МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
**«Вятский государственный университет»**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**«Основы DDL-запросов в PostgreSQL»**

Отчёт по лабораторной работе №1 дисциплины

«Базы данных»

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К.А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В. Л./

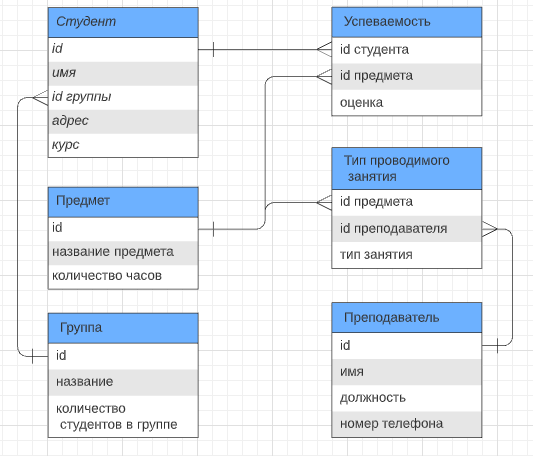
Киров 2022

**Цели лабораторной работы:**

1) Познакомится со схемами, пользователями и ролями в PostgreSQL;   
2) Познакомиться с типами данных в PostgreSQL;   
3) Освоить основные варианты DDl-запросов в PostgreSQL;  
4) Закрепить знания по проектированию структуры реляционной БД;   
5) Создать рабочий материал для следующих лабораторных работ.

**Задание на лабораторную работу:**   
Нужно выполнить следующие шаги:   
1. Придумать структуру базы данных на любую выбранную тему. Структура должна отвечать следующим условиям (это важно для следующих лабораторных работ):   
 • должно быть не меньше пяти таблиц;   
 • хотя бы одна таблица должна содержать колонку с числовыми данными;  
 • структура БД должна быть в третьей нормальной форме.   
2. Создать нового пользователя и пустую БД. Подключиться к созданной БД.   
3. Написать и выполнить SQL-скрипт, создающий таблицы согласно разработанной структуре БД. Созданный в п.2 пользователь должен иметь все права на созданные объекты. В этом же скрипте должны создаваться нужные ограничения и индексы:   
 • обязательно должны быть созданы внешние ключи для поддержания ссылочной целостности;   
 • желательно должны быть проставлены ограничения и уникальные индексы для поддержания консистентности данных (например, если в таблице есть столбец с температурой воздуха за окном, стоит ограничить возможный промежуток значений);   
 • желательно должны быть проставлены индексы для производительности там, где они могут помочь (из расчета, что созданная БД будет хранить достаточно большое количество данных).

**Ход работы:**Создадим базу данных по теме «Студенты». База данных будет содержать основные данные о студенте и его успеваемости по различным предметам. Каждый студент в рамках базы данных имеет ФИО, адрес проживания, номер курса и учебную группу. Также для каждого студента может иметься информация о его успеваемости по тому или иному предмету, включающая название предмета, оценку и длительность предмета в академических часах. База данных также должна содержать информацию о преподавателях, которая включает в себя: ФИО, должность, номер телефона. Помимо этого, база данных должна описывать какой преподаватель проводил занятия и какой тип занятия он вел (лекция, практика, лабораторная работа или семинар). При этом один преподаватель может вести различные предметы и их типы. Различные типы занятий по одному предмету также могут проводить различные преподаватели.   
  
Для реализации данной базы данных необходимо 6 таблиц: «Студент», «Группа», «Успеваемость», «Предмет», «Преподаватель», и промежуточная таблица «Тип проводимого занятия».   
Таблица «Студент» содержит столбцы id, имя, id группы, адрес, курс.   
Таблица «Группа» содержит столбцы id, название, количество студентов в группе.   
Таблица «Успеваемость» содержит столбцы id студента, id предмета, оценку.   
Таблица «Предмет» содержит столбцы id, название предмета, количество часов.   
Таблица «Преподаватель» содержит столбцы id, имя, должность, номер телефона.   
Таблица «Тип проводимого занятия» содержит столбы id предмета, id преподавателя, тип занятия.



--Создание типа перечисления для обозначения типа проводимого занятия

create type public.occupation\_type\_enum as enum (

'lecture', 'laboratory work', 'practical lesson', 'seminar');

--Создание типа перечисления для обозначения должности преподавателя

create type public.position\_type\_enum as enum (

'docent', 'professors', 'senior lecturer', 'instructor');

--Создание таблицы Группа

create table public.group(

id bigserial primary key,

name varchar(30) not null unique,

number\_of\_students int not null check (number\_of\_students >= 1)

);

--Создание таблицы Студент

create table public.student(

id bigserial primary key,

first\_name varchar(30) not null,

second\_name varchar(30) not null,

middle\_name varchar(30),

id\_group bigint not null references public.group(id),

address varchar(100),

cource int check (cource >= 1 and cource <=5)

);

--Создание таблицы Преподаватель

create table public.teacher(

id bigserial primary key,

first\_name varchar(30) not null,

second\_name varchar(30) not null,

middle\_name varchar(30),

position public.position\_type\_enum not null default 'instructor',

phone\_number varchar(30)

);

--Создание таблицы Успеваемость

create table public.progress(

id\_student bigint not null references public.student(id),

id\_subject bigint not null references public.subject(id),

mark int check (mark >= 2 and mark <=5)

);

--Создание таблицы Предмет

create table public.subject(

id bigserial primary key,

name\_subject varchar(30) not null,

hours int

);

--Создание таблицы Тип проводимого занятия

create table public.occupation(

id\_subject bigint not null references public.subject(id),

id\_teacher bigint not null references public.teacher(id),

position public.occupation\_type\_enum not null default 'lecture'

);

grant all on schema public to postgres;

**Вывод:**

В ходе выполнения данной лабораторной работы были освоены основные варианты DDI-запросов в PostgreSQL, закреплены знания по проектированию структуры реляционной БД, а также работали со схемами, пользователями, ролями и типами данных.