**Проверка обновления данных в системе управления студентами.**

**Описание задания:**

Целью данной итоговой работы является тестирование операций обновления данных   
в простой системе учета студентов.

Для достижения этой цели нам необходимо решить следующие задачи:

1. Обновить имя одного из студентов и проверить изменения.
2. Перенести нескольких студентов из одной группы в другую.
3. Попробовать изменить номер группы на несуществующий номер.
4. Обновить одновременно несколько полей для одного студента.

**Ожидаемый результат:**

1. Записаны SQL-команды, которые использовались для каждого обновления.
2. Проверено с помощью SELECT-запросов, что данные действительно изменились.
3. Описаны проблемы, возникшие при выполнении заданий.

**Стек технологий:**

1. DBeaver
2. SQLite
3. DeepSeek

**Подготовка**:

С помощью ИИ-ассистента DeepSeek в DBeaver создаем базу данных SQLite и в ней создаем две таблицы: "студенты" (students) и "группы" (groups), которые заполняем простыми тестовыми данными (5 групп, 15 студентов).

**Решение:**

В процессе выполнения работы мной было принято решение объединить **пункты** **1 и 4**, поэтому мы будем одновременно менять у одного студента несколько полей, а именно его имя и email.

Для начала посмотрим список всех студентов. Для этого введем запрос:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.SELECT \* FROM students ORDER BY student\_id;  
  
Как видно из скриншота, студенты в таблице располагаются хаотично, поэтому для удобства работы сделаем их сортировку по фамилиям, с названиями и id групп, названиями факультетов, а также поменяем местами столбцы с именем и фамилией (теперь столбец с фамилиями будет стоять первым). Для этого введем запрос:

SELECT

*s*.student\_id,

*s*.last\_name,

*s*.first\_name,

*s*.email,

*g*.group\_name,

*g.group\_id,*

*g*.faculty

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

ORDER BY *s*.last\_name, *s*.first\_name;

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.Получим следующий вид:

По условиям задачи нам необходимо обновить имя одному из студентов (п.1), а также изменить одновременно несколько полей для одного студента (п.4).

Вносить изменения мы будем студенту Кузнецовой Анне (id 4), переименуем ее в Екатерину   
и обновим её email. Для этого введем запрос:

UPDATE students

SET first\_name = 'Екатерина', email = 'ekaterina.kuznetsova@example.com'

WHERE student\_id = 4;

Теперь проверим изменения.

1. Конкретно для этого студента:

SELECT \* FROM students WHERE student\_id = 4;

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Также мы можем посмотреть список всех студентов с сохранением удобной для автора сортировки.   
   Так как столбцы group\_name и faculty находятся в таблице groups, а не в таблице students, нужно использовать JOIN:

SELECT

*s*.student\_id,

*s*.last\_name,

*s*.first\_name,

*s*.email,

*s*.group\_id,

*g*.group\_name,

*g*.faculty

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

ORDER BY *s*.last\_name, *s*.first\_name;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.Получаем следующий результат:

Итак, мы видим, что у одного студента успешно изменено одновременно несколько полей (имя и email), что позволяет нам считать пункты 1 и 4 данного задания выполненными.

**П.2. Перенести нескольких студентов из одной группы в другую**.

Существует разные варианты переносов, в этой работе рассмотрим перенос по id.

Итак, мы будем переносить всех студентов из группы ИТ-102 (id студентов 3, 4, 12) в группу   
ИТ-101, а также одного студента (id 9) из группы математиков, используя для этого id студентов.

Введем запрос:

UPDATE students

SET group\_id = (SELECT group\_id FROM groups WHERE group\_name = 'ИТ-101')

WHERE student\_id IN (3, 4, 9, 12);

Проверим изменения по перенесенным студентам:

SELECT '=== СТУДЕНТЫ ПОСЛЕ ПЕРЕНОСА ===' as *''*;

SELECT

*s*.student\_id,

*s*.first\_name,

*s*.last\_name,

*g*.group\_name

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

WHERE *s*.student\_id IN (3, 4, 9, 12)

ORDER BY *s*.student\_id;

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.Получим следующее:

И можем посмотреть общий список:

SELECT

*s*.student\_id,

*s*.last\_name,

*s*.first\_name,

*s*.email,

*g*.group\_name,

*g*.group\_id,

*g*.faculty

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

ORDER BY *s*.last\_name, *s*.first\_name;

Сделаем фильтром сортировку по id группы и видим, что у нас все студенты из группы ИТ-102 перенесены в группу ИТ-101, а также математиков стало не 3, а 2, один из них   
(с id 9) также теперь находится в группе ИТ-101. Группа ИТ-102 не содержит в себе студентов в списке больше не отображается.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

На этом задание № 2 можно считать выполненным.

**П.3. Попробовать изменить номер группы на несуществующий номер**

Для начала посмотрим, какие у нас есть группы:

SELECT group\_id, group\_name FROM groups;

Изображение выглядит как текст, снимок экрана, дисплей, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Теперь выберем группу для эксперимента. Пусть это будет ХИМ-401 (id 5).

Попробуем изменить id группы:

UPDATE groups SET group\_id = 10 WHERE group\_id = 5;

Проверяем результат:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.SELECT group\_id, group\_name FROM groups;

Видим, что id группы изменился.

Теперь проверим студентов в этой группе:

SELECT student\_id, first\_name, last\_name, group\_id

FROM students

WHERE group\_id = 10;

Изображение выглядит как снимок экрана, текст, линия, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Ой :(** Внутри никого нет:

Попробуем найти студентов с их группами:

SELECT

*s*.first\_name,

*s*.last\_name,

*g*.group\_name

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

WHERE *s*.group\_id = 10;

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.И снова никого нет:

Попробуем вернуть группе ее старый id:

UPDATE groups SET group\_id = 5 WHERE group\_id = 10;

Посмотрим, что получилось:

SELECT

s.first\_name,

s.last\_name,

g.group\_name

FROM students s

JOIN groups g ON s.group\_id = g.group\_id

WHERE s.group\_id = 5;

Изображение выглядит как текст, снимок экрана, дисплей, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Уф, нашлись.

Теперь посмотрим общий список:

SELECT

*s*.student\_id,

*s*.last\_name,

*s*.first\_name,

*s*.email,

*g*.group\_name,

*g*.group\_id,

*g*.faculty

FROM students *s*

JOIN groups *g* ON *s*.group\_id = *g*.group\_id

**Изображение выглядит как текст, программное обеспечение, число, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.**ORDER BY *s*.last\_name, *s*.first\_name;

Как видим, все вернулось как было и все студенты на своих местах**.**Какие можно сделать выводы из эксперимента по изменению id группы   
на несуществующий:

1. Группа получит новый ID (10)
2. Студенты останутся привязаны к старому ID (5), но такой группы больше нет.
3. Студенты "потеряются", их нельзя будет найти через обычные запросы.

**Выводы**

В ходе работы была создана простая база данных для учета студентов и протестированы различные операции обновления данных. Тесты успешно охватывают такой функционал как:

1. Обновление имени одного из студентов.
2. Перенесение нескольких студентов из одной группы в другую.
3. Изменение номера группы на несуществующий номер.
4. Одновременное обновление нескольких полей для одного студента.

**Ожидаемый результат**

1. Записаны SQL-команды, которые использовались для каждого обновления.
2. Проверено с помощью SELECT-запросов, что данные действительно изменились.
3. Описаны проблемы, возникшие при выполнении заданий.

Результаты тестирования демонстрируют корректную работу основных SQL-запросов, все тесты выполняются без ошибок, возвращая ожидаемые коды ответов.