АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

«АКАДЕМИЯ ТОП»

**ПРОЕКТ**

**по дисциплине «Технология доступа к базам данных ADO.NET»**

**«Разработка системы онлайн-обучения»**

Выполнили:

студенты группы: 9/3-РПО-23/2

Карпухова А.А, Ерофеев К.П, Тулуш И.

Преподаватель:

Рослова О.А

г. Санкт-Петербург

2025

**Введение**

**Цель:** Разработать стабильное, безопасное и производительное консольное приложение, связанное с онлайн-обучением. Основная техническая особенность - эффективное использование [ado.net](http://ado.net) для взаимодействия с базой данных, обеспечивая быстрое выполнение запросов, целостность данных и безопасность от внешних угроз (sql-инъекций). В качестве ado будет использоваться LINQ .

**Задачи проекта:**

1. Разработать базу данных, заполнить таблицы и нормализовать ее.
2. Организовать работу с базой данных:
   * добавление/удаление пользователей.
   * редактирование данных таблиц.
   * просмотр данных с возможностью сортировки.
   * регистрация пользователя.
   * хэширование и восстановление пароля по почте(без отправки сообщения на почту!)
3. Создать интуитивно - понятный консольный интерфейс.
4. Протестировать консольное приложение и базу данных.
5. Написать вывод, создать презентацию и защитить проект.

