Практическая работа №3 (кол-во часов 16)

Задание: разработать систему электронного голосования. Реализовать

пользовательскую модель. При входе для пользователя с определенной

ролью открывается меню, перечисленное в требованиях. Сохранение и

выгрузку результатов реализовать согласно требованиям к программе.

Требования к программе:

Роль: Администратор

1. Просмотр список пользователей и удаление

2. Просмотр и удаление Центральных избирательных комиссий (ЦИК)

3. Создание ЦИК - логин:пароль

4. Просмотр и удаление кандидатов

Роль: ЦИК

1. Создание голосования с окончанием при наступлении определенной

даты

2. Добавление кандидатов - логин: пароль

3. Печать результатов (PDF)

4. Выбор группировки результатов

5. Сортировка результатов

Роль: Кандидат

1. Заполнение данных о себе

2. Результаты предыдущего голосования

3. Все голосования, в которых принимал участие

Роль: Пользователь

1. Регистрация - по ФИО + дата рождения И/ИЛИ СНИЛС И/ИЛИ какие-то

еще уникальные данные

2. Голосование

3. Просмотр списка кандидатов

4. Все голосования, в которых голосовал/неголосовал

Требования к сохранению результатов:

1. Выгрузка должна осуществляться в рамках одного или нескольких

голосований.

2. При выгрузке нескольких голосований пользователю должен

задаваться вопрос о выгрузке одним файлом или по принципу

1 файл = 1 голосование.

│── config # Конфигурация базы данных (JDBC)

│── dao

│── model # Модели данных (POJO-классы)

│── service # Бизнес-логика

│── ui # Пользовательский интерфейс (консоль или Swing/JavaFX)

│── util # Вспомогательные утилиты

│── exception создать в основе этой логики заход как администратор

-- Таблица Администраторы

CREATE TABLE administrators (

id SERIAL PRIMARY KEY,

full\_name VARCHAR(255) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

phone VARCHAR(11) UNIQUE NOT NULL,

login VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL

);

-- Таблица ЦИК (Центральные Избирательные Комиссии)

CREATE TABLE election\_commissions (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

region VARCHAR(255) NOT NULL,

address TEXT NOT NULL,

contact\_email VARCHAR(255) UNIQUE NOT NULL,

contact\_phone VARCHAR(11) UNIQUE NOT NULL,

login VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL

);

-- Таблица Кандидатов

CREATE TABLE candidates (

id SERIAL PRIMARY KEY,

full\_name VARCHAR(255) NOT NULL,

birth\_date DATE NOT NULL,

party VARCHAR(255),

biography TEXT,

login VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

disqualified BOOLEAN DEFAULT FALSE -- Если TRUE, кандидат больше не может участвовать

);

-- Таблица Пользователи

CREATE TABLE users (

id SERIAL PRIMARY KEY,

full\_name VARCHAR(255) NOT NULL,

birth\_date DATE NOT NULL,

passport\_data VARCHAR(50) UNIQUE NOT NULL, -- Серия и номер паспорта в одном поле

login VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL

);

-- Таблица Голосования (Основная)

CREATE TABLE elections (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

commission\_id INT NOT NULL, -- Пока просто INT

start\_time TIMESTAMP NOT NULL DEFAULT NOW(),

end\_time TIMESTAMP NOT NULL DEFAULT (NOW() + INTERVAL '1 day'),

UNIQUE (name, start\_time) -- Запрещаем дублирование выборов с одинаковым названием

);;

История голосования

CREATE TABLE votes\_log (

id SERIAL PRIMARY KEY,

election\_id INT NOT NULL,

voter\_login VARCHAR(100) NOT NULL,

candidate\_login VARCHAR(100) NOT NULL, -- За кого он проголосовал

vote\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Когда проголосовал

UNIQUE (election\_id, voter\_login), -- Только один голос от пользователя в конкретном голосовании

FOREIGN KEY (election\_id) REFERENCES elections(id) ON DELETE CASCADE,

FOREIGN KEY (voter\_login) REFERENCES users(login) ON DELETE CASCADE,

FOREIGN KEY (candidate\_login) REFERENCES candidates(login) ON DELETE CASCADE

);

CREATE TABLE election\_results (

id SERIAL PRIMARY KEY,

election\_id INT NOT NULL,

candidate\_login VARCHAR(100) NOT NULL,

votes INT NOT NULL DEFAULT 0,

FOREIGN KEY (election\_id) REFERENCES elections(id) ON DELETE CASCADE,

FOREIGN KEY (candidate\_login) REFERENCES candidates(login) ON DELETE CASCADE,

UNIQUE (election\_id, candidate\_login)

);

String tableName = "elections\_" + electionId;

String createTableSQL = String.format(

"CREATE TABLE IF NOT EXISTS %s (" +

"id SERIAL PRIMARY KEY, " +

"login VARCHAR(100) NOT NULL, " + // Логин кандидата вместо user\_id

"votes INT NOT NULL DEFAULT 0, " +

"FOREIGN KEY (login) REFERENCES candidates(login) ON DELETE CASCADE)", // Связь с таблицей кандидатов

tableName

);

GRANT SELECT, INSERT, UPDATE, DELETE ON votes\_log TO username;

DELETE FROM имя\_таблицы WHERE id IN (1, 2, 3);

SHOW data\_directory; --посмотреть директорию БД

SELECT oid, datname FROM pg\_database WHERE datname = 'electronic\_voting\_system'; --посмотреть код и название в директории БД

Docker

docker-compose up –d (запуск docker-compose)

psql -h localhost -U username -d electronic\_voting\_system (для захода к БД в Docker на локальной машине)