# Задание на лабораторную 4

В 4 лабораторной вам предстоит:

* Восстановить базу данных NYC
* Установить, если их нет, расширения: postgis, pg\_stat\_statements, pgcrypto, hstore, uuid-ossp
* Выполнить запросы

Сделайте скриншоты каждого запроса и их результаты (вывод).

Скриншоты выкладывать в папку «Лабораторная 4» вашего репозитория.

Не забудьте сделать коммит, чтобы сдать лабораторную.

При возникновении проблем и вопросов пишите на почту [oleg.glukhikh@urfu.ru](mailto:oleg.glukhikh@urfu.ru).

## Восстановление БД NYC

1. Шаг 1 – Принимающая БД

Первое, что нам нужно сделать – это создать принимающую БД, в которую будем восстанавливать бэкап.

Воспользуемся GUI pgAdmin. Нажмите правой кнопкой мыши на Databases и выберите Create > Database.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Заполните открывшееся окно (введите название БД):

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Нажмите Save и перейдите к созданной БД.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Нажмите правой кнопкой мыши на БД, в контекстном меню выберите Query Tool.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Там введите и запустите по очереди:

CREATE EXTENSION postgis;

SELECT postgis\_full\_version();

1. Шаг 2 – Скачивание бэкапа БД

К лабораторной работе прилагается архив «postgis-workshop-2020.zip». Распакуйте его, внутри «postgis-workshop» будут папки «data» и «printing». В папке «data» вы увидите:

Изображение выглядит как текст, снимок экрана, меню, число

Автоматически созданное описание

Файл бэкапа называется «nyc\_data.backup». Далее – откройте pgAdmin и контекстное меню принимающей базы данных, выберите «Restore»:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

В новом окне выберите формат «Custom or tar» и нажмите символ папки справа от поля для названия файла.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Далее перейдите во вкладку «Data/Objects» этого окна, выберите «Do not save: Owner» и нажмите кнопку «Restore». Восстановление базы данных должно завершиться без ошибок.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

После завершения загрузки нажмите правой кнопкой мыши на БД nyc и выберите опцию «Refresh», чтобы обновить информацию клиента о том, какие таблицы существуют в базе данных.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

## Установка расширений

Находясь внутри БД nyc, воспользуйтесь командой:

CREATE EXTENSION IF NOT EXISTS extension\_name

WITH SCHEMA schema\_name

VERSION version

CASCADE

В команде также указаны необязательные, но полезные параметры, которые можно использовать при установке.

* IF NOT EXIST: проверка существования расширения с таким именем.
* WITH SCHEMA: имя схемы, для которой будет произведена установка. Если не указано, то установится в текущую схему.
* VERSION: версия для установки. Если не указано, то будет установлена последняя.
* CASCADE: автоматическая установка всех дополнительных расширений, необходимых для работы.

Вам нужно установить расширения:

1. PostGIS
2. pgcrypto
3. hstore
4. uuid-ossp
5. pg\_stat\_statements

В ходе установки базы данных NYC PostGIS уже был установлен. Последующие библиотеки можно установить схожим образом, кроме pg\_stat\_statements.

Перед установкой найдите и откройте файл «postgresql.conf» (пример расположения: C:\Program Files\PostgreSQL\15\data\postgresql.conf) и внесите следующие изменения:

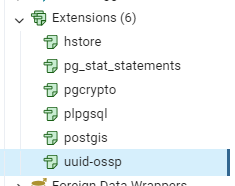
shared\_preload\_libraries = 'pg\_stat\_statements'

Потребуется раскомментировать строку с «shared\_preload\_libraries» (убрать символ «#» в начале строки).

После установки расширения pg\_stat\_statements, чтобы использовать его данные, потребуется перезапустить сервер PostgreSQL. Это можно сделать несколькими способами, например:

* Перезапуск компьютера
* Команда «pg\_ctl restart» (<https://www.postgresql.org/docs/current/app-pg-ctl.html>) (<https://postgrespro.ru/docs/postgrespro/15/app-pg-ctl>)

Чтобы показать выполнение этого задания, сделайте скриншот БД nyc с открытым списком «Extensions».



## Запросы

В ходе этого задания вам потребуется выполнить запросы:

* обращающиеся к функциям установленных расширений;
* реализующие все шесть типов соединений.

