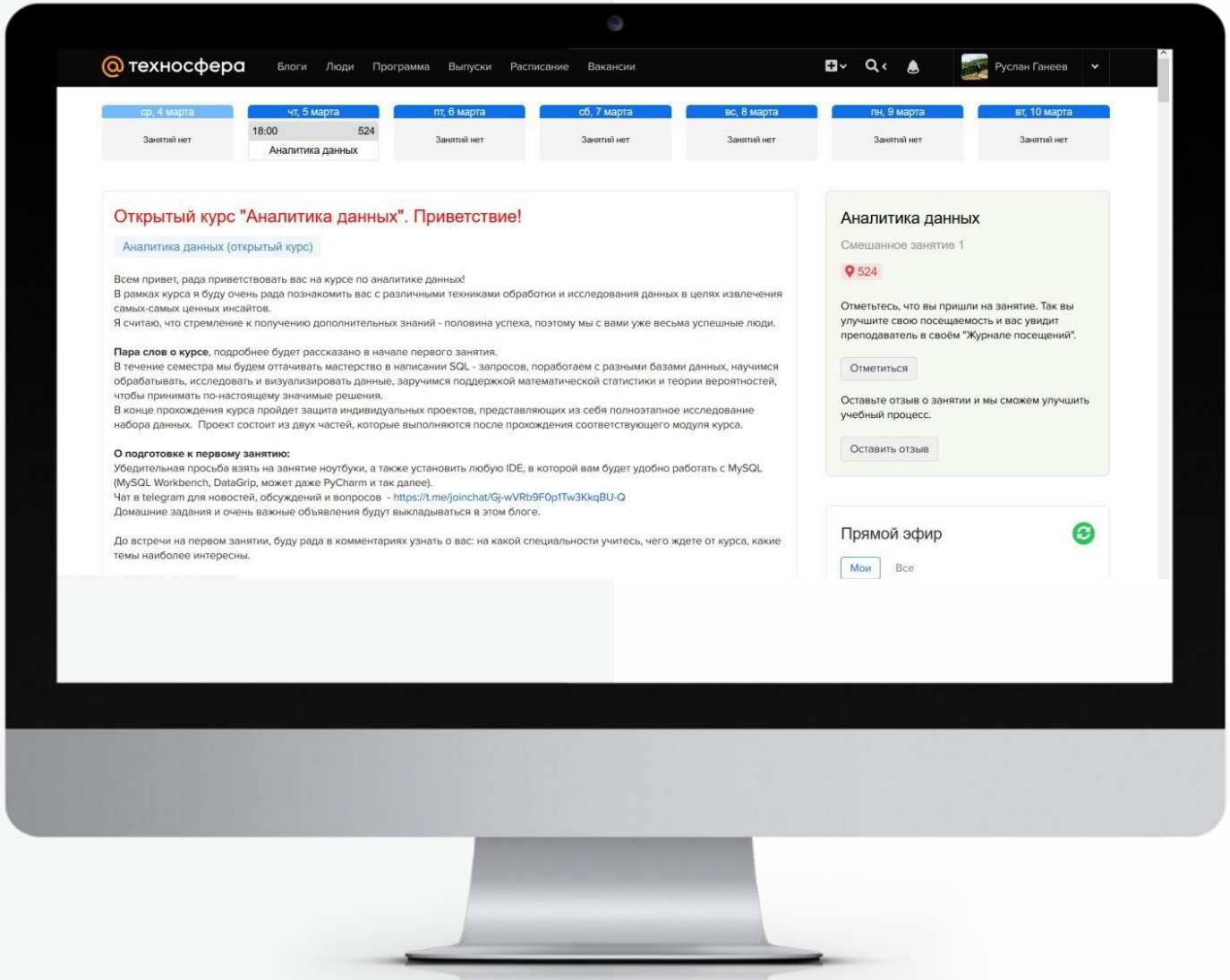
ТЕХНОСФЕРА

Введение в базы данных: PostgreSQL

Антон Кухтичев

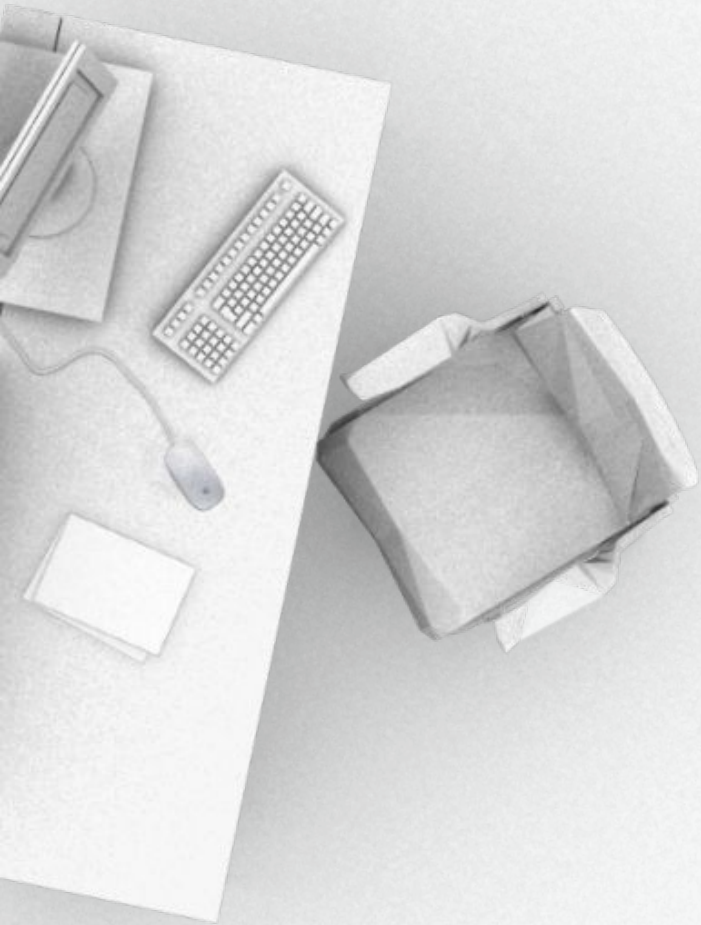
Содержание занятия

1. Квиз #4
