

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

**Факультет безопасности информационных технологий**

**Направление подготовки: 10.03.01 Информационная безопасность**

**Образовательная программа: "Информационная безопасность / Information security"**

**Дисциплина:**

**«Информационная безопасность баз данных»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

**«Создание базовых структур БД»**

**Выполнил студент(ы):**

группа/поток 1.3

Бардышев Артём Антонович/\_\_\_\_\_

*Подпись*

**Проверил:**

Карманова Наталья Андреевна/\_\_\_\_\_

*Подпись*

\_\_\_\_\_  
*Отметка о выполнении (один из вариантов:  
отлично, хорошо, удовлетворительно, зачтено)*

\_\_\_\_\_  
*Дата*

Санкт-Петербург

2025г.

## 1)Цель работы:

Получение навыков создания базовых структурных элементов базы данных.

## 2)Задание

1. Подключиться к базе данных с использованием клиента `psql`.
2. Создать базу данных в соответствии с шаблоном `номергруппы_номерстудента`.
3. Выбрать базу данных в соответствии с шаблоном `номергруппы_номерстудента`. Например, `n3247_1`.
4. Создать новую схему в своей БД с именем `номергруппы_номерстудента_schema_lab1`.
5. Создать таблицу `номергруппы_номерстудента_tbl1` в схеме по умолчанию и таблицу `номергруппы_номерстудента_tbl2` в схеме, созданной в пункте 4.
6. Создать таблицу `номергруппы_номерстудента_table_i5` в своей схеме (далее все объекты нужно создавать в своей схеме из пункта 4), в которой будут использоваться атрибуты с типами `integer`, `varchar`, `char`, `timestamp`, `date`, `bytea`.
7. Создать таблицу `номергруппы_номерстудента_table_i6`, в которой будет атрибут с любым типом со значением по умолчанию и атрибут с типом интервал времени.
8. Создать свой составной тип с именем `номергруппы_номерстудента_type` для описания свойств какого-либо объекта со свойствами `real`, `real`, `date`, `bytea`.
9. Создать свой тип перечисления с именем `номергруппы_номерстудента_enum` для возможности указания градации размера большой, средний, маленький.
10. Создать свой домен с именем `номергруппы_номерстудента_domain` с проверкой, что в нем содержится только 3 цифры
11. Создать последовательность с именем `номергруппы_номерстудента_seq` с началом 1000 и шагом -1. (1000,999, 998, ...)
12. Создать для таблицы с пункта 6 индекс с именем `номергруппы_номерстудента_idx1` по атрибуту с типом `integer`.
13. Создать составной индекс для таблицы из пункта 6 с именем `номергруппы_номерстудента_idx2` по атрибутам `integer`, `date`.
14. Создать индекс по выражению для таблицы из пункта 6 с именем `номергруппы_номерстудента_idx3` по атрибуту `integer`, выражение взятие по модулю 10.
15. Создать частичный индекс для таблицы из пункта 6 с именем `номергруппы_номерстудента_idx4` по атрибуту `integer`, исключая значения меньше 100 и больше 1000.
16. Создать таблицу как в пункте 6, но с ограничение `NOT NULL` на поле `char` с именем `номергруппы_номерстудента_notnull`
17. Создать таблицу как в пункте 6, но с ограничение `UNIQUE` на комбинацию полей `char`, `integer` с именем `номергруппы_номерстудента_unique`

18. Создать таблицу как в пункте 6, но с ограничением первичного ключа поля integer с именем номергруппы\_номерстудента\_pk
19. Создать таблицу как в пункте 6, но с ограничением проверкой поля varchar на наличие символа 'а' номергруппы\_номерстудента\_check
20. Создать представление, в котором из таблицы из пункта 6 будут представлены только атрибуты с типом varchar и date, имя представления номергруппы\_номерстудента\_view

### 3)Ход выполнения работы:

1. Заходим на официальный сайт PostgreSQL и устанавливаем версию для Windows 64-bit
2. Настраиваем БД по удобным для себя параметрам и пункт 1 из задания уже выполнен, начинаем выполнение лабораторной работы
3. Для выполнения пункта 2 нам потребуется ввести

```
CREATE DATABASE n3246_408221;
```

```
\c n3246_408221;
```