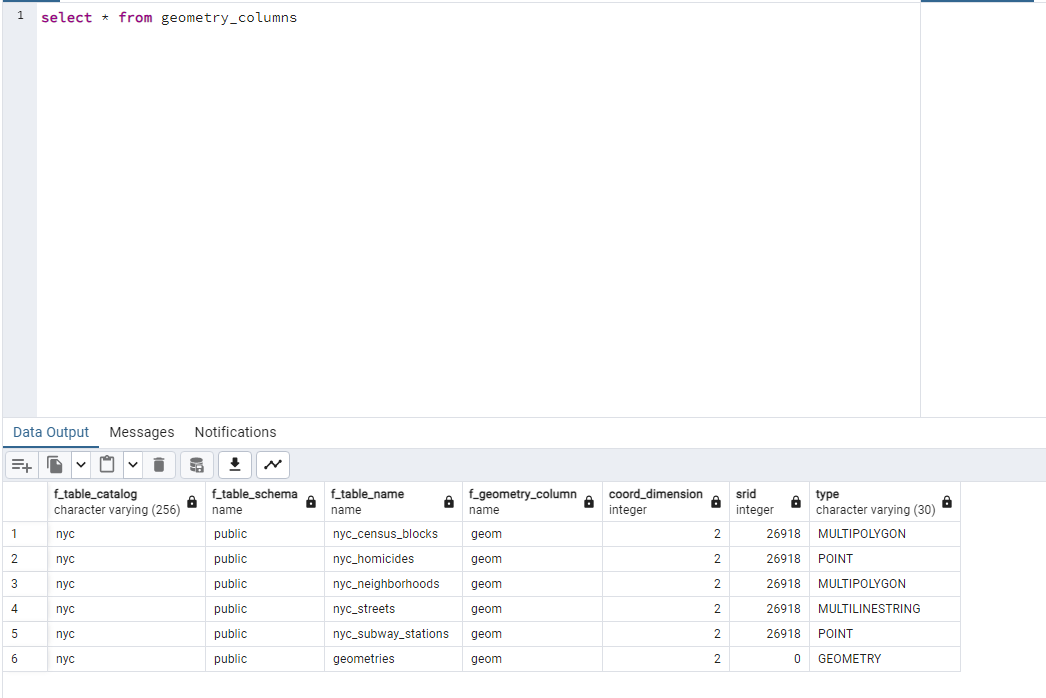
**PostGIS**

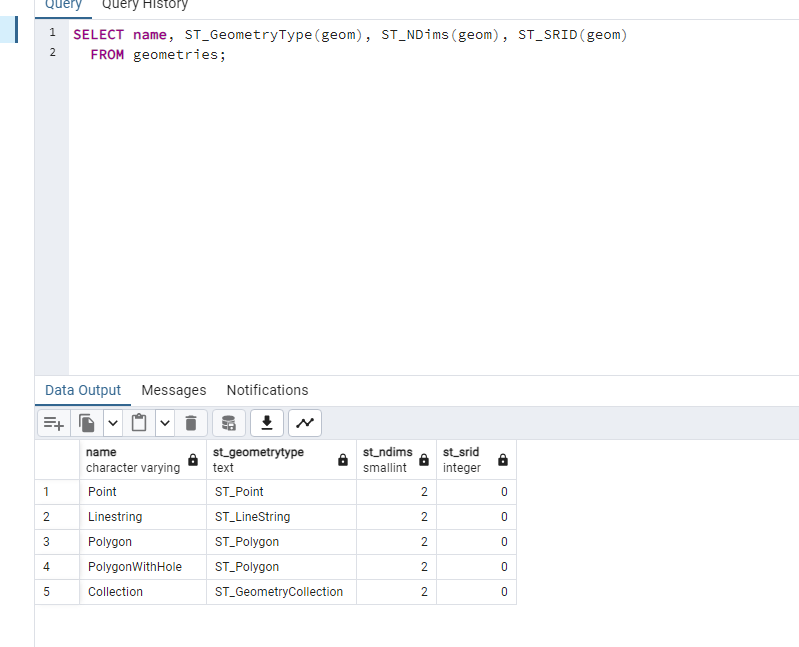
Изучите материал и запустите скрипты:

<https://postgis.net/workshops/postgis-intro/simple_sql_exercises.html>

<https://postgis.net/workshops/postgis-intro/geometries.html>

Сделайте скриншот таблиц «geometries» из пункта 9.3, «geometry\_columns» из пункта 9.2.