2. Результаты квиза #3
3. PostgreSQL
4. Язык SQL
5. Django и PostgreSQL
6. Домашнее задание



Напоминание отметиться на портале

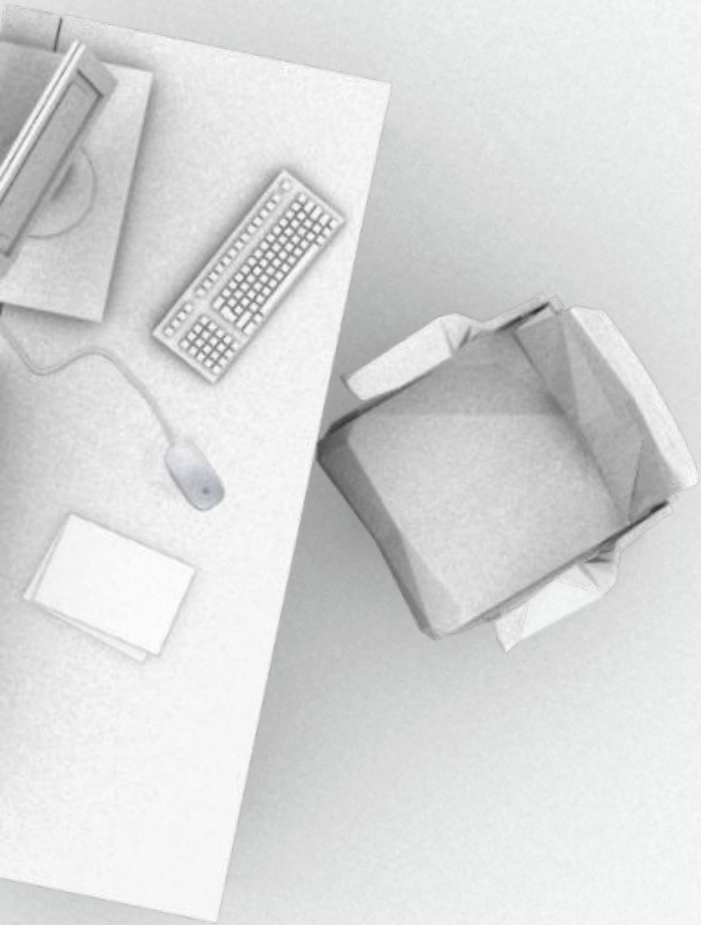
Иначе плохо всё будет.



