ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

*Кафедра прикладной математики и кибернетики*

Лабораторная работа №1

«Основы проектирования БД»

Выполнил: студент группы ЗМП-41

Лёвкин И.А.

Проверил: доцент кафедры ПМиК

Приставка П. А.

Новосибирск, 2025

# Задание

1. Спроектировать БД, содержащую как минимум две таблицы, связанные отношением один-ко-многим.
2. Спроектировать БД, содержащую как минимум две таблицы, связанные отношением многие-ко-многим.

# Ход работы

1. Для выполнения первого задания были созданы следующие запросы:

|  |
| --- |
| CREATE TABLE Authors (  author\_id SERIAL PRIMARY KEY,  full\_name VARCHAR(100) NOT NULL  email VARCHAR(100) UNIQUE  );  CREATE TABLE Books (  book\_id SERIAL PRIMARY KEY  title VARCHAR(200) NOT NULL  'year' INTEGER,  author\_id INTEGER NOT NULL  FOREIGN KEY (author\_id) REFERENCES Authors(author\_id) ON DELETE CASCADE  ) |

В данных запросах были созданы две таблицы Authors и Books со связью один-ко-многим. У каждого автора может быть несколько книг. Но у книги может быть только один автор (данное допущение принято в рамках выполнения задания)

На уровне таблиц это реализовано с помощью внешнего ключа (FOREIGN KEY) author\_id в таблице Books, который связан с первичным ключом author\_id в таблице Authors.

1. Для выполнения второго задания были созданы следующие запросы:

|  |
| --- |
| CREATE TABLE Students (  student\_id SERIAL PRIMARY KEY,  first\_name VARCHAR(50) NOT NULL,  last\_name VARCHAR(50) NOT NULL,  enrollment\_year INTEGER  )  CREATE TABLE Courses (  course\_id SERIAL PRIMARY KEY,  title VARCHAR(100) NOT NULL,  'description' TEXT  )  CREATE TABLE Enrollments(  enrollment\_id SERIAL PRIMARY KEY,  student\_id INTEGER NOT NULL,  course\_id INTEGER NOT NULL,  enrollment\_date DATE DEFAULT CURRENT\_DATE,  grade DECIMAL(3,1),  FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENTS(STUDENT\_ID) ON DELETE CASCADE,  FOREIGN KEY (COURSE\_ID) REFERENCES COURSES(COURSE\_ID) ON DELETE CASCADE,  UNIQUE(STUDENT\_ID, COURSE\_ID)  ) |

В данных запросах были созданы две таблицы Students и Courses с уникальными полями student\_id и course\_id, идентификатор студента и курса соответственно. Связь многие-ко-многим обеспечивает третья таблица Enrollments (Регистрации) которая отражает действие регистрации студента на курс. На одном и том же курсе может быть несколько студентов, так и студент может быть на нескольких курсах