



TEXHOTPEK

Урок 5

PostgreSQL

Кухтичев Антон



Не забудьте отметить!!!!

План занятия



- PostgreSQL;
- Что такое SQL;
- ORM в Django;
- Обсуждение квиза #3;
- Квиз #4!



TEXHOTPEK

PostgreSQL

Установка PostgreSQL



Установка на Ubuntu

```
sudo apt install postgresql-10
```

Установка на MacOS

```
brew install postgresql@10
```

Проверка подключения

```
sudo -u <USER_NAME> psql
```

Пользователи и базы из коробки



Пользователи

- `<USER_NAME>` - супер-пользователь внутри Postgres, может все;

Базы данных

- `template1` - шаблонная база данных;
- `template0` - шаблонная база данных на всякий случай;
- `postgres` - база данных по-умолчанию, копия `template1`.

Пользователи и базы из коробки



Создать пользователя и базу

```
postgres=# CREATE USER quack_db WITH password  
's3cr3t';  
CREATE ROLE  
postgres=# CREATE DATABASE quack_db OWNER quack;  
CREATE DATABASE
```

Проверить подключение

```
$ psql --host=localhost --user=quack quack  
Password for user quack: *****
```

Использование psql



```
$ psql db_name
```

или так

```
$ psql --host=127.0.0.1 --user=db_user  
db_name
```

Если вы хотите подключаться к своей базе без ввода пароля

```
postgres=# CREATE USER your_linux_user;  
postgres=# GRANT ALL ON DATABASE db_name TO  
your_linux_user;
```


Использование psql



`\?` показать список команд;

`\du` показать список пользователей с привилегиями;

`\l` показать список баз данных;

`\c db_name2` подключиться к другой базе данных;

`\dt` показать список таблиц;

`\d table_name` показать колонки таблицы

`\x` переключить режим вывода



ТЕХНОТРЕК

ЯЗЫК **SQL**

Типы данных



- `smallint`, `integer`, `bigint` - целое, 2/4/8 байт;
- `smallserial`, `serial`, `bigserial` - целое, 2/4/8 байт, автоувеличение;
- `timestamp` - дата и время, с точностью до микросекунд;
- `text` - строка произвольной длины;
- `uuid` - UUID;
- `jsonb` - JSON документ;

Виды операций



- SELECT - выборка данных;
- UPDATE - обновление значений столбцов;
- DROP - удаление;
- ALTER - изменение;
- JOIN - объединение;
 - INNER JOIN (или просто JOIN);
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN
 - CROSS JOIN
- INSERT - вставка;
- CREATE - создание.

Создание таблиц



```
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    nick TEXT NOT NULL UNIQUE  
        CHECK (length(nick) < 32),  
    name TEXT NOT NULL  
        CHECK (length(name) < 32)  
)
```

Создание таблиц (2)



```
CREATE TABLE messages (  
    message_id SERIAL PRIMARY KEY,  
    user_id INTEGER NOT NULL  
        REFERENCES users(user_id),  
    content TEXT NOT NULL  
        CHECK (length(content) < 65536),  
    added_at TIMESTAMP NOT NULL DEFAULT NOW()  
)
```

Изменение и удаление таблиц



```
ALTER TABLE users
    ADD COLUMN avatar TEXT DEFAULT NULL,
    DROP CONSTRAINT users_name_check,
    ADD CONSTRAINT users_name_check
        CHECK (length(name) < 64);
DROP TABLE users;
DROP TABLE users CASCADE; -- не повторять в
production :)
```

Добавление данных



```
INSERT INTO users (nick, name)
VALUES ('mort.rainey', 'Морт Рейни'),
      ('john.shooter', 'Кокни Шутер');
INSERT INTO users
VALUES (5, 'todd.downey', 'Тодд Дауни');
```


Обновление данных



```
UPDATE users  
SET name = 'Не такой как все'  
WHERE user_id = 2;  
DELETE FROM users  
WHERE user_id % 2 = 0;
```

Выборка данных



```
SELECT message_id, content, added_at::DATE
FROM messages
WHERE user_id = 3
      AND message_id < 100500
ORDER BY added_at DESC
LIMIT 10
```

Выборка из нескольких таблиц



```
SELECT nick AS author, name, messages.*  
FROM messages  
JOIN users USING (user_id)  
WHERE user_id = 3  
      AND message_id < 100500  
ORDER BY added_at DESC  
LIMIT 10
```