**Язык программирования:** С#

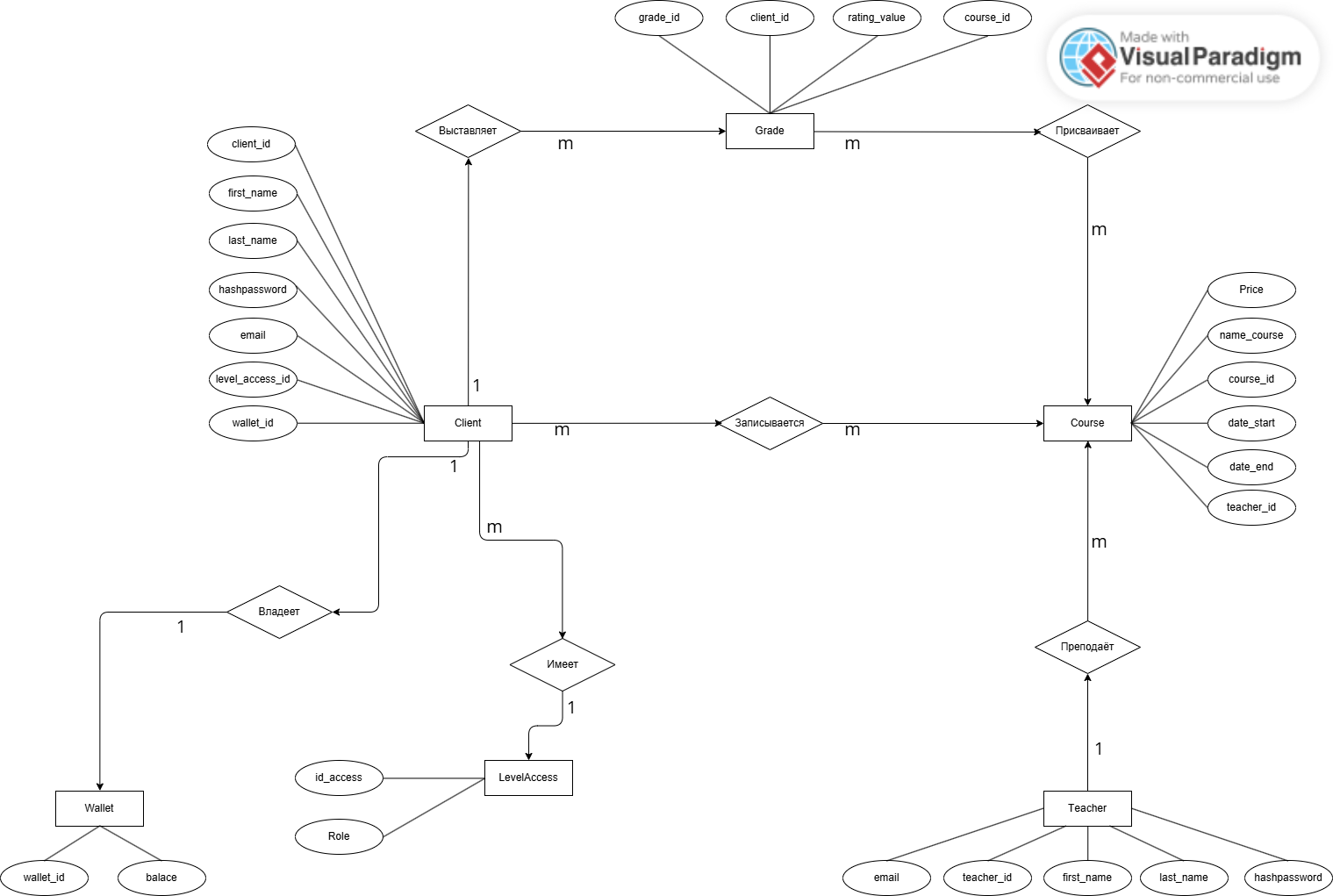
**СУБД:** MS SQL Server **+** SSMS

**Программа для реализации:** MS Visual Studio

**Объекты базы данных(сущности):** client, teacher, course, level access, wallet

**Проектирование базы данных**

Er-модель системы онлайн-обучения

****

Основные сущности и их назначение

1. Client (Клиент/Студент)

Атрибуты:

* client\_id - уникальный идентификатор клиента
* first\_name, last\_name - имя и фамилия
* email - почта
* password - хэш пароль для аутентификации
* level\_access\_id - идентификатор уровня доступа
* wallet\_id - идентификатор кошелька

2. LevelAccess (Уровень доступа)

Атрибуты:

* id\_access - идентификатор уровня доступа
* Role - роль пользователя

3. Course (Курс)

Атрибуты:

* course\_id - уникальный идентификатор курса
* name\_course - название курса
* date\_start, date\_end - даты начала и окончания курса
* teacher\_id - идентификатор преподавателя
* Price - цена на курс (цена может быть ноль)

4. Teacher (Куратор)

Атрибуты:

* teacher\_id - уникальный идентификатор куратора
* first\_name, last\_name - имя и фамилия
* email - почта
* password - хэш пароля

5. Wallet (Кошелек)

Атрибуты:

* wallet\_id - уникальный идентификатор кошелька
* balance - баланс средств

6. Grade (Оценка курса)

Атрибуты:

* grade\_id - уникальный идентификатор оценки
* client\_id - ссылка на студента
* course\_id - ссылка на курс
* rating\_value - оценка курса (от 1 до 5)

Связи между сущностями

1. Client ←→ Course: Связь многие-ко-многим. Так как один студент может записываться на множество курсов и один курс может иметь множество студентов. (Будет реализовано через составной ключ)

2. Client ←→ Wallet: Связь один-к-одному. Так как каждый студент имеет только один кошелек и каждый кошелек принадлежит одному студенту.