Квиз #4

Квиз №4

https://forms.gle/BrToFH3ExSTShEL86



Результаты квиза #3

Немного статистики

Абсолютный чемпион: **Тулаева Екатерина (10/10)**

Неизвестный пользователь: **Котофей Котейков** (9/10)

- Оценка отлично (9-10 баллов): 30.30%
- Оценка хорошо (7-8 баллов): 27.27%
- Оценка удовлетворительно (5-6 баллов): 36.36%
- Оценка неудовлетворительно (<5 баллов): 6.06%

Вопрос №1

Какой стек протоколов используется в протоколе HTTP?

1) HTTP-TCP-ICMP
2) HTTP-IP-TCP
3) HTTP-UDP-IP
4) **HTTP-TCP-IP**

Вопрос №2

В чём особенность SPA - single page application - для современный веб-приложений?

1. Приложение располагается на одной статической html-страничке
2. **html-страниц сколько угодно, но данные берутся через веб-сервер**
3. Каждый запрос к веб-серверу сопровождается перезагрузкой страницы
4. html-контент генерируется веб-сервером

Вопрос №3

На каком порту работает протокол HTTP по умолчанию?

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

Вопрос №4

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 №5

Какой из нижеперечисленных URL'ов НЕ соответствует схеме URL?

1. https://youtu.be/dQw4w9WgXcQ
2. https://fake:password@amazon.co.uk/b?&node=16308412031
3. https://habr.com:8000/ru/company/mailru/blog/470834/#comment_20752712
4. **http://go.mail.ru/&f=0**

Вопрос №6

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 №7

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

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

Вопрос №8

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

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

Вопрос №9

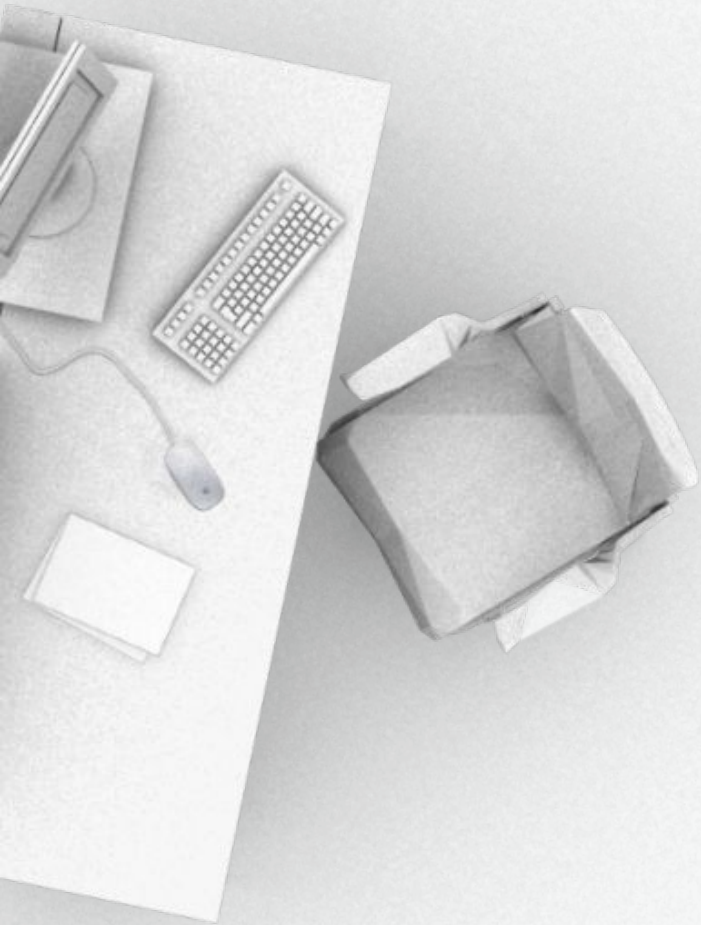
Веб сервер имеет location, который матчится по url-path /, внутри стоит директива root /home/www/;. Приходит на него HTTP-запрос, просящий страницу /path/to/doc.html, по какому пути на сервере будет искаться документ?

