

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет инфокоммуникационных технологий

Направление: Интеллектуальные системы в гуманитарной сфере

ОТЧЕТ

О Лабораторной работе №3

по теме: «Создание базы данных средствами реляционной СУБД. Генераторы схем в СУБД»

по дисциплине: Проектирование и реализация баз данных

Выполнил:

студент группы К3239

Котовщиков Андрей Романович

Проверил:

Говорова М. М. _____

Дата: “ “ октября 2023 года

Оценка _____

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

Практическое задание:

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД.

Указание:

Создать две резервные копии:

- с расширением *CUSTOM* для восстановления БД;
 - с расширением *PLAIN* для листинга (в отчете);
 - при создании резервных копий БД настроить параметры *Dump options* для *Type of objects* и *Queries*.
7. Восстановить БД.

Выполнение

Название создаваемой БД: «Распределение аудиторного фонда»

Создание базы данных с помощью SQL:

1. Создание базы данных с указанием кодировки и локали с пользователем «postgres» в качестве владельца:

```
CREATE DATABASE lab_3_2 WITH TEMPLATE = template0 ENCODING = 'UTF8'
LOCALE_PROVIDER = libc LOCALE = 'Russian_Russia.1251';
```

```
ALTER DATABASE lab_3_2 OWNER TO postgres;
```

2. Создание схемы (пространства имен) для базы данных с пользователем «postgres» в качестве владельца:

```
CREATE SCHEMA lab_3_2;
```

```
ALTER SCHEMA lab_3_2 OWNER TO postgres;
```

3. Создание собственных Enum типов:

```
CREATE TYPE lab_3_2.branch_city AS ENUM (
```

```
'Москва',
```

```
'Санкт-Петербург',
```

```
'Самара',
```

```
'Сочи',
```

```
'Анапа'
```

```
);
```

```
ALTER TYPE lab_3_2.branch_city OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.classroom_type AS ENUM (
```

```
'Лабораторная',
```

```
'Лекционная',
```

```
'Практическая'
```

```
);
```

```
ALTER TYPE lab_3_2.classroom_type OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.curriculum_status AS ENUM (
```

```
'Действующая',  
  
'Недействующая',  
  
'В разработке'  
  
);
```

```
ALTER TYPE lab_3_2.curriculum_status OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.schedule_week_day AS ENUM (  
  
    'пн',  
  
    'вт',  
  
    'ср',  
  
    'чт',  
  
    'пт',  
  
    'сб',  
  
    'вс'  
  
);
```

```
ALTER TYPE lab_3_2.schedule_week_day OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.student_in_group_status AS ENUM (
```

```
    'Отчислен',
```

```
    'Учится',
```

```
    'Академ'
```

```
);
```

```
ALTER TYPE lab_3_2.student_in_group_status OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.subject_attestation_type AS ENUM (
```

```
    'Экзамен',
```

```
    'Зачет',
```

```
    'Диф Зачет'
```

```
);
```

```
ALTER TYPE lab_3_2.subject_attestation_type OWNER TO postgres;
```

```
CREATE TYPE lab_3_2.subject_format AS ENUM (
```

```
    'Очно',
```

```
'Дистанционно',  
  
'Очно-Дистанционно'  
  
);
```

4. Создание всех таблиц с необходимыми ограничениями:

```
CREATE TABLE lab_3_2.branch (  
  
    id integer NOT NULL,  
  
    phone_number character varying(14) NOT NULL,  
  
    city lab_3_2.branch_city NOT NULL,  
  
    CONSTRAINT branch_phone_number_check CHECK (((phone_number)::text ~  
'^\\+?(\\d{1,3})?[- .]?\\((?:\\d{2,3})\\)?[- .]?\\d\\d\\d[- .]?\\d\\d\\d\\d$'::text))  
  
);
```

```
ALTER TABLE lab_3_2.branch OWNER TO postgres;
```

```
CREATE SEQUENCE lab_3_2.branch_id_seq  
  
    AS integer  
  
    START WITH 1  
  
    INCREMENT BY 1  
  
    NO MINVALUE  
  
    NO MAXVALUE
```

CACHE 1;

ALTER TABLE lab_3_2.branch_id_seq OWNER TO postgres;

