

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

**«Создание таблиц базы данных PostgreSQL. Заполнение таблиц  
рабочими данными»**

**по дисциплине «Проектирование и реализация баз данных»**

**Обучающийся:** Лазарев Марк Олегович

**Факультет** прикладной информатики

**Группа** K3241

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Мобильные и сетевые технологии 2025

**Преподаватель** Говорова Марина Михайловна

Санкт-Петербург  
2025

## **Цель работы**

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

## **Программное обеспечение**

СУБД PostgreSQL, 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.

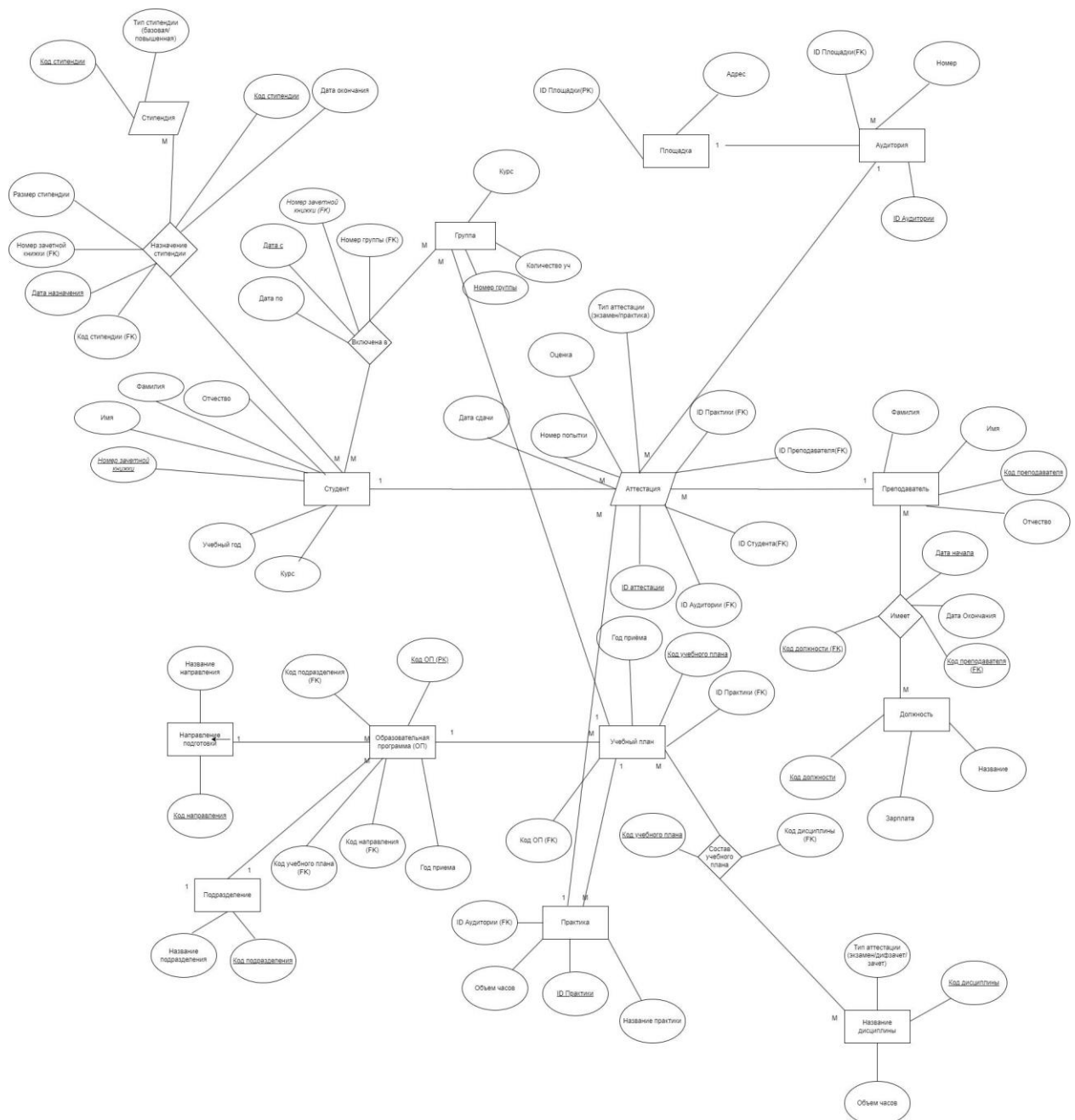
Указание:

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.

