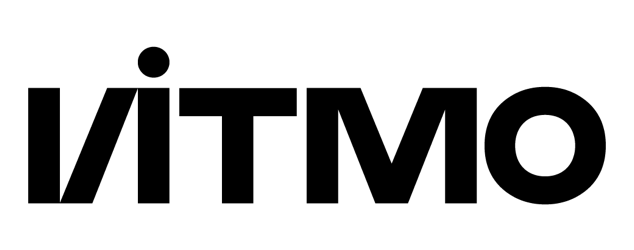
**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательской университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**



**Курсовая работа**

**Этап II**

*по дисциплине*

**Информационные системы**

*Выполнил студент группы P3307:*

**Разинкин Александр Владимирович**

**Батманов Даниил Евгеньевич**

*Преподаватель:*

**Пименов Данила Дмитриевич**

г. Санкт-Петербург

2024г.

**Содержание**

[1 Задание 3](#_Toc179373877)

[2 ER-модель базы данных 4](#_Toc179373878)

[3 Даталогическая модель 5](#_Toc179373879)

[4 Обеспечение целостности данных при помощи средств языка DDL и триггеров 5](#_Toc179373880)

[5 Скрипты для создания, удаления базы данных, заполнения базы тестовыми данными 6](#_Toc179373881)

[6 Pl/pgsql-функции и процедуры, для выполнения критически важных запросов 11](#_Toc179373882)

[7 Индексы на основе анализа использования базы данных в контексте описанных на первом этапе прецедентов, обоснование их полезности 12](#_Toc179373883)

[8 Заключение 12](#_Toc179373884)

[9 Список литературы 12](#_Toc179373885)

# 1 Задание

**Этап 2:**

1. Сформировать ER-модель базы данных (на основе описаний предметной области и прецедентов из предыдущего этапа).  
   ER-модель должна:  
   a. включать в себя не менее 10 сущностей;  
   ​​​​​​​b. содержать хотя бы одно отношение вида «многие-ко-многим».
2. Согласовать ER-модель с преподавателем. На основе ER-модели построить даталогическую модель.
3. Реализовать даталогическую модель в реляционной СУБД PostgreSQL.
4. Обеспечить целостность данных при помощи средств языка DDL и триггеров.
5. Реализовать скрипты для создания, удаления базы данных, заполнения базы тестовыми данными.
6. Предложить pl/pgsql-функции и процедуры, для выполнения критически важных запросов (которые потребуются при последующей реализации прецедентов).
7. Создать индексы на основе анализа использования базы данных в контексте описанных на первом этапе прецедентов. Обосновать полезность созданных индексов для реализации представленных на первом этапе бизнес-процессов.
8. Составить отчет.

# 2 ER-модель базы данных

A diagram of a computer

Description automatically generated

# 3 Даталогическая модель

A diagram of a computer

Description automatically generated

# 4 Обеспечение целостности данных при помощи средств языка DDL и триггеров

DDL средства отражены в даталогической модели.

**Триггеры:**

***Триггер для пересчёта рейтинга автошколы:***

-- Функция для пересчёта рейтинга автошколы

CREATE OR REPLACE FUNCTION update\_driving\_school\_rating()

RETURNS TRIGGER AS $$

BEGIN

-- Пересчитываем средний рейтинг для автошколы

UPDATE driving\_schools

SET rating = (

SELECT AVG(rate)

FROM driving\_school\_reviews

WHERE driving\_schools\_id = NEW.driving\_schools\_id

)

WHERE id = NEW.driving\_schools\_id;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Триггер для обновления рейтинга при добавлении нового отзыва

CREATE TRIGGER update\_rating\_trigger

AFTER INSERT OR UPDATE ON driving\_school\_reviews

FOR EACH ROW

EXECUTE FUNCTION update\_driving\_school\_rating();

# 5 Скрипты для создания, удаления базы данных, заполнения базы тестовыми данными

**Скрипт для создания базы данных:**  
  
-- Таблица roles

CREATE TABLE roles (

id SERIAL PRIMARY KEY,

name VARCHAR(84)

);