ALTER SEQUENCE lab_3_2.branch_id_seq OWNED BY lab_3_2.branch.id;

CREATE TABLE lab_3_2.branch_site (

 id integer NOT NULL,

 address character varying(70) NOT NULL,

 branch_id integer

);

ALTER TABLE lab_3_2.branch_site OWNER TO postgres;

CREATE SEQUENCE lab_3_2.branch_site_branch_id_seq

 AS integer

 START WITH 1

 INCREMENT BY 1

 NO MINVALUE

 NO MAXVALUE

CACHE 1;

ALTER TABLE lab_3_2.branch_site_branch_id_seq OWNER TO postgres;

ALTER SEQUENCE lab_3_2.branch_site_branch_id_seq OWNED BY
lab_3_2.branch_site.branch_id;

CREATE SEQUENCE lab_3_2.branch_site_id_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE lab_3_2.branch_site_id_seq OWNER TO postgres;

ALTER SEQUENCE lab_3_2.branch_site_id_seq OWNED BY lab_3_2.branch_site.id

CREATE TABLE lab_3_2.classroom (

code character varying(15) NOT NULL,

```
number character varying(15) NOT NULL,  
  
type lab_3_2.classroom_type NOT NULL,  
  
capacity smallint NOT NULL,  
  
branch_site_id integer NOT NULL,  
  
CONSTRAINT classroom_capacity_check CHECK ((capacity > 0))  
  
);
```

```
ALTER TABLE lab_3_2.classroom OWNER TO postgres;
```

```
CREATE TABLE lab_3_2.curriculum (  
  
    code character varying(15) NOT NULL,  
  
    name character varying(70) NOT NULL,  
  
    description text NOT NULL,  
  
    status lab_3_2.curriculum_status NOT NULL,  
  
    certificate character varying(50) NOT NULL  
  
);
```

```
ALTER TABLE lab_3_2.curriculum OWNER TO postgres;
```

```
CREATE TABLE lab_3_2.curriculum_subject (  
  
    id integer NOT NULL,  
  
    implemented_curriculum_id integer NOT NULL,  
  
    subject_code character varying(15) NOT NULL,  
  
    term numeric(1,0) NOT NULL,  
  
    CONSTRAINT curriculum_subject_term_check CHECK ((term > (0)::numeric))  
  
);
```

```
ALTER TABLE lab_3_2.curriculum_subject OWNER TO postgres;
```

```
CREATE SEQUENCE lab_3_2.curriculum_subject_id_seq  
  
    AS integer  
  
    START WITH 1  
  
    INCREMENT BY 1  
  
    NO MINVALUE  
  
    NO MAXVALUE  
  
    CACHE 1;
```

```
ALTER TABLE lab_3_2.curriculum_subject_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE lab_3_2.curriculum_subject_id_seq OWNED BY  
lab_3_2.curriculum_subject.id;
```

```
CREATE TABLE lab_3_2."group" (  
  
    name character varying(20) NOT NULL,  
  
    start_date date,  
  
    end_date date,  
  
    implemented_curriculum_id integer NOT NULL,  
  
    CONSTRAINT group_check CHECK ((end_date > start_date)),  
  
    CONSTRAINT group_end_date_check CHECK (((end_date)::text ~ '^d{4}-d{2}-  
d{2}$'::text)),  
  
    CONSTRAINT group_start_date_check CHECK (((start_date)::text ~ '^d{4}-d{2}-  
d{2}$'::text))  
  
);
```

```
ALTER TABLE lab_3_2."group" OWNER TO postgres;
```

```

CREATE TABLE lab_3_2.implemented_curriculum (

    id integer NOT NULL,

    qbp integer NOT NULL,

    start_date date NOT NULL,

    end_date date NOT NULL,

    branch_id integer,

    curriculum_code character varying(15) NOT NULL,

    CONSTRAINT implemented_curriculum_check CHECK ((end_date > start_date)),

    CONSTRAINT implemented_curriculum_end_date_check CHECK (((end_date)::text ~
'^\d{4}-\d{2}-\d{2}$'::text)),

    CONSTRAINT implemented_curriculum_qbp_check CHECK ((qbp >= 0)),

    CONSTRAINT implemented_curriculum_start_date_check CHECK (((start_date)::text ~
'^\d{4}-\d{2}-\d{2}$'::text))

);