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

Вопрос №10

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

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

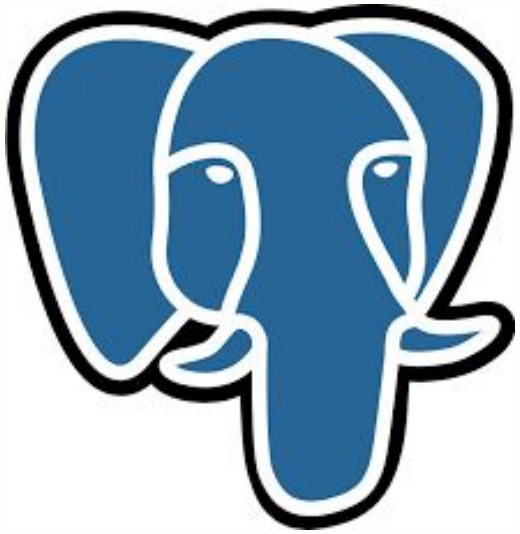


PostgreSQL

Клиент psql: установка, запуск, создание БД и новых пользователей.

Установка PostgreSQL

```
# Установка на Ubuntu
sudo apt install postgresql-10
# Установка на MacOS
brew install postgresql@10
# Проверка подключения
sudo -u <USER_NAME> psql
```



Сильные стороны PostgreSQL

- высокопроизводительные и надёжные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования;
- наследование;
- возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS;
- встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;
- расширяемость.

Пользователи и базы из коробки

Пользователи

- `postgres` — 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

Подключение через UNIX сокет, имя пользователя совпадает с пользователем Linux:

```
$ psql db_name
```

Подключение через TCP сокет:

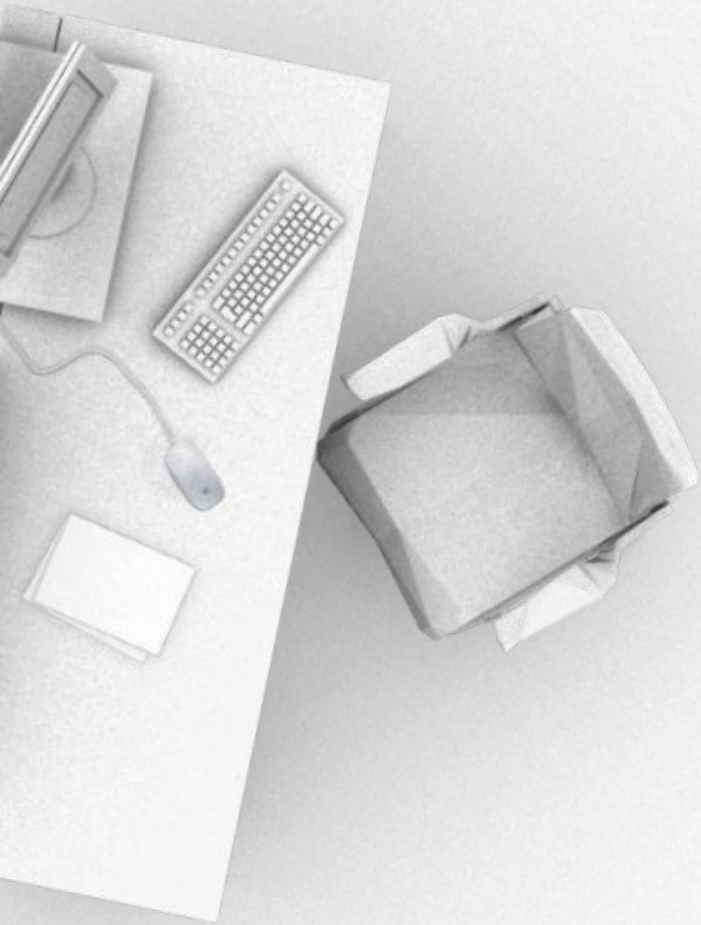
```
$ psql --host=127.0.0.1 --user=db_user db_name
```