## Модель для создания базы данных

Модель представляет организацию сессии внутри университета, включает в себя информацию о преподавателях и их должностях, студентах, стипендии, дисциплинах и их типах, расписании, аттестационной комиссии и прочем.



## Ход работы

Я установил pgAdmin 4 согласно указаниям лабораторной работы, с помощью GUI создал базу данных «bdlab3».

```
-- Таблица: field_of_study
CREATE TABLE field_of_study (
    field_of_study_id INTEGER PRIMARY KEY, -- Уникальный идентификатор
    направления
    field_name TEXT NOT NULL UNIQUE, -- Название направления
    fieldCode INTEGER UNIQUE NOT NULL CHECK (fieldCode > 0) -- Код
    направления должен быть положительным и уникальным
);

-- Таблица: auditorium
CREATE TABLE auditorium (
    auditorium_id INTEGER PRIMARY KEY, -- Уникальный идентификатор аудитории
    auditorium_name TEXT NOT NULL, -- Название аудитории
    field_of_study_id INTEGER NOT NULL, -- Ссылка на направление
    FOREIGN KEY (field_of_study_id) REFERENCES
    field_of_study(field_of_study_id) ON DELETE CASCADE -- При удалении
    направления удаляются связанные аудитории
);

-- Таблица: student
CREATE TABLE student (
    record_book_number INTEGER PRIMARY KEY, -- Уникальный номер зачетной
    книжки
    first_name TEXT NOT NULL, -- Имя студента
    last_name TEXT NOT NULL, -- Фамилия студента
    middle_name TEXT, -- Отчество (необязательно)
    birth_date DATE CHECK (birth_date <= CURRENT_DATE), -- Дата рождения не
    позже текущей даты
    academic_year TEXT NOT NULL, -- Учебный год
    course INTEGER CHECK (course BETWEEN 1 AND 6) -- Курс от 1 до 6
);

-- Таблица: curriculum
CREATE TABLE curriculum (
    curriculum_id INTEGER PRIMARY KEY, -- Уникальный ID учебного плана
    specialty_code INTEGER NOT NULL, -- Код специальности
    direction_code INTEGER NOT NULL, -- Код направления
    total_hours INTEGER CHECK (total_hours >= 0) -- Общее количество часов
    должно быть неотрицательным
);

-- Таблица: curriculum_discipline
CREATE TABLE curriculum_discipline (
    curriculum_discipline_id INTEGER PRIMARY KEY, -- Уникальный ID
    дисциплины
    curriculum_id INTEGER NOT NULL, -- Ссылка на учебный план
    course_id INTEGER, -- Идентификатор курса
    code_of_discipline INTEGER NOT NULL, -- Код дисциплины
    FOREIGN KEY (curriculum_id) REFERENCES curriculum(curriculum_id) ON
    DELETE CASCADE, -- Удаление при удалении учебного плана
    FOREIGN KEY (code_of_discipline) REFERENCES
    name_of_discipline(code_of_discipline) ON DELETE CASCADE -- Удаление при
    удалении дисциплины
);

-- Таблица: name of discipline
```

```

CREATE TABLE name_of_discipline (
    code_of_discipline INTEGER PRIMARY KEY, -- Уникальный код дисциплины
    type_of_certification TEXT NOT NULL, -- Тип аттестации (зачет, экзамен и
т.п.)
    hours INTEGER CHECK (hours > 0) -- Количество часов должно быть
положительным
);

-- Таблица: specialty
CREATE TABLE specialty (
    specialty_id INTEGER PRIMARY KEY, -- Уникальный ID специальности
    specialty_code INTEGER UNIQUE NOT NULL, -- Уникальный код специальности
    specialty_name TEXT NOT NULL -- Название специальности
);

-- Таблица: teacher
CREATE TABLE teacher (
    teacher_id INTEGER PRIMARY KEY, -- Уникальный ID преподавателя
    first_name TEXT NOT NULL, -- Имя
    last_name TEXT NOT NULL, -- Фамилия
    middle_name TEXT, -- Отчество
    position TEXT -- Должность
);

-- Таблица: teacher_position
CREATE TABLE teacher_position (
    teacher_position_id INTEGER PRIMARY KEY, -- Уникальный ID назначения
    start_date DATE NOT NULL, -- Дата начала
    end_date DATE, -- Дата окончания
    teacher_id INTEGER NOT NULL, -- Преподаватель
    specialty_id INTEGER NOT NULL, -- Специальность
    CHECK (end_date IS NULL OR end_date > start_date), -- Конец позже начала
    FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id) ON DELETE
CASCADE,
    FOREIGN KEY (specialty_id) REFERENCES specialty(specialty_id) ON DELETE
CASCADE
);

-- Таблица: practice
CREATE TABLE practice (
    practice_id INTEGER PRIMARY KEY, -- Уникальный ID практики
    practice_name TEXT NOT NULL, -- Название практики
    curriculum_id INTEGER NOT NULL, -- Ссылка на учебный план
    FOREIGN KEY (curriculum_id) REFERENCES curriculum(curriculum_id) ON
DELETE CASCADE
);

-- Таблица: assessment
CREATE TABLE assessment (
    assessment_id INTEGER PRIMARY KEY, -- Уникальный ID оценки
    assessment_date DATE NOT NULL, -- Дата оценки
    mark INTEGER CHECK (mark BETWEEN 0 AND 100), -- Оценка от 0 до 100
    student_id INTEGER NOT NULL, -- Студент
    teacher_id INTEGER NOT NULL, -- Преподаватель
    practice_id INTEGER NOT NULL, -- Практика
    auditorium_id INTEGER NOT NULL, -- Аудитория
    FOREIGN KEY (student_id) REFERENCES student(record_book_number) ON
DELETE CASCADE,
    FOREIGN KEY (teacher_id) REFERENCES teacher(teacher_id) ON DELETE
CASCADE,
    FOREIGN KEY (practice_id) REFERENCES practice(practice_id) ON DELETE
CASCADE,

