

Кафедра «Систем обработки информации и управления»

Лабораторная работа №1
по курсу Постреляционные базы данных

Тема: «Работа с объектно-реляционной БД - сложные типы данных»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-22М

Сметанкин К.И.

" " _____ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.

к.т.н., доцент

" " _____ 2020 г.

Создание таблиц

Структура ФИО

```
CREATE TYPE fullname AS
(
    firstname CITEXT,
    lastname CITEXT,
    middlename CITEXT
);
```

Перечисляемый тип

```
CREATE TYPE mood AS ENUM ('sad', 'ok', 'happy');
```

Таблица пользователей

```
CREATE TABLE IF NOT EXISTS users
(
    id SERIAL PRIMARY KEY,
    login CITEXT NOT NULL UNIQUE,
    password TEXT,
    name fullname NOT NULL DEFAULT ('empty', 'empty', 'empty')::fullname,
    avatar CITEXT NOT NULL,
    karma INT DEFAULT 0,
    registered timestampz DEFAULT now(),
    mood mood DEFAULT 'ok'
);

-- проверка длины пароля
ALTER TABLE users
    ADD CHECK ( length(password) > 5 );
```

Наследование таблиц

```
CREATE TYPE super_users_level AS ENUM ('base', 'ok', 'incredible');

CREATE TABLE IF NOT EXISTS super_users
(
    level super_users_level DEFAULT 'base'
)
INHERITS (users);
```

Таблица статей

```
CREATE TABLE IF NOT EXISTS posts
(
    id          SERIAL PRIMARY KEY,
    header      CITEXT  NOT NULL UNIQUE,
    short_topic CITEXT  NOT NULL UNIQUE,
    main_topic  CITEXT  NOT NULL UNIQUE,
    user_id     int      NOT NULL REFERENCES users (id),
    authors     CITEXT[] NOT NULL,
    show        bool      DEFAULT False,
    created     timestampz DEFAULT now()
);
```

Таблица тэгов статей

```
CREATE TABLE IF NOT EXISTS tag
(
    id      SERIAL PRIMARY KEY,
    name    CITEXT NOT NULL
);
```

Таблица комментариев к статьям

```
CREATE TABLE IF NOT EXISTS comments
(
    id          SERIAL PRIMARY KEY,
    parent_id   int REFERENCES comments (id),
    user_id     int NOT NULL REFERENCES users (id),
    post_id     int NOT NULL REFERENCES posts (id),
    payload     text NOT NULL,
    show        bool      DEFAULT True,
    created     timestampz DEFAULT now()
);
```

Работа с данными

Создание записей с вложенными структурами

```
insert into posts (header, short_topic, main_topic, user_id, authors)
values ('test_header', 'short', 'main', 1, '{"me", "my friend 1", "my friend 2"}');
insert into posts (header, short_topic, main_topic, user_id, authors)
values ('test_header_1', 'short_1', 'main_1', 3, '{"me", "my friend 1", "my friend 2"}');
insert into posts (header, short_topic, main_topic, user_id, authors)
values ('test_header_2', 'short_2', 'main_2', 1, '{"me", "my friend 1", "my friend 2"}');
```

Обновление сачти массива

```
update posts
set authors[1:2] = '{"me too", "second me", "third me"}'
where header = 'test_header';
```

```
update posts
set authors = array_remove(authors, 'me too');
```

Вывод всех элементов, удовлетворяющих условию

```
select *
from posts
      join users u on posts.user_id = u.id
where 'me too' = ANY (authors);
```

composite types

```
select *
from users;
```

```
select (name::fullname).firstname
from users;
```

```
update users
set name.lastname = 'my_last_name'
where id = 1;
```

```
update users
set name = ('kek', 'kek', 'kek')::fullname
where id = 1;
```

enum

```
update users
SET mood = 'kek'
where id = 1;
```

```
update users
SET mood = 'sad'
where id = 1;
```

```
select *
from users;
```

Работа с наследованием

```
update users
SET mood = 'kek'
where id = 1;
```

```
update users
SET mood = 'sad'
where id = 1;
```

```
select *
from users;
```

```
-- inherit
insert into super_users (login, password, name, avatar, karma, mood)
VALUES ('super', 'super_pass', ('super-kek', 's', 'a')::fullname, '', 0, 'ok');
```

```
select *
from users;
```

```
select *
from ONLY users;
```

```
select *
from super_users;
```