-- Таблица users

CREATE TABLE users (

id SERIAL PRIMARY KEY,

email VARCHAR(64) UNIQUE NOT NULL,

password VARCHAR(64) NOT NULL,

first\_name VARCHAR(64),

last\_name VARCHAR(64),

phone\_number VARCHAR(12)

);

-- Таблица roles\_users (связующая таблица между roles и users)

CREATE TABLE roles\_users (

id SERIAL PRIMARY KEY,

roles\_id SERIAL REFERENCES roles(id) ON DELETE CASCADE,

users\_id SERIAL REFERENCES users(id) ON DELETE CASCADE

);

-- Таблица status

CREATE TABLE status (

id SERIAL PRIMARY KEY,

name VARCHAR(64) NOT NULL

);

-- Таблица corporate\_requests

CREATE TABLE corporate\_requests (

id SERIAL PRIMARY KEY,

company\_name VARCHAR(84),

employee\_count INTEGER,

created\_at TIMESTAMP,

description VARCHAR(512)

);

-- Таблица driving\_schools

CREATE TABLE driving\_schools (

id SERIAL PRIMARY KEY,

name VARCHAR(84) NOT NULL,

address VARCHAR(512),

phone\_number VARCHAR(12),

email VARCHAR(64) NOT NULL UNIQUE,

description VARCHAR(512),

rating DECIMAL

);

-- Таблица driving\_school\_reviews

CREATE TABLE driving\_school\_reviews (

id SERIAL PRIMARY KEY,

rate DECIMAL,

review VARCHAR(512),

driving\_schools\_id SERIAL REFERENCES driving\_schools(id) ON DELETE CASCADE

);

-- Таблица instructors

CREATE TABLE instructors (

id SERIAL PRIMARY KEY,

experience INTEGER CHECK (experience >= 0),

description VARCHAR(512),

rating DECIMAL

);

-- Таблица instructor\_reviews

CREATE TABLE instructor\_reviews (

id SERIAL PRIMARY KEY,

rate DECIMAL,

review VARCHAR(512),

instructors\_id SERIAL REFERENCES instructors(id) ON DELETE CASCADE

);

-- Таблица instructor\_services

CREATE TABLE instructor\_services (

id SERIAL PRIMARY KEY,

name VARCHAR(84),

description VARCHAR(512),

price DECIMAL

);

-- Таблица instructors\_driving\_schools (связующая таблица между instructors и driving\_schools)

CREATE TABLE instructors\_driving\_schools (

id SERIAL PRIMARY KEY,

instructors\_id SERIAL REFERENCES instructors(id) ON DELETE CASCADE,

driving\_schools\_id SERIAL REFERENCES driving\_schools(id) ON DELETE CASCADE

);

-- Таблица vehicle\_categories

CREATE TABLE vehicle\_categories (

id SERIAL PRIMARY KEY,

name VARCHAR(84) NOT NULL UNIQUE,

description VARCHAR(512)

);

-- Таблица study\_formats

CREATE TABLE study\_formats (

id SERIAL PRIMARY KEY,

name VARCHAR(84) NOT NULL UNIQUE,

description VARCHAR(512)

);

-- Таблица transmission\_types

CREATE TABLE transmission\_types (

id SERIAL PRIMARY KEY,

name VARCHAR(84) NOT NULL UNIQUE,

description VARCHAR(512)

);

-- Таблица vehicles

CREATE TABLE vehicles (

id SERIAL PRIMARY KEY,

model VARCHAR(84),

year VARCHAR(4)

);

-- Таблица courses

CREATE TABLE courses (

id SERIAL PRIMARY KEY,

name VARCHAR(84) NOT NULL,

price DECIMAL,

duration DECIMAL,

description VARCHAR(512),

vehicle\_categories\_id SERIAL REFERENCES vehicle\_categories(id),

study\_formats\_id SERIAL REFERENCES study\_formats(id),

transmission\_types\_id SERIAL REFERENCES transmission\_types(id)

);

-- Таблица additional\_services

CREATE TABLE additional\_services (

id SERIAL PRIMARY KEY,

name VARCHAR(84),

description VARCHAR(512),

price DECIMAL NOT NULL

);