TEXHOTPEK

Django и PostgreSQL

Добавить БД в settings.py



```
DATABASES = {  
    'default': {  
        'ENGINE':  
'django.db.backends.postgresql_psycopg2',  
        'NAME': 'quack_db',  
        'USER': 'quack',  
        'PASSWORD': 's3cr3t',  
        'HOST': '127.0.0.1',  
        'PORT': '5432',  
    }  
}
```

А дальше всё как обычно



```
# Делаем миграцию  
./manage.py migrate
```

```
# Создаем суперпользователя  
./manage.py createsuperuser
```

```
# и запускаем сервер  
./manage.py runserver.
```

Некоторые полезные опции



- `./manage.py dbshell` - запустить клиент базы данных;
- `./manage.py showmigrations` - показать историю миграций;
- `./manage.py makemigrations` - создать миграции;
- `./manage.py migrate` - применить миграции;
- `./manage.py shell` - запустить python shell;
- `./manage.py validate` - проверить структуру моделей.

Типы полей



- IntegerField
- AutoField
- BooleanField
- CharField
- EmailField
- DateField
- DateTimeField
- FloatField
- TextField

Типы полей (2)



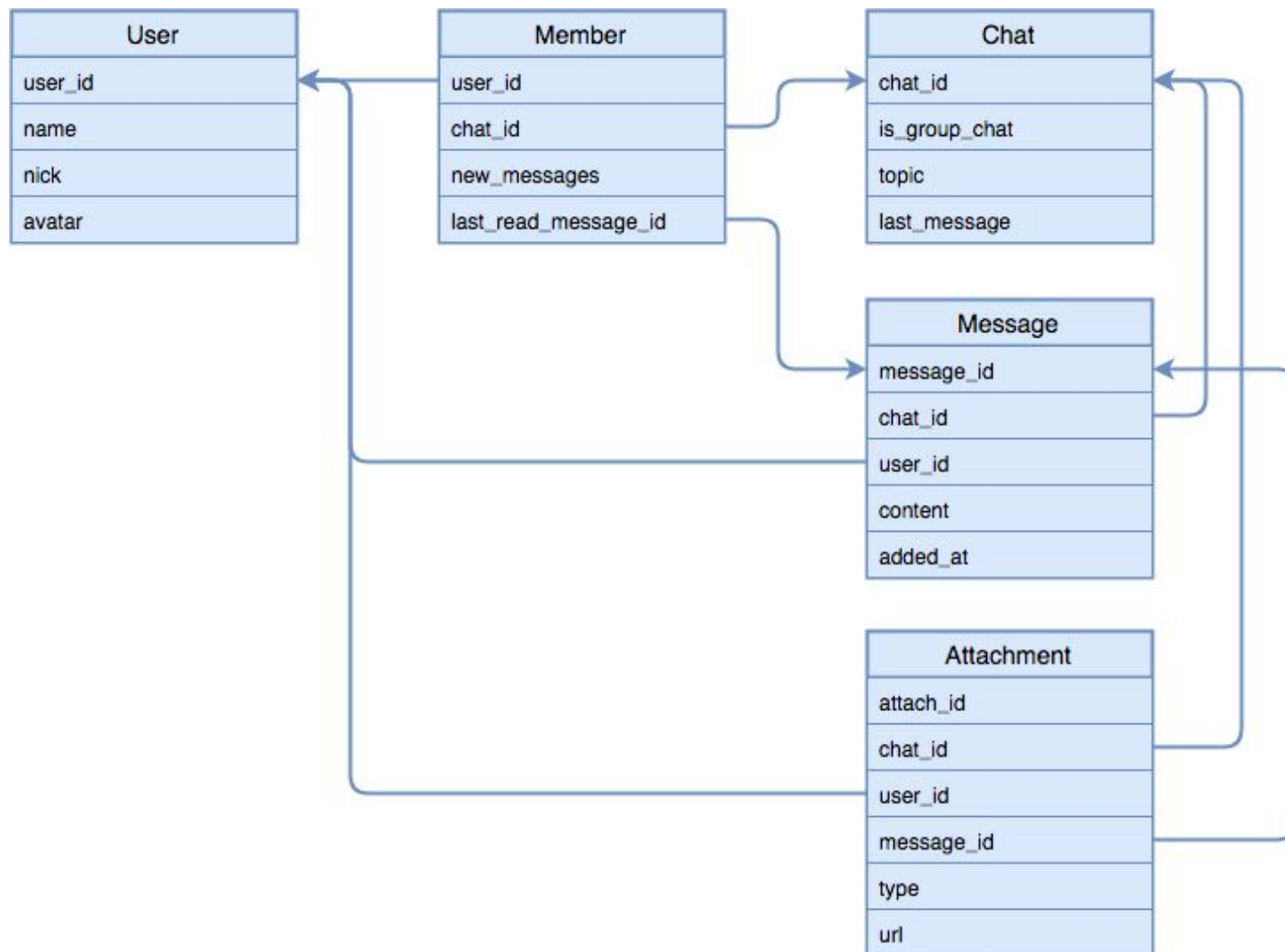
- Один-к-одному -> `OneToOneField`
- Один-ко-многим -> `ForeignKey`
- Многим ко многим -> `ManyToManyField`

