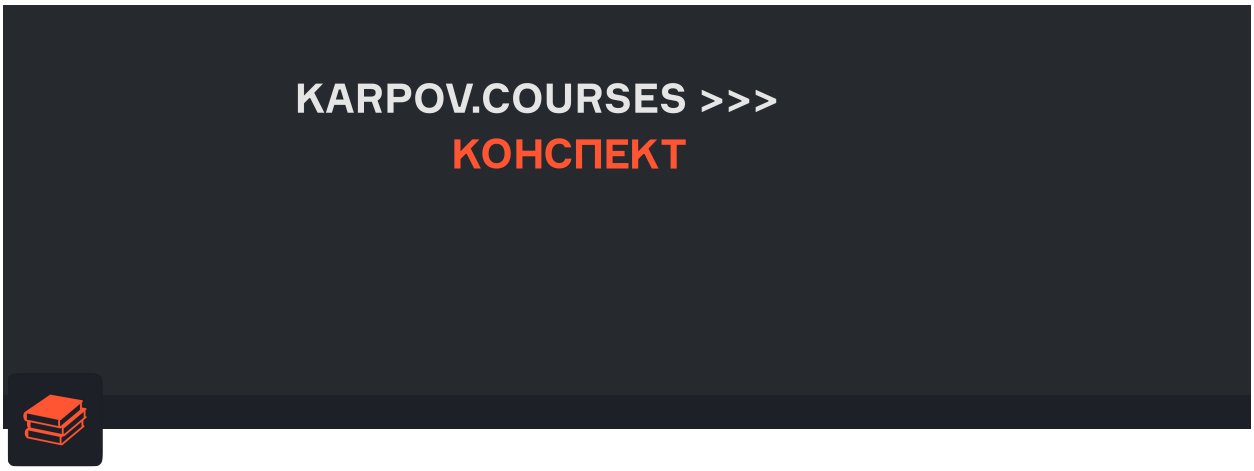

> Конспект > 6 урок > Дополнительные возможности GreenPlum: Практика

> Оглавление

> Как работает PostGIS

Окружение - это кластер Greenplum, собранный на одном хосте. На него установлен PostGIS версия 2.1.5. В нем есть Greenplum Text - расширение для работы с текстом. Также установлено расширение PL\Python.

Можно использовать и PL\Perl, или Java, или R

PostGIS – расширение для географических и пространственных данных, при помощи которого можно:

- обрабатывать точки, кривые, фигуры;
- получать статистику по ним;
- строить карты или находить маршруты;
- сохранять полученные данные в таблицу;
- преобразовывать географические и/или пространственные данные в друг друга;
- индексировать и обрабатывать полученные данные.

Для использования расширения можно:

- в коммерческой версии скачивается и устанавливается .package;
- или это расширение можно собрать их исходников и установить самостоятельно

Для примера создадим таблицу, где одним из типов данных будет geometry.

Полученные геометрические данные можно обрабатывать на сегментах, частями ускоряя обработку массива. Можно вставить:

- полигон (POLYGON)
- линию (LINESTRING), прямую или ломаную
- ломаную линию, из которой будет создан полигон (MULTIPOINTS)

```
CREATE TABLE geom_test (gid int4, geom geometry,
  name varchar(25) );

INSERT INTO geom_test ( gid, geom, name )
  VALUES (1, 'POLYGON((0 0 0,0 5 0,5 5 0,5 0 0,0 0 0))', '3D Square');
INSERT INTO geom_test ( gid, geom, name )
  VALUES (2, 'LINESTRING((1 1 1,5 5 5,7 7 5))', '3D Line');
INSERT INTO geom_test ( gid, geom, name )
  VALUES (3, 'MULTIPOINT((3 4,8 9))', '2D Aggregate Point');
INSERT INTO geom_test ( gid, geom, name )
  VALUES (1, ST_Polygon(ST)GeomFromText('LINESTRING(75.15 29.53,77 30,77.6 29.5 75.15 29.53)'), 4326, '3D Poly');
```

Можно выбрать из таблицы, какие типы данных содержатся в каком поле с каким именем

```
SELECT geometrytype (geom), NAME FROM geom_test
```

Можно выбрать все содержимое таблицы и получить наглядное представление об информации, которая хранится в геометрическом поле.

```
SELECT * FROM geom_test;
```

Еще одной возможностью DBeaver является его способность отобразить объект на карте и посмотреть, что это такое.

```
SELECT ST_Polygon (ST_GeomFromText('LINESTRING(75.15 29.53,77 30,77.6 29.5, 75.15 29.53)'), 4326);
```

Можно поставить точку для произвольных координат и посмотреть ее местонахождение на реальной карте.

Расширение PostGIS оно также работает в PostgreSQL. При необходимости анализа большого объема информации она складывается в Greenplum и обрабатывается параллельно на всех сегментах.

PostGIS поддерживает создание GiST, что позволяет ускорить обработку географических запросов и проще выполнять операции поиска-вхождения или принадлежности к одной и той же зоне разных объектов. Такой географический GiST индекс будет представлять собой дерево, в котором будут последовательно отображаться все объекты.