**pgcrypto**

Изучите материалы:

<https://postgrespro.ru/docs/postgresql/15/pgcrypto>

<https://docs.arenadata.io/ru/ADPG/current/how-to/manage-database-access/encryption.html>

Запустите скрипты (потребуется несколько дополнительных скриптов из второй статьи):

* SELECT (crypt('my\_secure\_password', gen\_salt('bf'))) as c,

(encode(digest('MyPassword','sha256'),'hex')) as e,

(hmac('MyPassword', 'My\_key', 'sha256')) as h;

* UPDATE table1 SET password\_hash = crypt('new\_password', gen\_salt('md5'));
* SELECT (password\_hash = crypt('entered\_password', password\_hash)) AS password\_match FROM table1;
* SELECT username, pgp\_sym\_decrypt(password::bytea, 'aes\_key') as password

FROM users WHERE (username LIKE '%Robert%');

Сделайте скриншот таким образом, чтобы все запросы были видны в окне «Query», а результат (DataOutput) был по одному (любому) из них.



**hstore**

Изучите материалы:

<https://postgrespro.ru/docs/postgresql/15/hstore>

<https://docs.arenadata.io/ru/ADPG/current/how-to/queries/use-complex-types/hstore.html>

Запустите скрипты (потребуется несколько дополнительных скриптов из второй статьи):

* SELECT name, attributes->'author' as author FROM products

WHERE attributes -> 'genre' = 'sci-fi';

* UPDATE products

SET attributes = attributes || '"freeshipping"=>"yes"' :: hstore

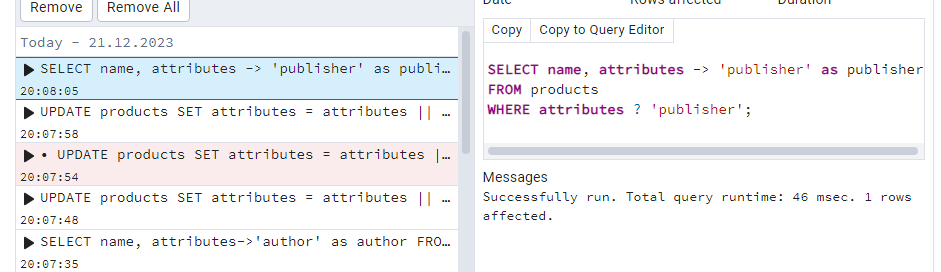
WHERE name = 'Hyperion';

* UPDATE products SET attributes = attributes || hstore(ARRAY[['pages', '32'],['media', 'ebook']]) WHERE id = 4;
* SELECT name, attributes -> 'publisher' as publisher

FROM products

WHERE attributes ? 'publisher';

Сделайте скриншот таким образом, чтобы все перечисленные запросы были видны в окне «Query», а результат («DataOutput») был по одному (любому) из них.



**uuid-ossp**

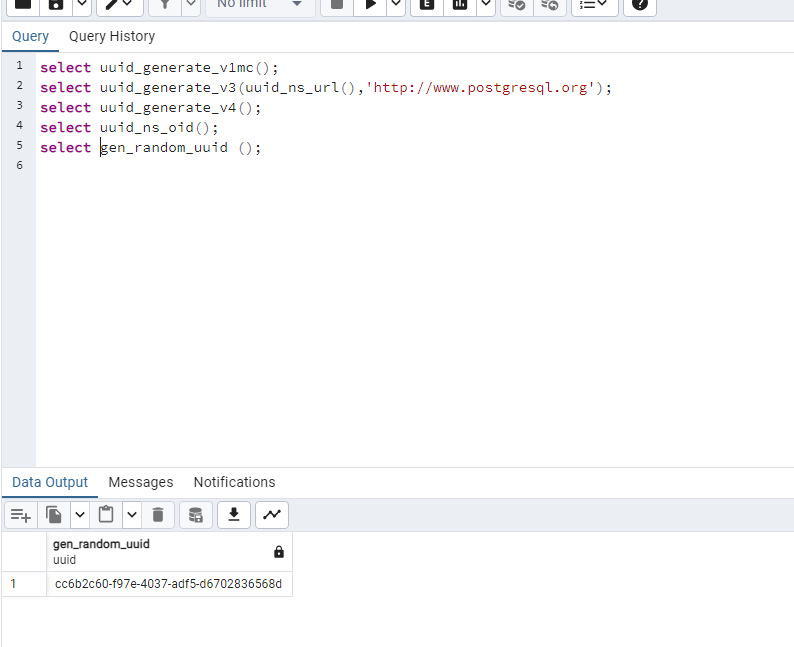
Изучите материал:

<https://postgrespro.ru/docs/postgresql/15/uuid-ossp>

Запустите функции:

* uuid\_generate\_v1mc
* uuid\_generate\_v3
* uuid\_generate\_v4
* uuid\_ns\_oid
* gen\_random\_uuid (из другого расширения)

Сделайте скриншот таким образом, чтобы запросы с перечисленными функциями были видны в окне «Query», а результат («DataOutput») был по одному (любому) из них.