SQL запросы напрямую из Django (2)



```
>>> people = Person.objects.raw('SELECT *,  
age(birth_date) AS age FROM myapp_person')  
>>> for p in people:  
...     print("%s is %s." % (p.first_name,  
p.age))
```

Предлагаемая схема БД



Домашнее задание №3



1. Установить Postgres, создать нового пользователя и БД и настроить доступ - 4 балла;
2. Спроектировать базу данных проекта, подготовить модели и мигрировать их в БД - 5.

Срок сдачи

*Нет сроков, только
ограничение в 2 дз за
занятие.*



ТЕХНОТРЕК



Результаты квиза

Вопрос #1



На каком порту работает протокол HTTP по умолчанию?

1. 8080
2. 8000
3. 80
4. 443

Вопрос #2



Что является путём (path) в следующем URL'е
`https://example.com:5000/some/file/doc.html?n=60&f=0#12345`

1. `https`
2. `example.com:5000`
3. `/some/file/doc.html`
4. `n=60&f=0`
5. `12345`

Вопрос #3



Какой из нижеперечисленных URL'ов НЕ соответствует схеме URL?

1. `https://fake:password@amazon.co.uk/b?&node=16308412031`
2. `https://habr.com:8000/ru/company/mailru/blog/470834/#comment_20752712`
3. `http://youtu.be/qb4D17Ms1p8`
4. `http://go.mail.ru/&f=0`

Вопрос #4



У вас трёхзвенная архитектура приложения. При обращении к документу по URL сервер вернул код ответа 500 Internal server error. Что это означает?

1. Документ не найден
2. Нет доступа, нужна авторизация
3. Проблемы на сервере
4. Документ найден, но он пустой

Вопрос #5



У вас трёхзвенная архитектура приложения. При обращении к документу по URL сервер вернул код ответа 204. Что это означает?

1. Документ найден
2. Редирект
3. Проблемы на сервере
4. Документ найден, но он пустой.

Вопрос #6



Какой код ошибки соответствует, что документ перемещён (редирект)?

1. 200
2. 301
3. 500
4. 304
5. 403

Вопрос #7



Веб сервер имеет `location`, который матчится по `url-path /`, внутри стоит директива `root /home/www/;`. Приходит на него HTTP-запрос, просящий страницу `/path/to/doc.html`, по какому пути на сервере будет искаться документ?

1. `/home/www/path/to/doc.html`
2. `/home/www/doc.html`
3. `/path/to/doc.html`
4. `/root/path/to/doc.html`

Вопрос #8



Какой из методов HTTP-запроса применяется только для извлечения метаданных, валидации URL документа?

1. GET
2. POST
3. TRACE
4. HEAD
5. PUT

Вопрос #9



Что делает метод POST?

1. Используется для запроса содержимого указанного ресурса
2. Применяется для передачи пользовательских данных заданному ресурсу
3. Применяется для загрузки содержимого запроса на указанный в запросе URI
4. Удаляет указанный ресурс



ТЕХНОТРЕК

КВИЗ #4

URL: <https://forms.gle/WYFMYNCRVRNyzMVA7>



Не забудьте поставить
оценки и оставить
отзывы!!!!





ТЕХНОТРЕК

**Спасибо за
внимание!**

Антон Кухтичев

a.kukhtichev@corp.mail.ru