#### 4. Создание схемы

```
CREATE SCHEMA n3246_408221_schema_lab1;
```

#### 5. Создание таблиц

```
CREATE TABLE n3246_408221_tbl1 (
```

```
    id SERIAL PRIMARY KEY,
```

```
    name VARCHAR(50)
```

```
);
```

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_tbl2 (
```

```
    id SERIAL PRIMARY KEY,
```

```
    description TEXT
```

```
);
```

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_table_i5 (
```

```
    id SERIAL PRIMARY KEY,
```

```
    num_attr INTEGER,
```

```
    text_attr VARCHAR(100),
```

```
    char_attr CHAR(10),
```

```
    timestamp_attr TIMESTAMP,
```

```
    date_attr DATE,
```

```
    bytea_attr BYTEA
```

);

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_table_i6 (  
    id SERIAL PRIMARY KEY,  
    default_attr TEXT DEFAULT 'default_value',  
    interval_attr INTERVAL
```

);

#### 6. Создание перечисляемого типа ENUM

```
DROP TYPE IF EXISTS n3246_408221_enum CASCADE;
```

```
CREATE TYPE n3246_408221_enum AS ENUM ('big', 'medium', 'small');
```

\\ из-за того что postgres конфликтует с кириллицей названия enum'ов записаны на английском языке

#### 7. Создание домена

```
DROP DOMAIN IF EXISTS n3246_408221_domain CASCADE;
```

```
CREATE DOMAIN n3246_408221_domain AS TEXT  
    CHECK (VALUE ~ '^[0-9]{3}$');
```

#### 8. Создание последовательности

```
CREATE SEQUENCE n3246_408221_seq  
START WITH 1000  
INCREMENT BY -1  
MAXVALUE 1000  
MINVALUE 0;
```

#### 9. Создание индексов

```
CREATE INDEX n3246_408221_idx1  
ON n3246_408221_schema_lab1.n3246_408221_table_i5 (num_attr);
```

```
CREATE INDEX n3246_408221_idx2  
ON n3246_408221_schema_lab1.n3246_408221_table_i5 (num_attr, date_attr);
```

```
CREATE INDEX n3246_408221_idx3  
ON n3246_408221_schema_lab1.n3246_408221_table_i5 ((num_attr % 10));
```

```
CREATE INDEX n3246_408221_idx4
```

```
ON n3246_408221_schema_lab1.n3246_408221_table_i5 (num_attr)
WHERE num_attr >= 100 AND num_attr <= 1000;
```

#### 10. Создание таблиц с ограничениями

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_notnull (
    id SERIAL PRIMARY KEY,
    num_attr INTEGER,
    text_attr VARCHAR(100),
    char_attr CHAR(10) NOT NULL,
    timestamp_attr TIMESTAMP,
    date_attr DATE,
    bytea_attr BYTEA
);
```

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_unique (
    id SERIAL PRIMARY KEY,
    num_attr INTEGER,
    text_attr VARCHAR(100),
    char_attr CHAR(10),
    timestamp_attr TIMESTAMP,
    date_attr DATE,
    bytea_attr BYTEA,
    UNIQUE (char_attr, num_attr)
);
```

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_pk (
    id SERIAL PRIMARY KEY,
    num_attr INTEGER NOT NULL
);
```

```
CREATE TABLE n3246_408221_schema_lab1.n3246_408221_check (
    id SERIAL PRIMARY KEY,
    text_attr VARCHAR(100) CHECK (text_attr LIKE '%a%'),
```

```
num_attr INTEGER,  
char_attr CHAR(10),  
timestamp_attr TIMESTAMP,  
date_attr DATE,  
bytea_attr BYTEA  
);
```

#### 11. Создание представления

```
CREATE OR REPLACE VIEW n3246_408221_view AS  
SELECT text_attr, date_attr  
FROM n3246_408221_schema_lab1.n3246_408221_table_i5;
```

#### Заключение

В ходе лабораторной работы был изучен принцип работы с базой данных в PostgreSQL, в частности, создание, наполнение, написание условий.