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;
```

Язык SQL

- `\?` — 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
- `\q` — выход из psql



Язык SQL

Типы данных, синтаксис, основные виды операции.

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;
- `varchar` — строка ограниченное длины;
- `char` — строка ограниченное длины, дополненная пробелами;
- `uuid` — UUID;
- `jsonb` — JSON документ;

Виды операций

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

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

```sql
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

```sql
UPDATE users SET name = 'Не такой как все'
WHERE user_id = 2;

DELETE FROM users WHERE user_id % 2 = 0;
```

Выборка данных

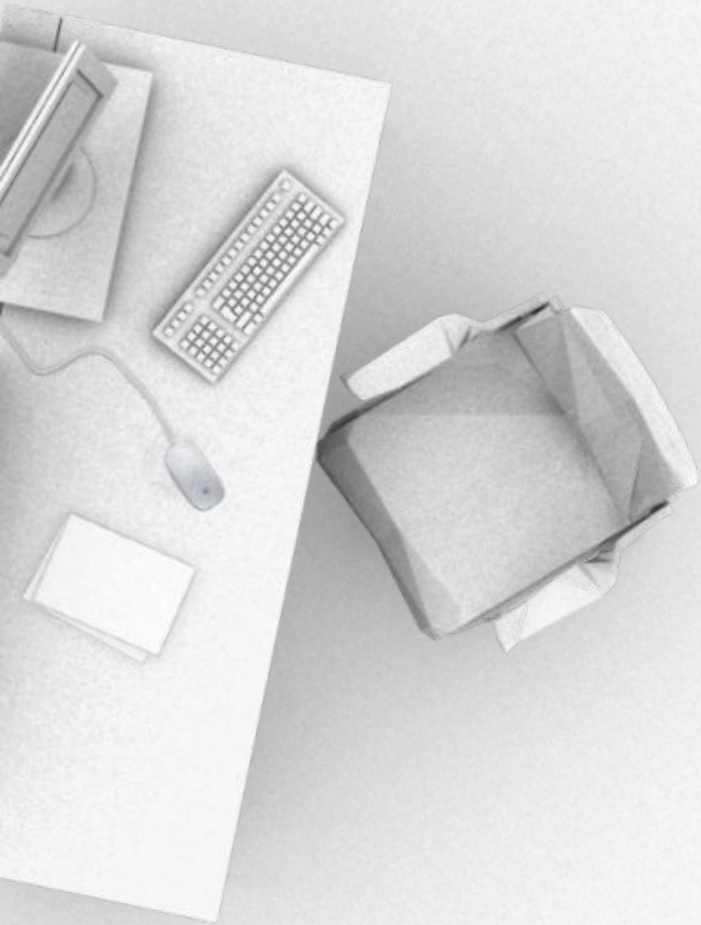
```
SELECT message_id, content, added_at::DATE
FROM messages
WHERE user_id = 3
  AND message_id < 100500
ORDER BY added_at DESC
LIMIT 10
```

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
```

I KNOW SQL

http://www.sql-ex.ru



Django и PostgreSQL

ORM, как подключить БД к Django-приложению, создание и миграция моделей.

Добавить базу данных 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',
    }
}
```

Добавить базу данных в settings.py

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

# Создаем суперпользователя
./manage.py createsuperuser

# и запускаем сервер
./manage 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` — проверть структуру моделей.

Откат до предыдущей миграции

```
# В общем случае команда выглядит так:
$ ./manage.py migrate <ваше приложение> <название предыдущей миграции>
# Откат всех миграций!
$ ./manage.py migrate <ваше приложение> zero
# Примерчик:
$ ./manage.py showmigrations movies
movies
 [X] 0001_initial
 [X] 0002_movie_genre
 [X] 0003_auto_20201005_1456
$ ./manage.py migrate movies 0002_movie_genre
```

Типы полей (1)

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

Типы полей (2)

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

Как сделать своё поле с первичным ключём

Если по каким-либо причинам Вы не хотите, чтобы создавалось поле id, то надо создать своё с `primary_key=True`!

```
class Blog(models.Model):
    my_super_pk = models.AutoField(primary_key=True)
    ...
```