3. Client ←→ LevelAccess: Связь многие-к-одному. Так как многие студенты имеют только один уровень доступа.

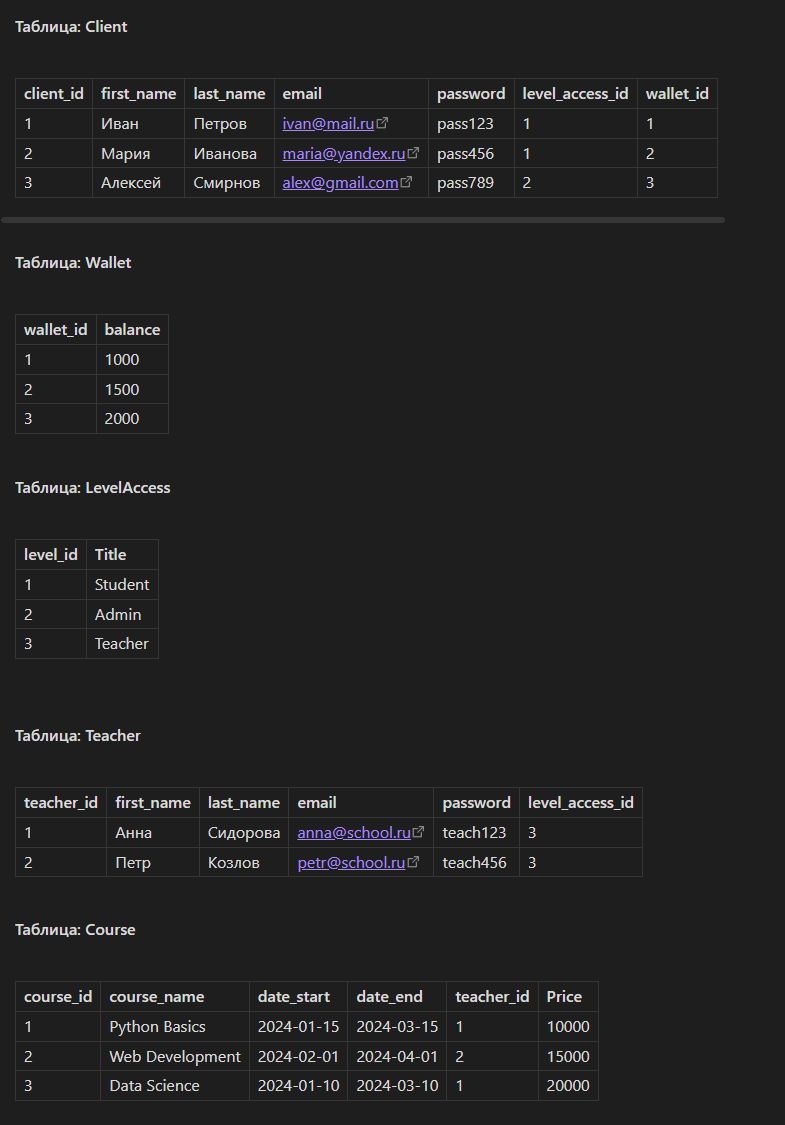
4. Teacher ←→ Course: Связь один-ко-многим. Так как один куратор может вести множество курсов.

5. Grade ←→ Client: Связь многие-к-одному. Так как один студент может оставить множество оценок разным курсам, но одна оценка принадлежит ровно одному студенту.

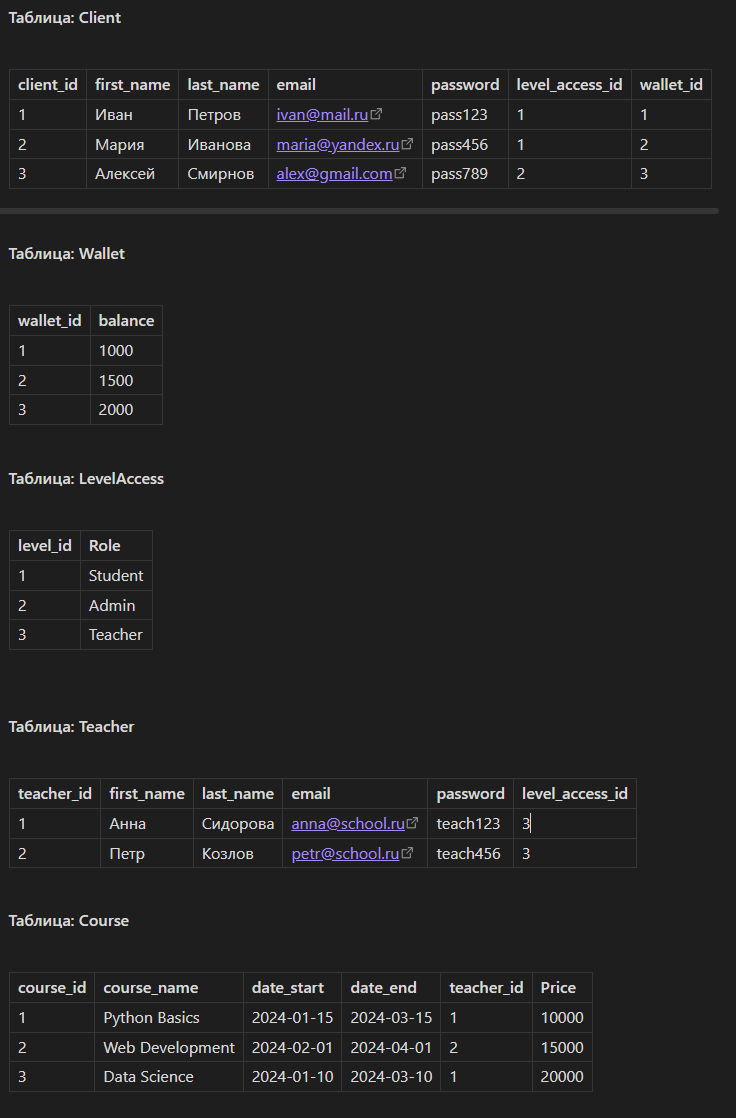
6. Grade ←→ Course: многие-к-многим. Так как один курс может получить много оценок от разных студентов и один студент может оценить много курсов. (Будет реализовано через составной ключ)