```

```

        FOREIGN KEY (auditorium_id) REFERENCES auditorium(auditorium_id) ON
DELETE CASCADE
);

-- Таблица: group
CREATE TABLE "group" (
    group_id INTEGER PRIMARY KEY, -- Уникальный ID группы
    group_name TEXT NOT NULL, -- Название группы
    students_count INTEGER CHECK (students_count >= 0), -- Количество
студентов
    specialty_code INTEGER NOT NULL, -- Специальность
    course INTEGER CHECK (course BETWEEN 1 AND 6) -- Курс от 1 до 6
);

-- Таблица: group_curriculum
CREATE TABLE group_curriculum (
    group_curriculum_id INTEGER PRIMARY KEY, -- Уникальный ID
curriculum_id INTEGER NOT NULL, -- Учебный план
    group_id INTEGER NOT NULL, -- Группа
    FOREIGN KEY (curriculum_id) REFERENCES curriculum(curriculum_id) ON
DELETE CASCADE,
    FOREIGN KEY (group_id) REFERENCES "group"(group_id) ON DELETE CASCADE
);

-- Таблица: schedule
CREATE TABLE schedule (
    schedule_id INTEGER PRIMARY KEY, -- Уникальный ID расписания
    date_from DATE NOT NULL, -- Начало действия
    date_to DATE NOT NULL, -- Конец действия
    exam_date DATE, -- Дата экзамена
    group_id INTEGER NOT NULL, -- Группа
    group_number INTEGER, -- Номер группы
    CHECK (date_to >= date_from), -- Дата окончания >= дате начала
    FOREIGN KEY (group_id) REFERENCES "group"(group_id) ON DELETE CASCADE
);

-- Таблица: scholarship
CREATE TABLE scholarship (
    scholarship_code INTEGER PRIMARY KEY, -- Уникальный код стипендии
    type_of_scholarship TEXT NOT NULL, -- Тип стипендии
    scholarship_amount INTEGER CHECK (scholarship_amount > 0), -- Сумма > 0
    end_date DATE -- Дата окончания
);

-- Таблица: scholarship_assignment
CREATE TABLE scholarship_assignment (
    scholarship_assignment_id INTEGER PRIMARY KEY, -- Уникальный ID
назначения
    assignment_date DATE NOT NULL, -- Дата назначения
    scholarship_code INTEGER NOT NULL, -- Код стипендии
    amount_of_scholarship INTEGER CHECK (amount_of_scholarship > 0), --
Сумма назначения
    record_book_number INTEGER NOT NULL, -- Студент
    purpose_of_the_scholarship TEXT, -- Назначение
    FOREIGN KEY (scholarship_code) REFERENCES scholarship(scholarship_code)
ON DELETE CASCADE,
    FOREIGN KEY (record_book_number) REFERENCES student(record_book_number)
ON DELETE CASCADE
);

-- Таблица: education_institution
CREATE TABLE education_institution (
    education_institution_id SERIAL PRIMARY KEY, -- Уникальный ID