-- Таблица promotions

CREATE TABLE promotions (

id SERIAL PRIMARY KEY,

title VARCHAR(64),

description VARCHAR(512),

start\_date TIMESTAMP,

end\_date TIMESTAMP

);  
  
**Скрипт для удаления базы данных:**  
  
-- Удаление зависимых таблиц

DROP TABLE IF EXISTS roles\_users;

DROP TABLE IF EXISTS instructor\_reviews;

DROP TABLE IF EXISTS instructor\_services;

DROP TABLE IF EXISTS instructors\_driving\_schools;

DROP TABLE IF EXISTS driving\_school\_reviews;

DROP TABLE IF EXISTS corporate\_requests;

DROP TABLE IF EXISTS vehicles;

DROP TABLE IF EXISTS promotions;

DROP TABLE IF EXISTS additional\_services;

-- Удаление основных таблиц

DROP TABLE IF EXISTS courses;

DROP TABLE IF EXISTS instructors;

DROP TABLE IF EXISTS driving\_schools;

DROP TABLE IF EXISTS status;

DROP TABLE IF EXISTS transmission\_types;

DROP TABLE IF EXISTS vehicle\_categories;

DROP TABLE IF EXISTS study\_formats;

DROP TABLE IF EXISTS users;

DROP TABLE IF EXISTS roles;

**Скрипт для заполнения базы данных тестовыми данными:**  
  
-- Заполнение таблицы roles

INSERT INTO roles (name) VALUES

('Admin'),

('Instructor'),

('Student');

-- Заполнение таблицы users

INSERT INTO users (email, password, first\_name, last\_name, phone\_number) VALUES

('admin@example.com', 'password123', 'John', 'Doe', '1234567890'),

('instructor@example.com', 'password456', 'Jane', 'Smith', '0987654321'),

('student@example.com', 'password789', 'Bob', 'Brown', '1122334455');

-- Заполнение таблицы status

INSERT INTO status (name) VALUES

('Active'),

('Inactive');

-- Заполнение таблицы corporate\_requests

INSERT INTO corporate\_requests (company\_name, employee\_count, created\_at, description) VALUES

('Company A', 100, CURRENT\_TIMESTAMP, 'Request from Company A'),

('Company B', 50, CURRENT\_TIMESTAMP, 'Request from Company B');

-- Заполнение таблицы driving\_schools

INSERT INTO driving\_schools (name, address, phone\_number, email, description, rating) VALUES

('Best Driving School', '123 Main St', '1234567890', 'info@bestdriving.com', 'Top-rated driving school', 4.5),

('Fast Track Driving School', '456 High St', '0987654321', 'info@fasttrack.com', 'Affordable driving lessons', 4.2);

-- Заполнение таблицы vehicle\_categories

INSERT INTO vehicle\_categories (name, description) VALUES

('Car', 'Four-wheeled motor vehicle'),

('Motorcycle', 'Two-wheeled motor vehicle');

-- Заполнение таблицы study\_formats

INSERT INTO study\_formats (name, description) VALUES

('Online', 'Learn driving theory online'),

('In-person', 'Learn driving in person');

-- Заполнение таблицы transmission\_types

INSERT INTO transmission\_types (name, description) VALUES

('Manual', 'Manual transmission vehicle'),

('Automatic', 'Automatic transmission vehicle');

-- Заполнение таблицы courses

INSERT INTO courses (name, price, duration, description) VALUES

('Basic Driving Course', 200.00, 30, 'Basic driving skills and road safety'),

('Advanced Driving Course', 300.00, 40, 'Advanced driving techniques and rules');

-- Заполнение таблицы additional\_services

INSERT INTO additional\_services (name, description, price) VALUES

('Road Test Preparation', 'Prepare for the driving test', 50.00),

('Night Driving Lessons', 'Practice driving at night', 80.00);

-- Заполнение таблицы promotions

INSERT INTO promotions (title, description, start\_date, end\_date) VALUES

('Summer Discount', '20% off all courses in summer', '2024-06-01', '2024-08-31'),

('Winter Offer', 'Free additional lessons in winter', '2024-12-01', '2025-02-28');