**Нормализация БД**

1. Не нормализованная таблица



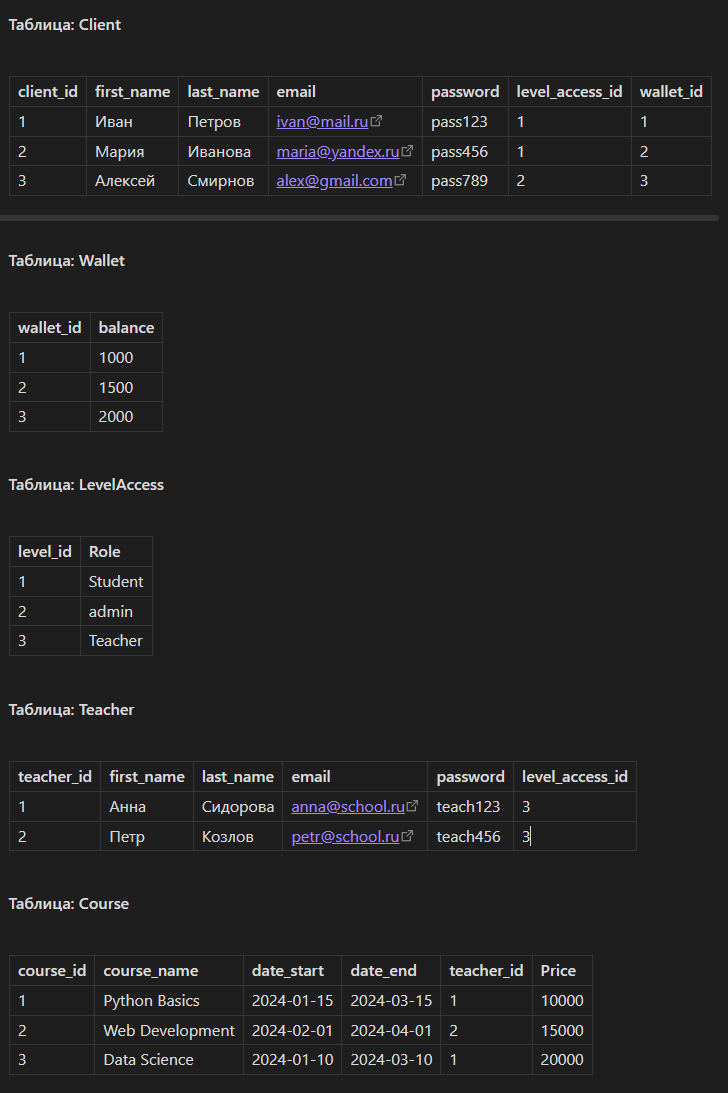
2. 1NF



Анализ исходных данных:

* Все атрибуты атомарны
* Нет повторяющихся групп данных
* Результат: Все таблицы соответствуют 1nf.