```

```

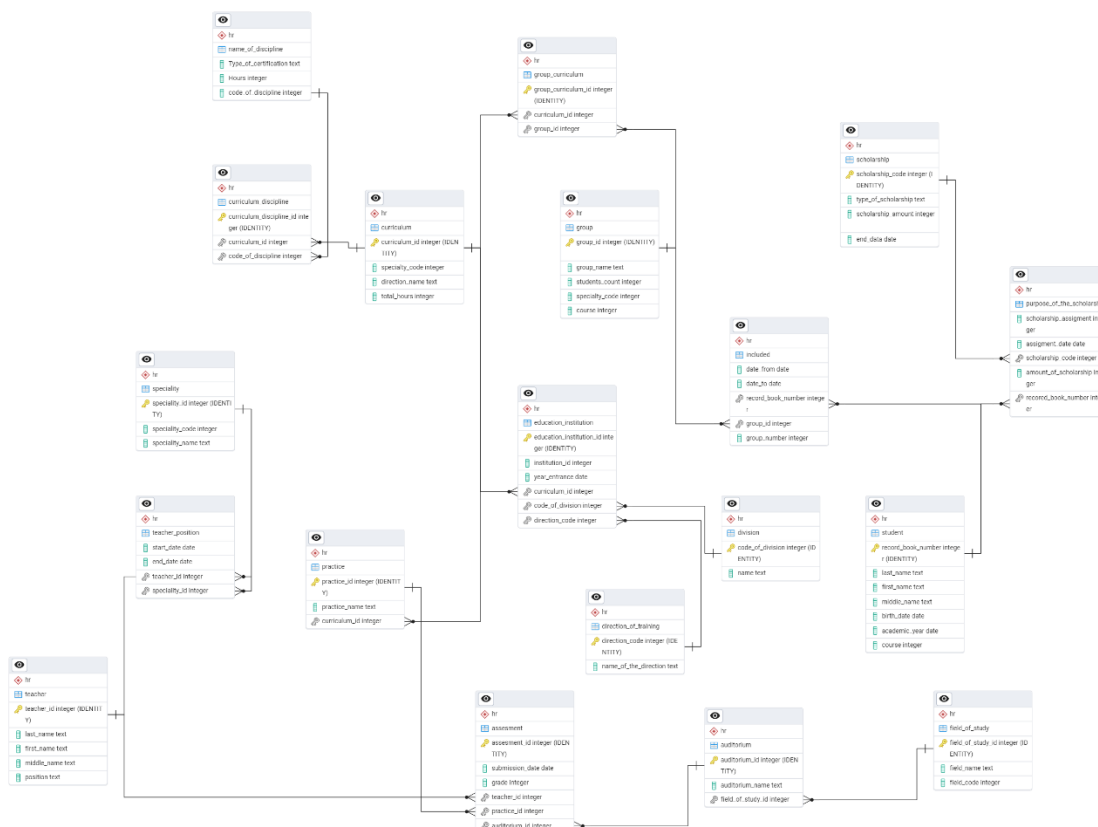
institution_id INTEGER NOT NULL, -- Код учреждения
institution_name TEXT NOT NULL, -- Название учреждения
code_of_division INTEGER NOT NULL, -- Код подразделения
direction_code INTEGER NOT NULL, -- Код направления
FOREIGN KEY (code_of_division) REFERENCES division(code_of_division) ON
DELETE CASCADE,
FOREIGN KEY (direction_code) REFERENCES
director_of_training(direction_code) ON DELETE CASCADE
);

-- Таблица: division
CREATE TABLE division (
code_of_division INTEGER PRIMARY KEY, -- Код подразделения
name TEXT NOT NULL -- Название
);

-- Таблица: director_of_training
CREATE TABLE director_of_training (
direction_code INTEGER PRIMARY KEY, -- Код направления
name_of_the_direction TEXT NOT NULL -- Название направления
);

```

Далее я работал с «ERD for Database».



