектов:

- в форме **Well-Known Text** (WKT)
- в форме **Well-Known Binary** (WKB)

Well-Known Text (WKT)

- 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 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
- GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))

Well-Known Binary (WKB)

- Точка POINT(2.0 4.0) в формате WKB представляется следующей строкой:

000000000140000000000000004010000000000000, где:

- 1-byte integer 00 or 0: big endian
- 4-byte integer 00000001 or 1: POINT (2D)
- 8-byte float 4000000000000000 or 2.0: x-coordinate
- 8-byte float 4010000000000000 or 4.0: y-coordinate

Extended Well-Known Text (EWKT)

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

Спасибо за внимание!

e-mail: **dmitry.baryshnikov@nextgis.ru**, **info@nextgis.ru**, web: **www.nextgis.ru**