```

```

ALTER TABLE lab_3_2.implemented_curriculum OWNER TO postgres;

```

```

CREATE SEQUENCE lab_3_2.implemented_curriculum_id_seq

    AS integer

    START WITH 1

```

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE lab_3_2.implemented_curriculum_id_seq OWNER TO postgres;

ALTER SEQUENCE lab_3_2.implemented_curriculum_id_seq OWNED BY
lab_3_2.implemented_curriculum.id;

CREATE TABLE lab_3_2.schedule (

id integer NOT NULL,

week_day lab_3_2.schedule_week_day NOT NULL,

is_even boolean NOT NULL,

datetime timestamp without time zone NOT NULL,

classroom_code character varying(15),

curriculum_subject_id integer NOT NULL,

teacher_id integer

);

```
ALTER TABLE lab_3_2.schedule OWNER TO postgres;
```

```
CREATE SEQUENCE lab_3_2.schedule_id_seq
```

```
AS integer
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NO MINVALUE
```

```
NO MAXVALUE
```

```
CACHE 1;
```

```
ALTER TABLE lab_3_2.schedule_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE lab_3_2.schedule_id_seq OWNED BY lab_3_2.schedule.id;
```

```
CREATE TABLE lab_3_2.student (
```

```
id integer NOT NULL,
```

```
name character varying(30) NOT NULL,
```

```
surname character varying(30) NOT NULL,
```

```

phone_number character varying(14) NOT NULL,

email character varying(50),

passport numeric(10,0) NOT NULL,

CONSTRAINT student_email_check CHECK (((email)::text ~ '^[a-z0-9._%+-]+@[a-z0-9-]+\.[a-z]{2,4}$'::text)),

CONSTRAINT student_passport_check CHECK ((length((passport)::text) = 10)),

CONSTRAINT student_passport_check1 CHECK ((passport > (0)::numeric)),

CONSTRAINT student_phone_number_check CHECK (((phone_number)::text ~
'^\+?(\d{1,3})?[-.]?(\d{2,3})\d{2,3}[-.]?\d\d\d[-.]?\d\d\d\d$'::text))

);

```

```

ALTER TABLE lab_3_2.student OWNER TO postgres;

```

```

CREATE SEQUENCE lab_3_2.student_id_seq

```

```

AS integer

```

```

START WITH 1

```

```

INCREMENT BY 1

```

```

NO MINVALUE

```

```

NO MAXVALUE

```

```

CACHE 1;

```



```
ALTER TABLE lab_3_2.student_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE lab_3_2.student_id_seq OWNED BY lab_3_2.student.id;
```

```
CREATE TABLE lab_3_2.student_in_group (  
  
    code character varying(15) NOT NULL,  
  
    status lab_3_2.student_in_group_status NOT NULL,  
  
    start_date date NOT NULL,  
  
    end_date date NOT NULL,  
  
    student_id integer NOT NULL,  
  
    group_name character varying(20),  
  
    CONSTRAINT student_in_group_end_date_check CHECK (((end_date)::text ~ '^d{4}-  
d{2}-d{2}$'::text)),  
  
    CONSTRAINT student_in_group_start_date_check CHECK (((start_date)::text ~  
'^d{4}-d{2}-d{2}$'::text))  
  
);
```

```
ALTER TABLE lab_3_2.student_in_group OWNER TO postgres;
```

```
CREATE TABLE lab_3_2.subject (  
  
    code character varying(15) NOT NULL,  
  
    name character varying(50) NOT NULL,  
  
    format lab_3_2.subject_format NOT NULL,  
  
    attestation_type lab_3_2.subject_attestation_type NOT NULL,  
  
    practical_hours smallint NOT NULL,  
  
    lecture_hours smallint NOT NULL,  
  
    laboratory_hours smallint NOT NULL,  
  
    consultation_hours smallint NOT NULL,  
  
    CONSTRAINT subject_practical_hours_check CHECK ((practical_hours > 0)),  
  
    CONSTRAINT subject_practical_hours_check1 CHECK ((practical_hours > 0)),  
  
    CONSTRAINT subject_practical_hours_check2 CHECK ((practical_hours > 0)),  
  
    CONSTRAINT subject_practical_hours_check3 CHECK ((practical_hours > 0))  
  
);
```

```
ALTER TABLE lab_3_2.subject OWNER TO postgres;
```

```
CREATE TABLE lab_3_2.subject_in_certificate (  