Следующим шагом было необходимо вставить рабочие данные в таблицы базы данных. Вставка происходила с помощью INSERT.

```
INSERT INTO hr.assessment (assessment_id, submission_date, grade, teacher_id,
practice_id, auditorium_id)
VALUES
    (1, '2023-12-20', 1, 1, 1, 1),
    (2, '2023-12-21', 1, 2, 2, 2);
SELECT * FROM hr.assessment
ORDER BY assessment_id ASC

INSERT INTO hr.teacher_position (start_date, end_date, teacher_id,
speciality_id)
VALUES
    ('2020-09-01', NULL, 1, 1),
    ('2019-01-01', '2023-01-01', 2, 2);
SELECT * FROM hr.teacher_position

INSERT INTO hr.curriculum_discipline (curriculum_discipline_id,
curriculum_id, code_of_discipline)
VALUES
    (1, 1, 101),
    (2, 1, 102);
SELECT * FROM hr.curriculum_discipline
ORDER BY curriculum_discipline_id ASC

INSERT INTO hr.curriculum (curriculum_id, specialty_code, direction_name,
total_hours)
VALUES
    (3, 5252, 5522, 2500),
    (4, 6969, 2255, 2800);
SELECT * FROM hr.curriculum

INSERT INTO hr.direction_of_training (direction_code, name_of_the_direction)
VALUES
    (1, 'Информационные технологии'),
    (2, 'Прикладная математика'),
    (3, 'Экономика'),
    (4, 'Юриспруденция');
SELECT * FROM hr.direction_of_training
ORDER BY direction_code ASC;

INSERT INTO hr.division (code_of_division, name)
VALUES
    (101, 'Физико-математический факультет'),
    (102, 'Инженерный факультет');
SELECT * FROM hr.division ORDER BY code_of_division;

INSERT INTO hr.education_institution (education_institution_id,
institution_id, year_entrance, curriculum_id, code_of_division,
direction_code)
VALUES
    (1, 1001, '21-02-2021', 101, 1, 1),
    (2, 1002, '01-03-2022', 102, 2, 2);
SELECT * FROM hr.education_institution

INSERT INTO hr.field_of_study (field_of_study_id, field_name, field_code)
VALUES
```



```

        (1, 'Программная инженерия', 4101),
        (2, 'Прикладная математика', 4102);
SELECT * FROM hr.field_of_study

INSERT INTO hr."group" (group_id, group_name, students_count,
specialty_code, course)
VALUES
    (1, 'ПИН', 25, 120100, 3),
    (2, 'ИБТ', 20, 140200, 2);
SELECT * FROM hr."group"
ORDER BY group_id ASC

INSERT INTO hr.teacher_position (start_date, end_date, teacher_id,
speciality_id)
VALUES
    ('2020-09-01', NULL, 1, 1),
    ('2019-01-01', '2023-01-01', 2, 2);
SELECT * FROM hr.teacher_position

INSERT INTO hr.included (date_from, date_to, record_book_number, group_id,
group_number)
VALUES
    ('20-10-2020', '03-02-2021', 1, 1, 1),
    ('11-02-2020', '10-02-2022', 2, 2, 2);
SELECT * FROM hr.included

INSERT INTO hr.practice (practice_id, practice_name, curriculum_id)
VALUES
    (1, 'Учебная практика', 1),
    (2, 'Производственная практика', 2);
SELECT * FROM hr.practice

INSERT INTO hr.purpose_of_the_scholarship (scholarship_assignment,
assignment_date, scholarship_code, amount_of_scholarship,
recorded_book_number)
VALUES
    (1, '2023-09-01', 1, 3000, 10001),
    (2, '2023-09-01', 2, 2500, 10002);
SELECT * FROM hr.purpose_of_the_scholarship

INSERT INTO hr.scholarship (scholarship_code, type_of_scholarship,
scholarship_amount, end_data)
VALUES
    (1, 'Академическая', 3000, '2024-06-01'),
    (2, 'Социальная', 2500, NULL);
SELECT * FROM hr.scholarship

INSERT INTO hr.speciality (speciality_id, speciality_code, speciality_name)
VALUES
    (1, 120100, 'Информационные технологии'),
    (2, 140200, 'Математическое моделирование');
SELECT * FROM hr.speciality

INSERT INTO hr.teacher (teacher_id, first_name, last_name, middle_name,
position)
VALUES
    (10, 'Смирнов', 'Георгий', 'Валерьевич', 'доцент'),
    (20, 'Лазарев', 'Марк', 'Олегович', 'профессор');
SELECT * FROM hr.teacher

-- Таблица: teacher_position