**pg\_stat\_statements**

Изучите материалы:

<https://postgrespro.ru/docs/postgresql/15/pgstatstatements>

<https://timeweb.cloud/tutorials/postgresql/rasshireniya-dlya-postgresql>

Запустите скрипты:

* SELECT query, (total\_exec\_time / 1000 / 60) as total\_min, mean\_exec\_time as avg\_ms, calls

FROM pg\_stat\_statements

ORDER BY 1

DESC LIMIT 10;

* SELECT query, calls, total\_time, rows, shared\_blks\_hit, shared\_blks\_read

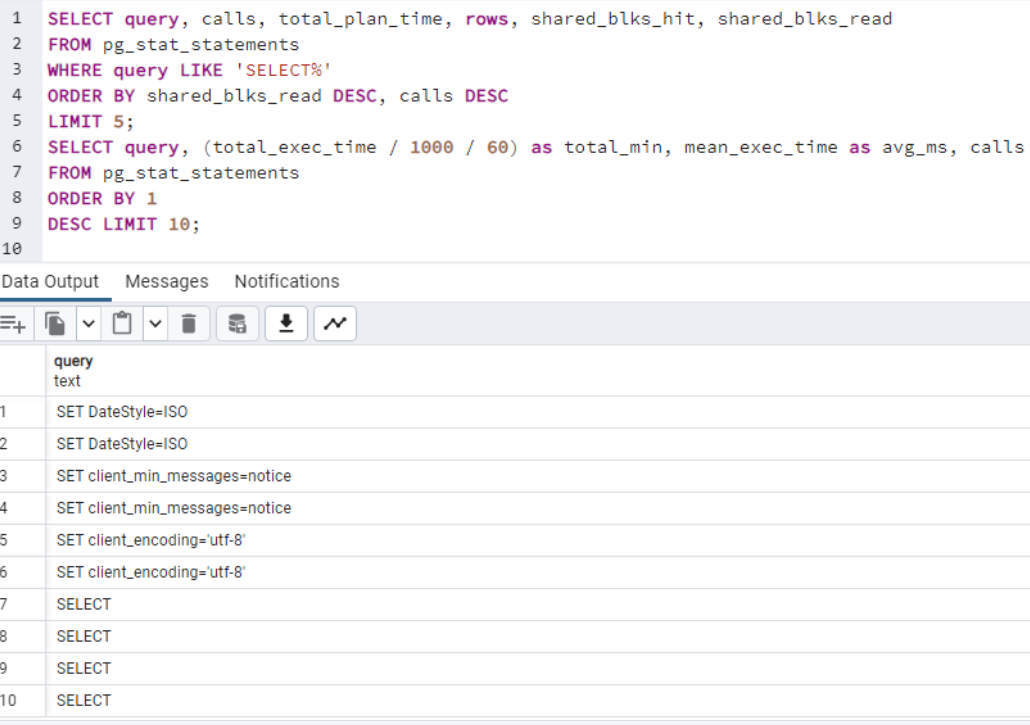
FROM pg\_stat\_statements

WHERE query LIKE 'SELECT%'

ORDER BY shared\_blks\_read DESC, calls DESC

LIMIT 5;

Сделайте скриншот таким образом, чтобы оба запроса были видны в окне «Query», а результат («DataOutput») был по одному (любому) из них.



**Соединения**

Изучите материалы:

<https://postgrespro.ru/docs/postgresql/15/sql-select>

<https://docs.arenadata.io/ru/ADPG/current/how-to/queries/join.html>

Реализуйте перечисленные соединения:

* Естественное соединение (INNER JOIN)
* Внешнее соединение, одно из:
  + LEFT OUTER JOIN
  + RIGHT OUTER JOIN
  + FULL OUTER JOIN
* Соединение таблицы с самой собой или декартово соединение (CROSS JOIN)
* Полусоединение или анти-полусоединение

В качестве основы для запросов можно использовать примеры из статей, описания демонстрационной базы данный или любой другой источник (в т. ч. можно создать и заполнить нужные таблицы), в одном запросе можно сделать несколько соединений. Основное условие – в запросе комментарием (он указывается с помощью «--» в начале строки) явно укажите, какое или какие соединения присутствуют.