-- Заполнение таблицы vehicles

INSERT INTO vehicles (model, year) VALUES

('Toyota Corolla', '2020'),

('Honda Civic', '2019');

-- Заполнение таблицы instructors

INSERT INTO instructors (experience, description, rating) VALUES

(5, 'Experienced instructor with a focus on safety', 4.8),

(3, 'Specializes in beginner drivers', 4.5);

-- Заполнение таблицы instructor\_reviews

INSERT INTO instructor\_reviews (rate, review) VALUES

(5, 'Great instructor, very patient'),

(4, 'Good instructor, but lessons could be longer');

-- Заполнение таблицы instructor\_services

INSERT INTO instructor\_services (name, description, price) VALUES

('Beginner Lessons', 'Lessons for new drivers', 150.00),

('Advanced Lessons', 'Lessons for experienced drivers', 200.00);

-- Заполнение таблицы roles\_users

INSERT INTO roles\_users (roles\_id, users\_id) VALUES

(1, 1), -- Admin

(2, 2), -- Instructor

(3, 3); -- Student

-- Заполнение таблицы driving\_school\_reviews

INSERT INTO driving\_school\_reviews (rate, review) VALUES

(5, 'Excellent school, very professional'),

(4, 'Good experience, but a bit expensive');

-- Заполнение таблицы instructors\_driving\_schools

INSERT INTO instructors\_driving\_schools (instructors\_id, driving\_schools\_id) VALUES

(1, 1),

(2, 2);

# 6 Pl/pgsql-функции и процедуры, для выполнения критически важных запросов

-- Функция для получения автошкол с рейтингом выше указанного

CREATE OR REPLACE FUNCTION get\_driving\_schools\_by\_rating(min\_rating NUMERIC)

RETURNS TABLE (

school\_id INT,

name TEXT,

rating NUMERIC

) AS $$

BEGIN

RETURN QUERY

SELECT id, name, rating

FROM driving\_schools

WHERE rating >= min\_rating

ORDER BY rating DESC;

END;

$$ LANGUAGE plpgsql;

-- Функция для получения списка активных промоакций

CREATE OR REPLACE FUNCTION get\_active\_promotions()

RETURNS TABLE (

promotion\_id INT,

title TEXT,

description TEXT,

start\_date DATE,

end\_date DATE

) AS $$

BEGIN

RETURN QUERY

SELECT id, title, description, start\_date, end\_date

FROM promotions

WHERE start\_date <= CURRENT\_DATE

AND end\_date >= CURRENT\_DATE;

END;

$$ LANGUAGE plpgsql;

# 7 Индексы на основе анализа использования базы данных в контексте описанных на первом этапе прецедентов, обоснование их полезности

***В системе предполагается отбор по цене, ускорить этот процесс поможет b-tree индекс.***

-- Создание B-tree индекса на столбец price в таблице courses

CREATE INDEX idx\_courses\_price\_btree ON courses USING btree (price);

-- Создание B-tree индекса на столбец price в таблице additional\_services

CREATE INDEX idx\_additional\_services\_price\_btree ON additional\_services USING btree (price);

***В системе предполагается поиск автошколы по названию, ускорить процесс поиска позволит b-tree индекс.***

-- Создание B-tree индекса на поле name в таблице driving\_schools

CREATE INDEX idx\_driving\_schools\_name\_btree ON driving\_schools USING btree (name);

# 8 Заключение

В ходе выполнения второго этапа курсовой работы нам удалось составить инфологическую, даталогическую модели БД, создать триггеры, индексы и функции для неё, написать скрипты для создания, удаления и наполнения тестовыми данными БД.

# 9 Список литературы

* Предметная область - <https://avtoshkola177.ru/article/kak-vybrat-avtoshkolu-dlya-obucheniya-na-prava>
* Предметная область - <https://proffskills.ru/news/kak-vyibrat-avtoshkolu-10-klyuchevyih-kriteriev>
* Предметная область - https://autoexamen.ru/articles/nachinayushchim-avtomobilistam/trebovaniya-k-avtoshkolam-v-zakonakh-rf/
* Легендарный конспект - https://se.ifmo.ru/courses/storage