```

```

INSERT INTO hr.teacher_position (start_date, end_date, teacher_id,
speciality_id)
VALUES
    ('2020-09-01', NULL, 1, 1),
    ('2019-01-01', '2023-01-01', 2, 2);
SELECT * FROM hr.teacher_position

```

Далее с помощью запросов `SELECT * FROM` получилось вывести таблицы с новыми данными (Все вышеуказанные таблицы сопоставлены со скринами ниже в соответствующем порядке).

	assesment_id [PK] integer	submission_date date	grade integer	teacher_id integer	practice_id integer	auditorium_id integer
1	1	2023-12-20	1	1	1	1
2	2	2023-12-21	1	2	2	2

	auditorium_id [PK] integer	auditorium_name text	field_of_study_id integer
1	1	Аудитория 101	1
2	2	Аудитория 202	2

	<b>curriculum_id</b> [PK] integer	<b>specialty_code</b> integer	<b>direction_name</b> text	<b>total_hours</b> integer
1	1	120100	1	2500
2	2	140200	2	2800
3	3	5252	5522	2500
4	4	6969	2255	2800

	<b>curriculum_discipline_id</b> [PK] integer	<b>curriculum_id</b> integer	<b>code_of_discipline</b> integer
1	1	1	101
2	2	1	102

	<b>direction_code</b> [PK] integer	<b>name_of_the_direction</b> text
1	1	Информационные технологии
2	2	Прикладная математика
3	3	Экономика
4	4	Юриспруденция

	<b>code_of_division</b> [PK] integer	<b>name</b> text
1	101	Физико-математический факультет
2	102	Инженерный факультет

	<b>education_institution_id</b> [PK] integer	<b>institution_id</b> integer	<b>year_entrance</b> date	<b>curriculum_id</b> integer	<b>code_of_division</b> integer	<b>direction_code</b> integer
1	1	1001	2021-02-21	101	1	1
2	2	1002	2022-03-01	102	2	2

	<b>field_of_study_id</b> [PK] integer	<b>field_name</b> text	<b>field_code</b> integer
1	1	Программная инженер...	4101
2	2	Прикладная математика	4102

	<b>group_id</b> [PK] integer	<b>group_name</b> text	<b>students_count</b> integer	<b>specialty_code</b> integer	<b>course</b> integer
1	1	ПИН	25	120100	3
2	2	ИВТ	20	140200	2

	group_curriculum_id [PK] integer	curriculum_id integer	group_id integer
1	1	1	1
2	2	2	2

	date_from date	date_to date	record_book_number integer	group_id integer	group_number integer
1	2020-10-20	2021-02-03	1	1	1
2	2020-02-11	2022-02-10	2	2	2

	practice_id [PK] integer	practice_name text	curriculum_id integer
1	1	Учебная практика	1
2	2	Производственная практика	2

	scholarship_assignment integer	assignment_date date	scholarship_code integer	amount_of_scholarship integer	recorded_book_number integer
1	1	2023-09-01	1	3000	10001
2	2	2023-09-01	2	2500	10002

