

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»

Отчет

по лабораторной работе «**Установка СУБД PostgreSQL. Создание базы
данных в PGAdmin. Создание таблиц базы данных PostgreSQL.
Заполнение таблиц рабочими данными**»
по дисциплине «Базы данных»

Автор:

Пырков Владислав

Факультет:
Инфокоммуникацио
нных технологий
(ИКТ)

Группа: K32402

Преподаватель:
Говорова М. М.

Санкт-Петербург

2023 г.

Цель работы 1.1: овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4.

Практическое задание 1.1:

1. Установить СУБД PostgreSQL 1X.
2. Создать базу данных с использованием pgadmin 4.

Цель работы 1.2: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Практическое задание 1.2:

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

Наименование БД: Courses

ERD диаграмма:

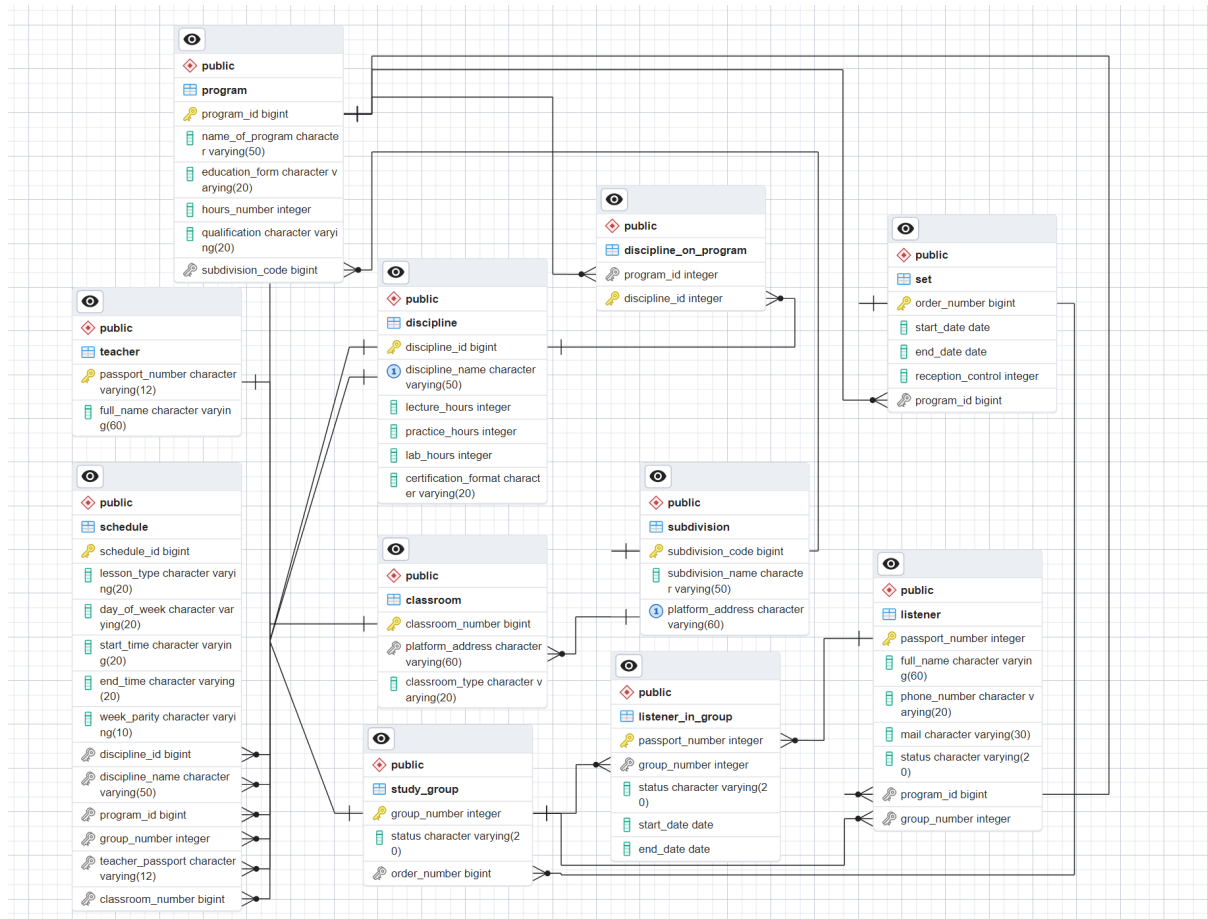


Рисунок 1 - ERD диаграмма

Dump БД: dump базы данных в двух вариантах, сгенерированный в pgAdmin, приложен к отчету. Файлы: backup_custom.sql, backup_plain.sql

Скрипты работы с БД для создания таблиц:

