

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа 1

Выполнил:

Сахно Ярослав

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

Спроектировать БД, придерживаясь нотации ERD. Платформа домашних заданий для университетов.

Возможности администратора:

- Просмотр всех пользователей системы с возможностью фильтрации по типу.
- Изменение информации пользователей и их привязку к курсам.
- Создание новых пользователей.
- Просмотр списка регистраций и принятие или отклонение запросов.
- Создание курсов и управление доступом к ним.
- Добавление аватарки пользователям

Возможности преподавателя:

- Запрос доступа на управление курсами.
- Изменение информации о курсах.
- Создание, изменение и удаление домашних заданий в курсах.
- Просмотр статистики решений студентов.

Возможности студента:

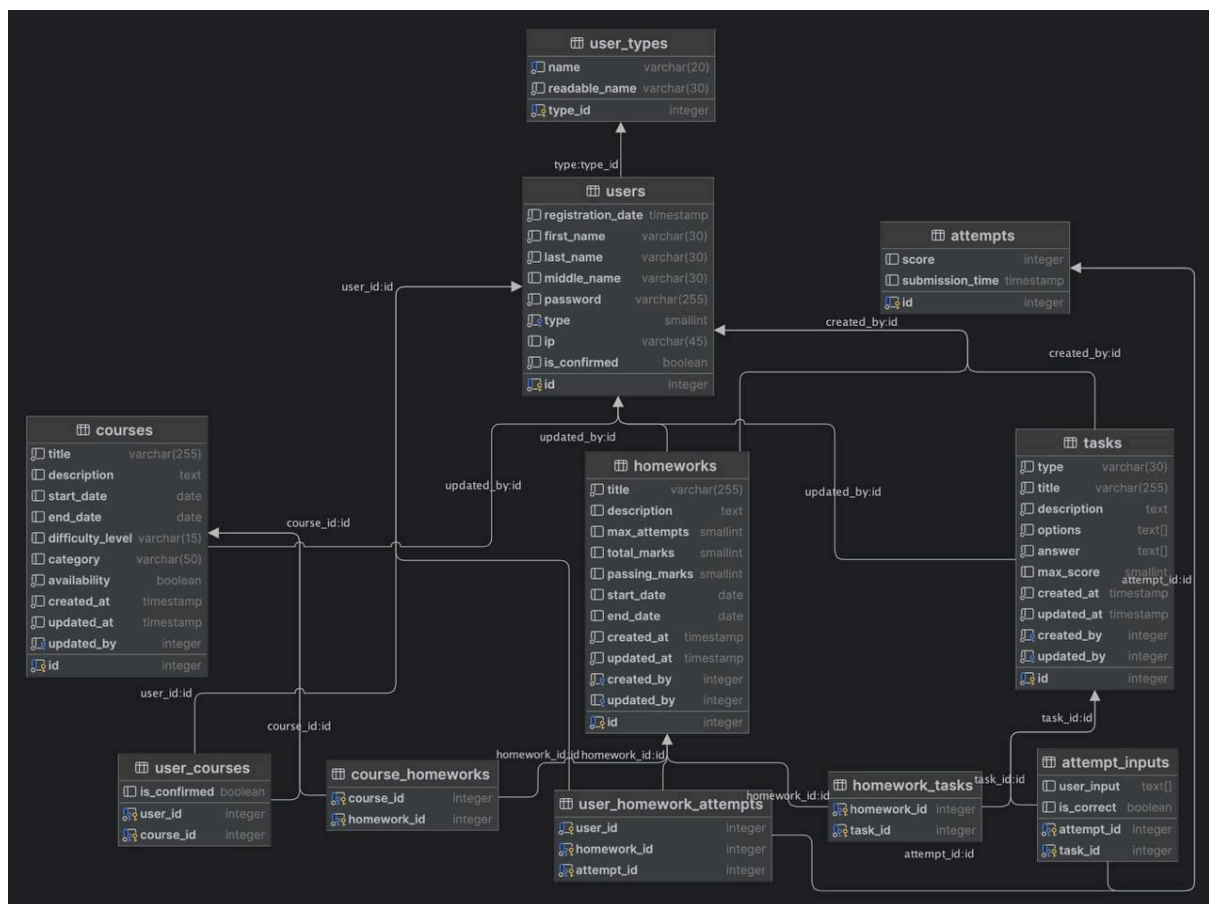
- Запрос доступа к курсам.
- Решение домашних заданий в рамках курсов.

## Ход работы

№	Таблица	Описание полей
1	user_types	type_id (PK, serial), name (varchar, уникально), readable_name (varchar) — справочник ролей пользователей (student, teacher, admin...).
2	users	id (PK), registration_date, first_name, last_name, middle_name, password (хеш), type (FK → user_types.type_id), ip, is_confirmed.

3	courses	id (PK), title, description, start_date, end_date, difficulty_level, category, availability (публикация), created_at, updated_at, updated_by (FK → users.id).
4	user_courses	user_id (FK → users.id), course_id (FK → courses.id), is_confirmed; PK — составной (user_id, course_id) (участие пользователей в курсах).
5	homeworks	id (PK), title, description, max_attempts, total_marks, passing_marks, start_date, end_date, created_at, updated_at, created_by & updated_by (FK → users.id).
6	course_homeworks	course_id (FK → courses.id), homework_id (FK → homeworks.id); PK — (course_id, homework_id) — связь «домашнее задание ↔ курс».
7	tasks	id (PK), type (quiz, multichoice, match...), title, description, options (text[]), answer (text[]), max_score, created_at, updated_at, created_by, updated_by (оба FK → users.id).
8	homework_tasks	homework_id (FK → homeworks.id), task_id (FK → tasks.id); PK — (homework_id, task_id) — набор задач в домашнем задании.
9	attempts	id (PK), score, submission_time — единичная попытка решения.
10	user_homework_attempts	user_id (FK → users.id), homework_id (FK → homeworks.id), attempt_id (FK → attempts.id);

		РК — тройной ключ — журнал всех попыток пользователей.
11	attempt_inputs	attempt_id (FK → attempts.id), task_id (FK → tasks.id), user_input (text[]), is_correct; РК — (attempt_id, task_id) — ответы пользователя по каждому заданию.



## Связи между сущностями

- Один тип пользователя связан со многими пользователями. Один пользователь-преподаватель может создать любое число курсов, другие пользователи записываются через user\_courses (m:n). Домашнее задание (homeworks) создаётся преподавателем и прикрепляется к одному или нескольким курсам (course\_homeworks, m:n).
- Каждое задание (tasks) может входить в несколько домашних работ через homework\_tasks (m:n).

- Попытка (attempts) фиксирует прохождение домашней работы; связка «кто — что — какая попытка» хранится в user\_homework\_attempts.
- Таблица attempt\_inputs сохраняет детальные ответы пользователя по каждой задаче и результат проверки.
- При удалении пользователя каскадно удаляются созданные им курсы, задания и др., что обеспечивает referential integrity.

Такое распределение обеспечивает:

- гибкую Many-to-Many логику для курсов, домашних заданий и задач;
- прозрачное хранение попыток и статистики успеваемости;
- явное разграничение ролей и прав через user\_types.

## **Вывод**

В рамках работы была создана схема базы данных в соответствие с функциональными требованиями.