Геометрические поля можно добавлять к уже существующим таблицам. Для этого применяется специфический оператор PostGIS

```
AddGeometryColumn
```

```
DROP TABLE geotest;
CREATE TABLE geotest (id INT4, name VARCHAR(32) );
SELECT AddGeometryColumn ('geotest','geopoint', 4326,'POINT',2);
```

После создания таблицы, можно добавить к ней колонку с геометрическими данными, которые можно дополнять и смотреть их на карте.

```
INSERT INTO geotest (id, name, geopoint)
  VALUES (1, 'Olympia', ST_GeometryFromText('POINT(-122.90 46.97)', 4326));
INSERT INTO geotest (id, name, geopoint)
  VALUES (2, 'Renton', ST_GeometryFromText('POINT(-122.90 47.50)', 4326));

SELECT name, ST_AsText(geopoint) FROM geotest;

SELECT geopoint FROM geotest WHERE name ='Olympia'
```

Таким образом, расширение PostGIS позволяет манипулировать большими объемами картографических и географических данных

> Что можно сделать с Greenplum Text

- Разбиение текста на лексемы (группы ассоциированных слов);
- Построение индексов по тексту;
- Ранжирование текст по релевантности результатов поиска;

Greenplum Text доступен в open-source версии, в коммерческой версии это VMware Tanzu - он разворачивает Apache Solr кластер рядом с GreenPlum и использует его возможности по индексированию и хранению индексов текста для ускорения и облегчения поиска текста.

Основные единицы хранения и использования текста:

- документ, тип данных tsvector
- запрос, который представлен типом tsquery

Можно создать таблицу, в которой будет храниться документ и его нормализованная форма типа tsvector. К этому полю можно обращаться при помощи типов tsquery, чтобы понять вхождение поисковых слов.

GreenPlum позволяет ранжировать документы по количеству и качеству розыскиваемых лексем.

Посмотрим синтаксис запроса, с помощью которого мы пытаемся оценить вхождение необходимых нам слов в документ.

```
SELECT 'a fat cat sat on a mat and ate a fat rat'::tsvector @@ 'cat & rats'::tsquery;
```

Результатом будет true or false (правда или ложь). Так как документ не приведен к лексемам по набору правил, поэтому результат будет false. Это произошло из-за того, что слово rat и rats являются для Greenplum Text разными. Поэтому в своей работе вам необходимо сперва создать лексемы и по ним оценивать результативность вашего запроса.

Greenplum Text позволяет преобразовать текст в формат tsvector. Для этого необходимо вызвать функцию to_tsvector, которая преобразует документ в формат tsvector.

С помощью функции to_tsvector можно получать комбинированное представление документа, создавая объединенный tsvector из разных полей таблицы. Эти поля могут быть отранжированы по значимости тех или иных полей (на ваш выбор)

Пример запроса по имени, объединенный по типу, с рассмотрением комментариев.

```
SELECT to_tsvector('english',p_name) || to_tsvector('english',p_type) || to_tsvector('english',p_comment)
FROM tpch1.part LIMIT 50
```

Используя такое составное поле, можно искать по всему документу, найдя вхождение интересующих нас запросов

> Как применить PL\Python

Greenplum поддерживает использование следующих языков:

- PL\Perl
- Java
- R
- PL\ Python
- C/C++

Можно вставлять текст, написанный на этих языках программирования и это все будет выполняться над данными, которые хранятся или будут храниться в Greenplum.

Все дополнительные языки нужно прописать как отдельные расширения.
Командой create extension можно создать расширение для языка PL\Python

Расширения могут быть безопасными и небезопасными. Последние выполняются в операционной системе и могут получить доступ к файлам операционной системы. Поэтому такие расширения могут создавать суперпользователи.

Можно создать функцию, которая проверяет существует или не существует файл в системе. Выполнение самой функции будет происходить в самой операционной системе, создавая такую функцию мы можем использовать все модули доступные Python, который установлен вместе с Greenplum.

Например, мы хотим проверить существует ли такой файл в домашней директории

```
SELECT pyfileexists (‘home/gpadmin/gpinitsystem_config.txt’)
```

Такого файла нет

Формируем новый запрос и проверяем наличие вот этого файла - файл на месте

```
SELECT pyfileexists (‘home/gpadmin/gpinitsystem_config’)
```

Для примера можно создать функцию шифровальщика, который будет кодировать предлагаемую строку, в соответствии со словарем шифрования.

```
CREATE OR REPLACE FUNCTION pyencoder(instr text,
    mapfrom text DEFAULT 'abcdedfhijklmnopqrstuvwxyz',
    map to text DEFAULT 'bc567ghijk432opqrstuvwxyza' )
RETURN text AS
$$
    import string
    strt = string.maketrans(mapfrom, mapto)
    retirn instr.lower().translate(strt)
$$
  LANGUAGE 'plpythonnu' VOLATILE
```

Создаем декодер

```
CREATE OR REPLACE FUNCTION pydecoder(instr text,
    mapfrom text DEFAULT 'abcdedfhijklmnopqrstuvwxyz',
    map to text DEFAULT 'bc567ghijk432opqrstuvwxyza' )
RETURN text AS
$$
    import string
    strt = string.maketrans(mapto, mapfrom)
    retirn instr.lower().translate(strt)
$$
  LANGUAGE 'plpythonnu' VOLATILE
```

И проверяем как работает первая часть, полученный ответ нас удовлетворяет

```
select pyencoder('Welcome to Africa');
x735p27 up bhsj5b
```

Безопасное использование языков подразумевает использование контейнеров. Для этого необходимо создать контейнер, установить язык программирования и импортируемые модули. Такой порядок действий защищает операционную систему, так как вызываемая функция выполняется в контейнере. Такую функцию может выполнять любой пользователь.

Это подключаемый модуль PL/Container

> Глоссарий

Geometry - это плоский пространственный тип данных в евклидовом пространстве (плоской системе координат)

Лексемы — это группа ассоциированных слов или слово как абстрактная единица морфологического анализа. Сейчас в данном значении используют термин семантическое поле.