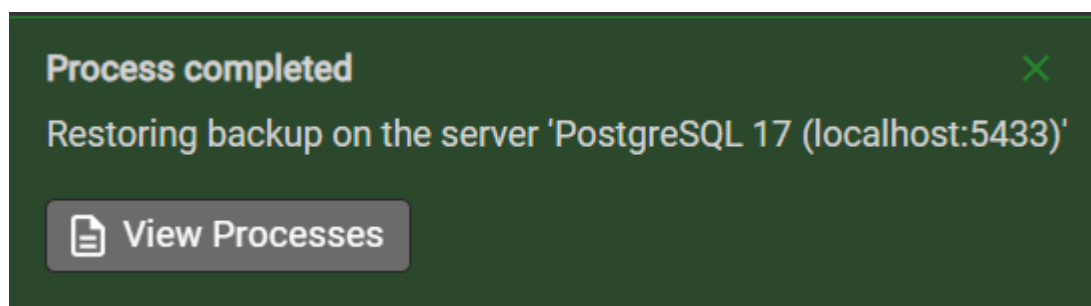
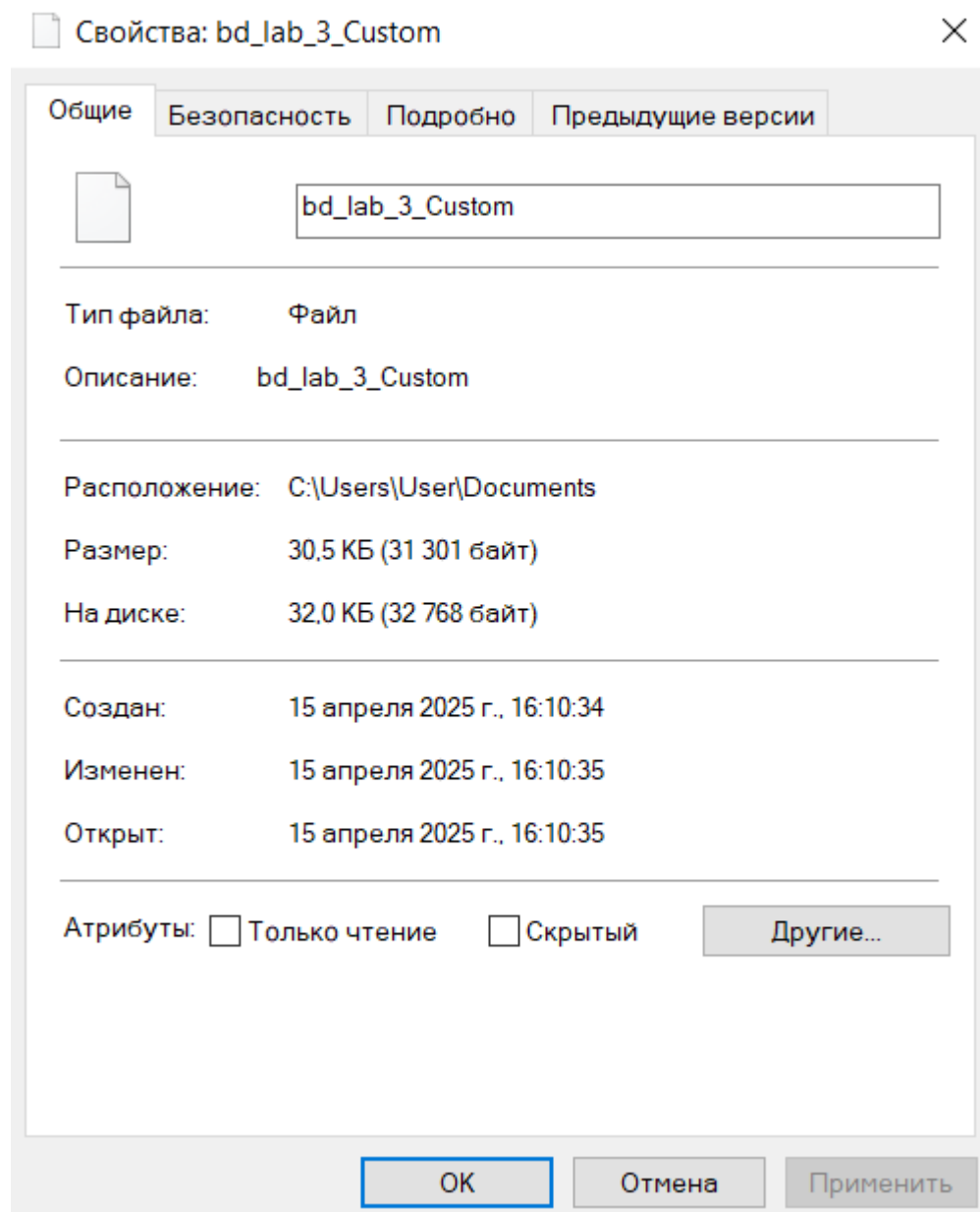
	scholarship_code [PK] integer	type_of_scholarship text	scholarship_amount integer	end_data date
1	1	Академическая	3000	2024-06-01
2	2	Социальная	2500	[null]