3. 2nf

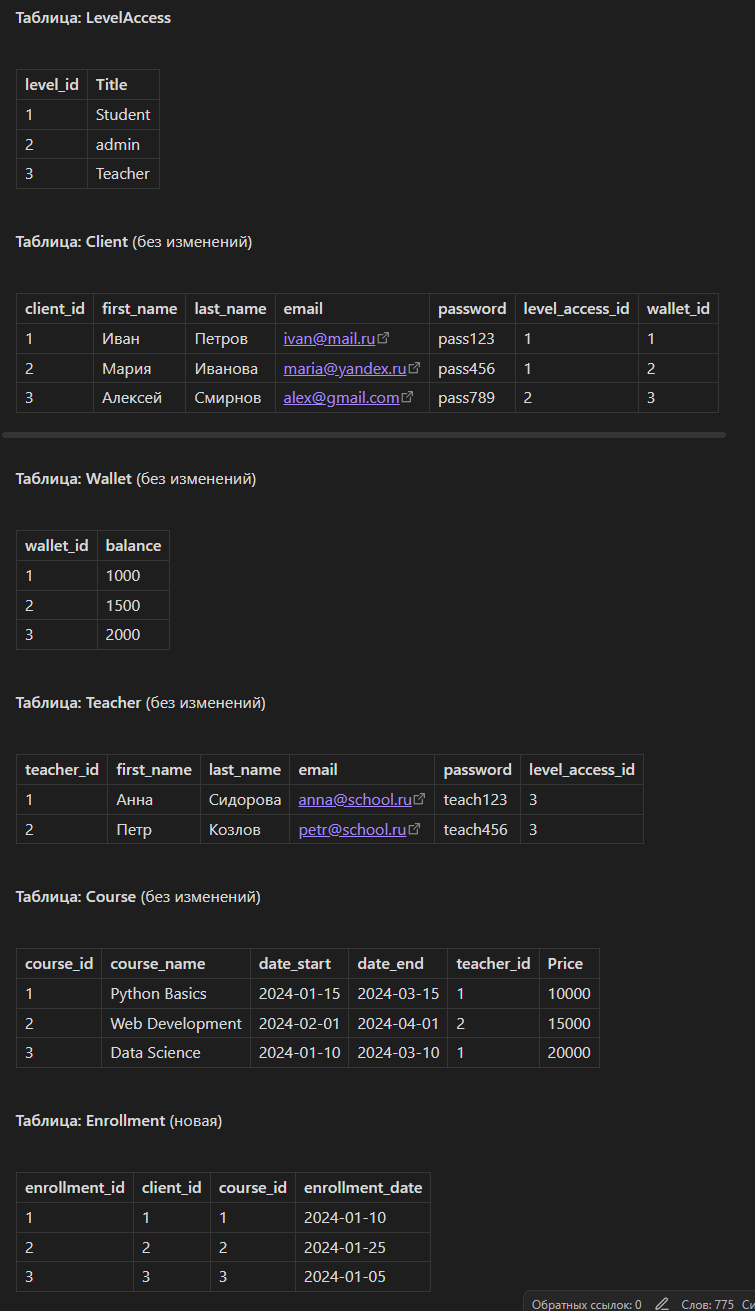


Анализ исходных данных

* Все таблицы имеют простые первичные ключи
* Все неключевые атрибуты полностью зависят от первичных ключей

Результат: Все таблицы соответствуют 2nf.