Параметры у поля в модели

`verbose_name` — человекочитаемое название поля;

`null` — может ли быть поле NULL в БД, по умолчанию - False

`blank` — может ли быть поле пустым в форме, по умолчанию False

`choices` — выбор значения из списка;

`default` — значение по умолчанию;

`db_index` — нужно ли создать индекс в БД;

`help_text` — подробное описание поля;

`primary_key` —

`unique` — уникальное ли поле;

`editable` — можно ли поле изменять;

SQL-запросы напрямую из Django (1)

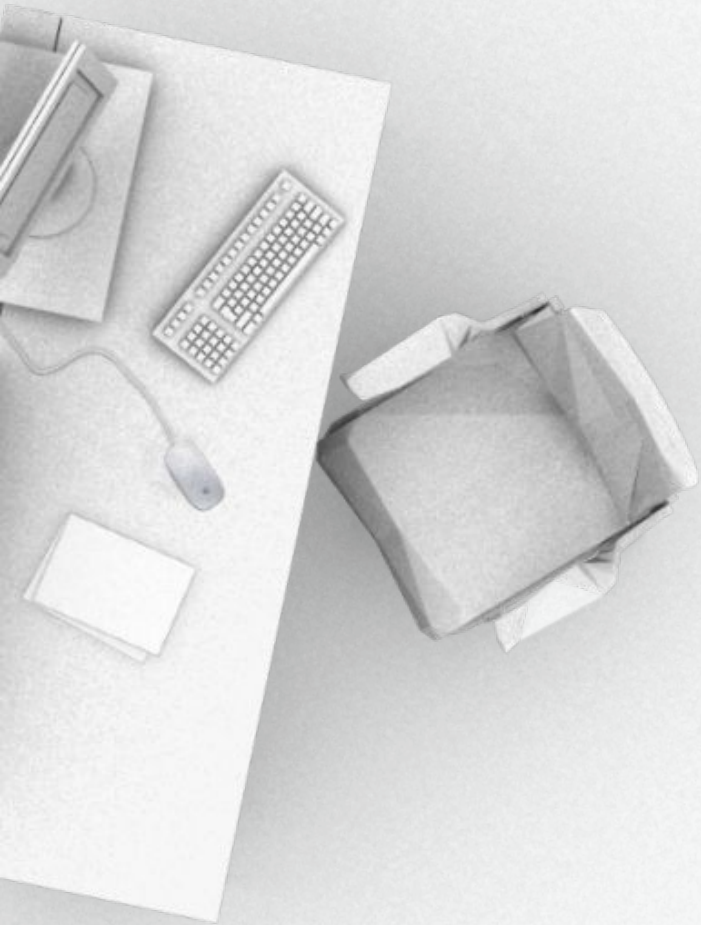
```
from django.db import connection

def my_custom_sql(self):
    with connection.cursor() as cursor:
        cursor.execute("SELECT foo FROM bar WHERE baz = %s", [self.baz])
        row = cursor.fetchone()

    return row
```

SQL-запросы напрямую из Django (2)