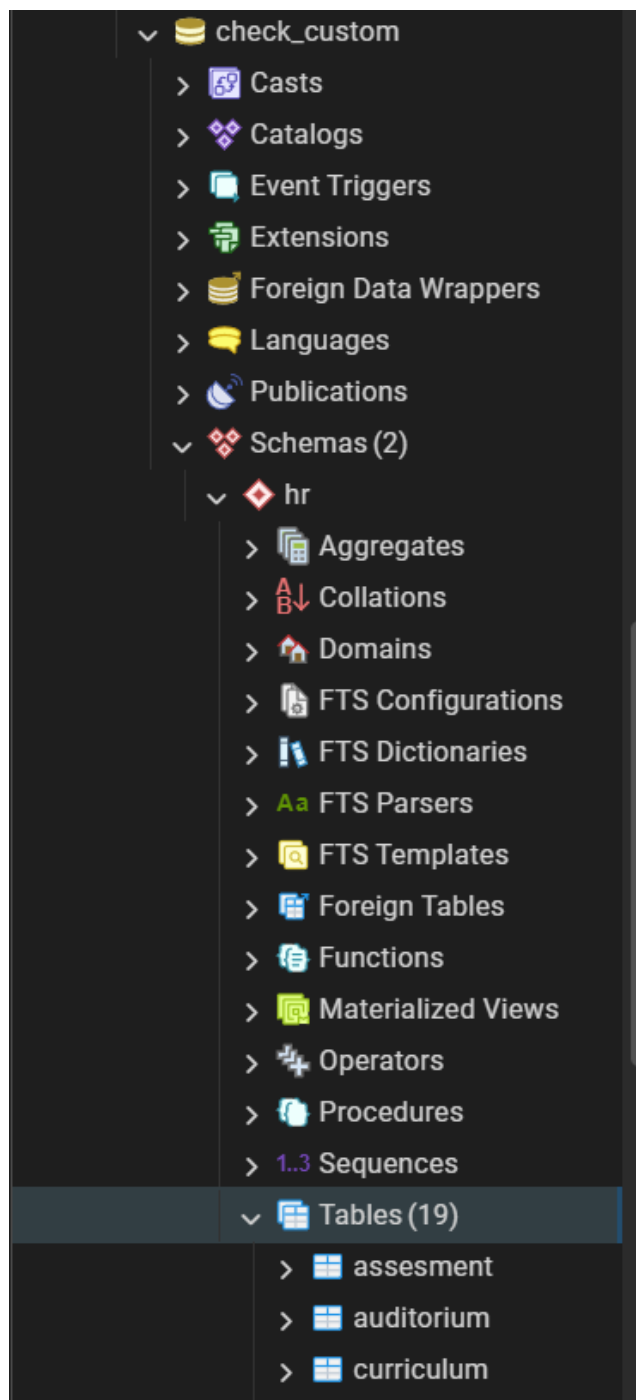
	speciality_id [PK] integer	speciality_code integer	speciality_name text
1	1	120100	Информационные технологии
2	2	140200	Математическое моделирование

	record_book_number [PK] integer	last_name text	first_name text	middle_name text	birth_date date	academic_year date	course integer
1	1	Сергеевич	Михаил	Смирнов	2000-03-15	2023-09-01	2
2	2	Петровна	Анна	Попова	2002-11-30	2020-09-01	4
3	3	Александровна	Елена	Лебедева	2001-06-18	2024-09-01	1

	teacher_id [PK] integer	last_name text	first_name text	middle_name text	position text
1	10	Георгий	Смирнов	Валерьевич	доцент
2	20	Марк	Лазарев	Олегович	профессор

	start_date date	end_date date	teacher_id integer	speciality_id integer
1	2020-09-01	[null]	1	1
2	2019-01-01	2023-01-01	2	2





Затем я создал пустую базу данных и восстановил исходную базу данных с помощью .backup файла

## Вывод

В ходе выполнения данной лабораторной работы я научился работать с pgAdmin 4, проработал навыки с SQL-запросами, узнал о механизме заполнения таблиц рабочими данными и научился создавать резервные копии баз данных и восстанавливать их.