4. 3nf



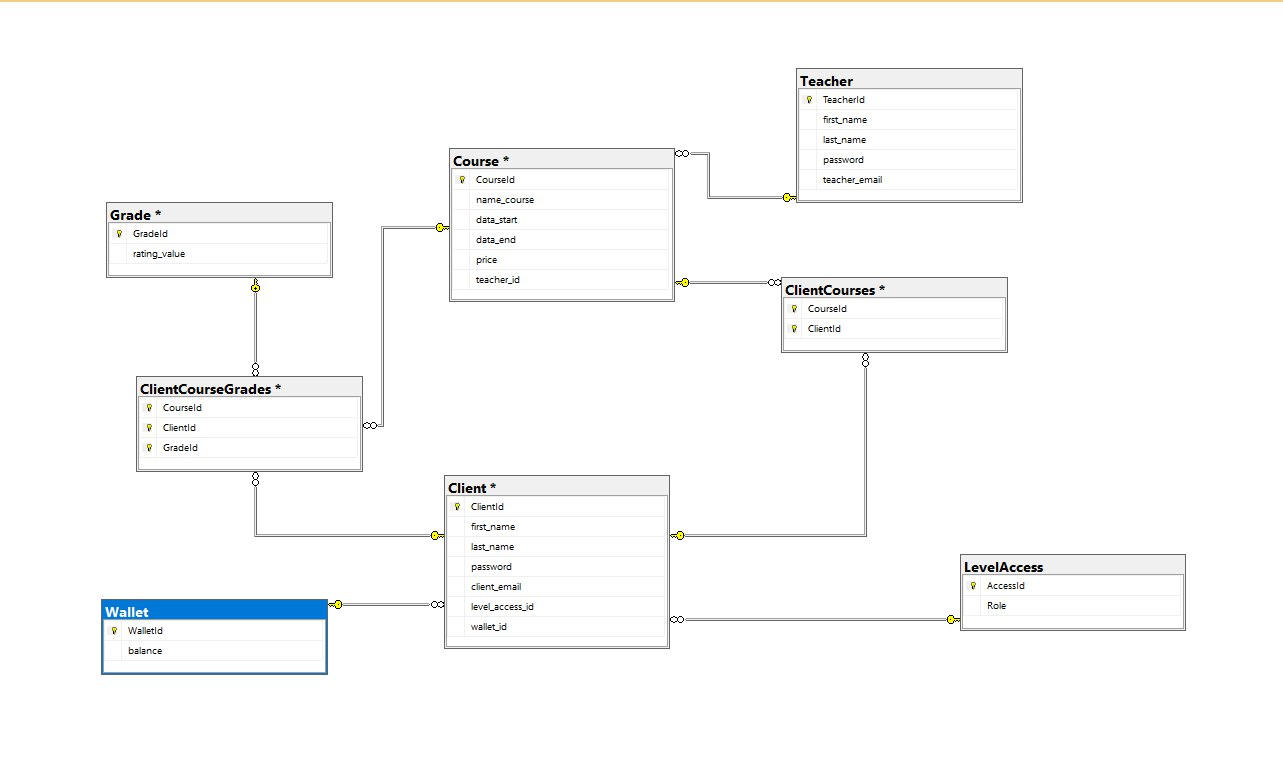
Анализ исходных данных:

* Отсутствует таблица связи многие-ко-многим между Client и Course

Исправление:

* Создана таблица Enrollment для связи клиентов и курсов

Таблица реализованная в СУБД



Убрать систему оценивания курсов! Также, добавить Role для учителя (Если учителю не нужен кошелек, то в wallet\_id пишет 0). (Добавить из-за того, что у клиента и учителя одни и те же атрибуты, но будет разный функционал).

Изменить в таблице LevelAccess атрибут Role на Title (Для более интуитивно-понятного восприятия)

(Если будет время, добавить процедуру для защиты от DDos’a)

**Таблицы для базы данных**

client:  
 id\_client int primary key,

first\_name\_clinet

last\_name\_client

password

level\_access\_id

email\_client

wallet\_id

course\_id

Level Access

id\_access

Role

course

course\_id

name\_course

start\_date\_course

end\_date\_course

teacher\_id

number\_of\_seats

occupied\_places

Price

teacher

teacher\_id

frist\_name

last\_name

email