CREATE TABLE IF NOT EXISTS public.teacher

(

passport_number character varying(12) NOT NULL,

```
full_name character varying(60) COLLATE pg_catalog."default" NOT NULL,
```

```
CONSTRAINT teacher_pkey PRIMARY KEY (passport_number)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS public.study_group
```

```
(
```

```
group_number integer NOT NULL GENERATED ALWAYS AS IDENTITY (  
INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE  
1 ),
```

```
status character varying(20) COLLATE pg_catalog."default",
```

```
order_number bigint,
```

```
CONSTRAINT study_group_pkey PRIMARY KEY (group_number),
```

```
CONSTRAINT study_group_order_number_fkey FOREIGN KEY  
(order_number)
```

```
REFERENCES public.set (order_number) MATCH SIMPLE
```

```
ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION
```

```
);
```

```
CREATE TABLE IF NOT EXISTS public.set
```

```
(
```

```
order_number bigint NOT NULL GENERATED ALWAYS AS IDENTITY (  
INCREMENT 1 START 1 MINVALUE 1 MAXVALUE  
9223372036854775807 CACHE 1 ),
```

```
start_date date NOT NULL,
```

```
end_date date NOT NULL,
```

```

reception_control integer NOT NULL,

program_id bigint,

CONSTRAINT set_pkey PRIMARY KEY (order_number),

CONSTRAINT set_program_id_fkey FOREIGN KEY (program_id)

REFERENCES public.program (program_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT set_check CHECK (start_date <= end_date)

);

CREATE TABLE IF NOT EXISTS public.program

(

    program_id bigint NOT NULL GENERATED ALWAYS AS IDENTITY (

INCREMENT 1 START 1 MINVALUE 1 MAXVALUE

9223372036854775807 CACHE 1 ),

    name_of_program character varying(50) COLLATE pg_catalog."default"

NOT NULL,

    education_form character varying(20) COLLATE pg_catalog."default" NOT

NULL,

    hours_number integer NOT NULL,

    qualification character varying(20) COLLATE pg_catalog."default",

    subdivision_code bigint,

    CONSTRAINT program_pkey PRIMARY KEY (program_id),

    CONSTRAINT program_subdivision_code_fkey FOREIGN KEY

(subdivision_code)

REFERENCES public.subdivision (subdivision_code) MATCH SIMPLE

```

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT program_education_form_check CHECK
(education_form::text = ANY (ARRAY['distance'::character varying,
'full-time'::character varying, 'full-time distance'::character varying]::text[]))

);

CREATE TABLE IF NOT EXISTS public.subdivision

(

subdivision_code bigint NOT NULL GENERATED ALWAYS AS
IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE
9223372036854775807 CACHE 1),

subdivision_name character varying(50) COLLATE pg_catalog."default"
NOT NULL,

platform_address character varying(60) UNIQUE COLLATE
pg_catalog."default" NOT NULL,

CONSTRAINT subdivision_pkey PRIMARY KEY (subdivision_code)

);

CREATE TABLE IF NOT EXISTS public.classroom

(

classroom_number bigint NOT NULL,

platform_address character varying(60) NOT NULL,

classroom_type character varying(20) COLLATE pg_catalog."default" NOT
NULL,

CONSTRAINT classroom_pkey PRIMARY KEY (classroom_number),

```
CONSTRAINT classroom_classroom_type_check CHECK
(classroom_type::text = ANY (ARRAY['practice'::character varying,
'lab'::character varying, 'lecture'::character varying]::text[])),
```

```
CONSTRAINT platform_address_subdivision FOREIGN KEY
(platform_address)
```

```
REFERENCES public.subdivision(platform_address) MATCH SIMPLE ON
UPDATE NO ACTION ON DELETE NO ACTION
```

```
);
```

```
CREATE TABLE IF NOT EXISTS public.discipline
```

```
(
```

```
discipline_id bigint NOT NULL GENERATED ALWAYS AS IDENTITY (
INCREMENT 1 START 1 MINVALUE 1 MAXVALUE
9223372036854775807 CACHE 1 ),
```

```
discipline_name character varying(50) COLLATE pg_catalog."default" NOT
NULL,
```

```
lecture_hours integer NOT NULL,
```

```
practice_hours integer NOT NULL,
```

```
lab_hours integer NOT NULL,
```

```
certification_format character varying(20) COLLATE pg_catalog."default"
NOT NULL,
```

```
CONSTRAINT discipline_pkey PRIMARY KEY (discipline_id),
```

```
CONSTRAINT discipline_discipline_name_key UNIQUE
(discipline_name),
```

```
CONSTRAINT discipline_lab_hours_check CHECK (lab_hours >= 0),
```

```
CONSTRAINT discipline_lecture_hours_check CHECK (lecture_hours >=
0),
```

```

    CONSTRAINT discipline_practice_hours_check CHECK (practice_hours >=
0)
);

CREATE TABLE IF NOT EXISTS public.listener
(
    passport_number integer NOT NULL,

    full_name character varying(60) COLLATE pg_catalog."default" NOT
NULL,

    phone_number character varying(20) COLLATE pg_catalog."default",
    mail character varying(30) COLLATE pg_catalog."default",
    status character varying(20) COLLATE pg_catalog."default" NOT NULL,
    program_id bigint,
    group_number integer,
    CONSTRAINT listener_pkey PRIMARY KEY (passport_number),
    CONSTRAINT listener_group_number_fkey FOREIGN KEY
(group_number)
REFERENCES public.study_group (group_number) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
    CONSTRAINT listener_program_id_fkey FOREIGN KEY (program_id)
REFERENCES public.program (program_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);

```


Скрипты работы с БД для вставки данных:

```
INSERT INTO teacher (passport_number, full_name) VALUES
```

```
('3617349391', 'Joseph Smith'),
```

```
('3123534231', 'Jane Simpson'),
```

```
('3467665234', 'Andrew Johnson');
```

```
INSERT INTO classroom (classroom_number, platform_address,  
classroom_type) VALUES
```

```
(2041, '123 Main St', 'lecture'),
```

```
(2042, '123 Main St', 'practice'),
```

```
(1013, '789 Maple Ln', 'lecture');
```

```
INSERT INTO discipline (discipline_name, lecture_hours, practice_hours,  
lab_hours, certification_format) VALUES
```

```
('Databases', 16, 16, 8, 'full-time'),
```

```
('Algebra', 16, 16, 8, 'full-time'),
```

```
('Physical Culture', 0, 16, 0, 'full-time');
```

Выводы

В процессе выполнения лабораторной работы удалось детально ознакомиться с работой с PostgreSQL, используя pgAdmin, получить навыки создания таблиц, установки ограничений на таблицы, а также создания и восстановления резервных копий баз данных.