```

```
id integer NOT NULL,

mark smallint,

attempt smallint DEFAULT 0 NOT NULL,

date date NOT NULL,

curriculum_subject_id integer NOT NULL,

teacher_id integer NOT NULL,

student_in_group_code character varying(15) NOT NULL,

CONSTRAINT subject_in_certificate_attempt_check CHECK ((attempt > 0)),

CONSTRAINT subject_in_certificate_date_check CHECK (((date)::text ~ '^\\d{4}-\\d{2}-\\d{2}$'::text)),

CONSTRAINT subject_in_certificate_mark_check CHECK ((mark > 0))

);
```

```
ALTER TABLE lab_3_2.subject_in_certificate OWNER TO postgres;
```

```
CREATE SEQUENCE lab_3_2.subject_in_certificate_id_seq
```

```
AS integer
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NO MINVALUE
```

NO MAXVALUE

CACHE 1;

ALTER TABLE lab_3_2.subject_in_certificate_id_seq OWNER TO postgres;

ALTER SEQUENCE lab_3_2.subject_in_certificate_id_seq OWNED BY
lab_3_2.subject_in_certificate.id;

CREATE TABLE lab_3_2.teacher (

id integer NOT NULL,

name character varying(30) NOT NULL,

surname character varying(30) NOT NULL,

patronymic character varying(30) NOT NULL,

"position" character varying(50) NOT NULL,

branch_id integer NOT NULL

);

ALTER TABLE lab_3_2.teacher OWNER TO postgres;

```
CREATE SEQUENCE lab_3_2.teacher_id_seq
```

```
AS integer
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NO MINVALUE
```

```
NO MAXVALUE
```

```
CACHE 1;
```

```
ALTER TABLE lab_3_2.teacher_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE lab_3_2.teacher_id_seq OWNED BY lab_3_2.teacher.id;
```

```
ALTER TABLE ONLY lab_3_2.branch ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.branch_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.branch_site ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.branch_site_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.branch_site ALTER COLUMN branch_id SET DEFAULT  
nextval('lab_3_2.branch_site_branch_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.curriculum_subject ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.curriculum_subject_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.implemented_curriculum ALTER COLUMN id SET  
DEFAULT nextval('lab_3_2.implemented_curriculum_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.schedule ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.schedule_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.student ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.student_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.subject_in_certificate ALTER COLUMN id SET  
DEFAULT nextval('lab_3_2.subject_in_certificate_id_seq'::regclass);
```

```
ALTER TABLE ONLY lab_3_2.teacher ALTER COLUMN id SET DEFAULT  
nextval('lab_3_2.teacher_id_seq'::regclass);
```

5. Вставка данных в созданные таблицы:

```
INSERT INTO lab_3_2.branch VALUES (173, '+8822223535', 'Москва');
```

```
INSERT INTO lab_3_2.branch VALUES (174, '+8899223535', 'Санкт-Петербург');
```

```
INSERT INTO lab_3_2.branch VALUES (175, '+8818223535', 'Самара');
```

```
INSERT INTO lab_3_2.branch VALUES (176, '+88168223535', 'Сочи');
```

```
INSERT INTO lab_3_2.branch VALUES (177, '+88001668872', 'Анапа');
```

...

Выводы

В рамках данной лабораторной работы была создана полноценная база данных со всеми необходимыми ограничениями и проверками для поддержания целостности данных. Были установлены «UNIQUE CONSTRAINTS» для поддержания уникальности некоторых атрибутов (например серии и номера паспорта и номера телефона), добавлены «CHECK CONSTRAINTS» для валидации и проверки, а также созданы собственные Enum типы для указаний ограниченного набора значений для атрибутов. Были созданные две резервные копии (Custom и Plain) для полного восстановления базы данных со всей хранящейся в ней информацией.