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



Домашнее задание

Домашнее задание #5

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

Домашнее задание по уроку #5

Домашнее задание №5

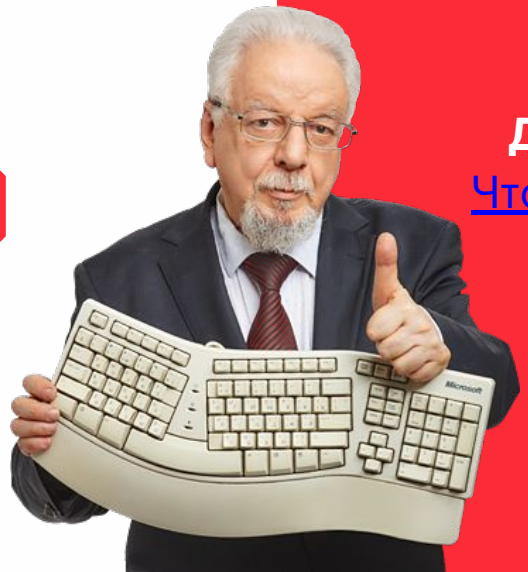
8

Баллов за задание

Сроков нет, но вы держитесь

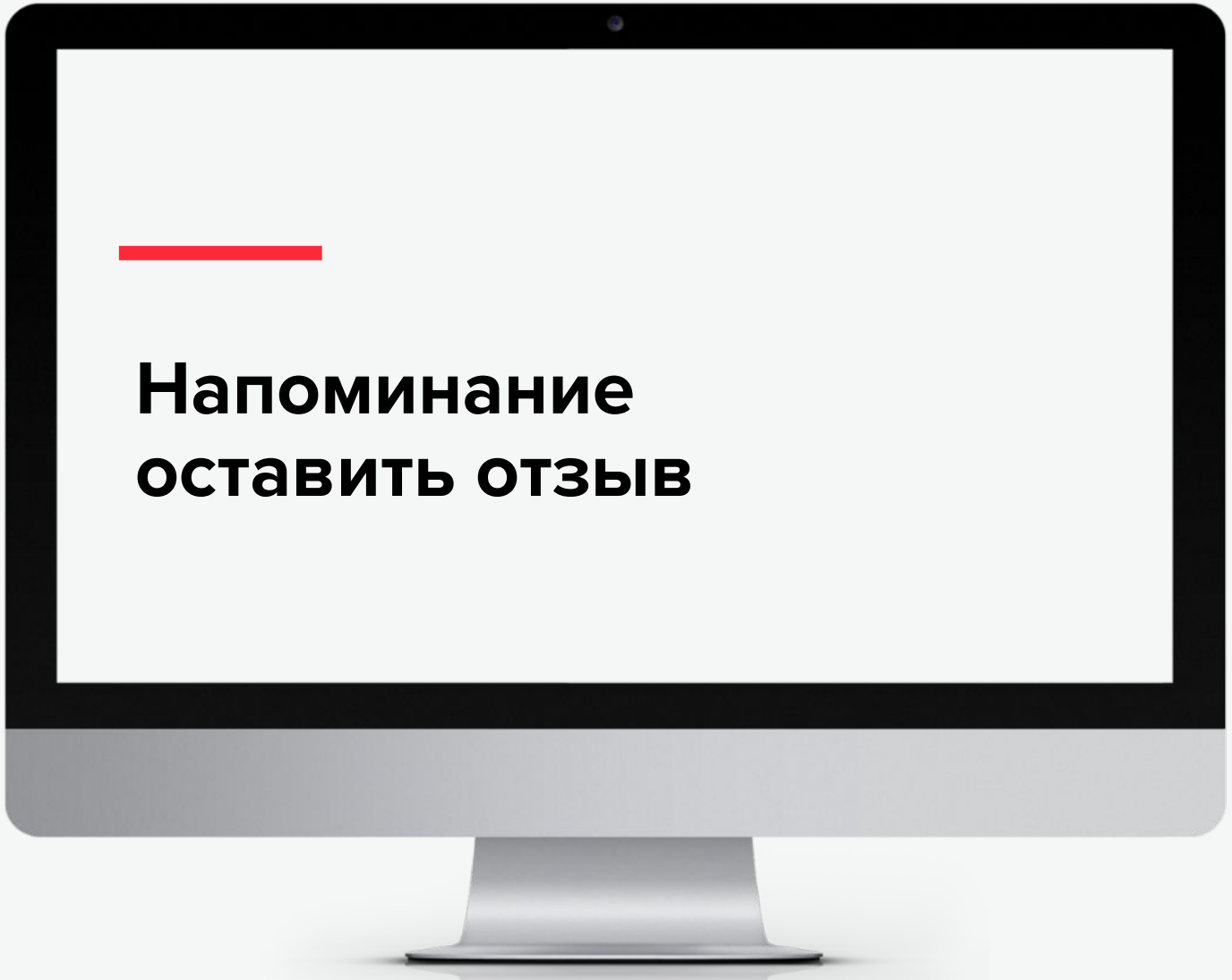
Срок сдачи

Смотри подробнее файл homework.md в репозитории с лекциями!



Для саморазвития (опционально)

Чтобы не набирать двумя пальчиками


Напоминание
оставить отзыв

СПАСИБО
ЗА ВНИМАНИЕ