Ижевск, 2025

Тестирование запросов в базе данных

петухова ирина сергеевна

Оглавление

[Введение 2](#_Toc209015061)

[Используемые инструменты 3](#_Toc209015062)

[Создание базы данных 4](#_Toc209015063)

[Тестирование запросов в базе данных 5](#_Toc209015064)

[Результаты тестирования 8](#_Toc209015065)

[Заключение 9](#_Toc209015066)

# Введение

В данной работе я тестировала простые запросы в базе данных Университета. Для этого создала базу данных Университета, заполнила данными о студентах, предметах и группах, написала различные запросы и составила чек-листы для проверки этих запросов.

# Используемые инструменты

1. СУБД: Postgresql
2. Графический редактор: pgAdmin 4
3. Docker - для запуска базы данных в контейнере.
4. Нейросеть Deepseek – для создания таблиц и заполнения их данными.

# Создание базы данных

1. В pgAdmin я создала базу данных DB\_college\_project.
2. С помощью нейросети Deepseek написала запрос на создание таблиц "Студенты", "Группы" и "Предметы" (CREATE TABLE) и заполнила таблицы данными (INSERT INTO).
3. В редакторе Query Tool запустила запросы, полученные через нейросеть.
4. Написала запрос на создание таблицы для связи предметов с группами.

# Тестирование запросов в базе данных

Для тестирования базы данных я выполнила проверку корректности выполнения запросов SELECT, INSERT, UPDATE, DELETE.

Для этого вначале я составила чек-листы, сгруппированные по этим четырем типам операций.

Для проверки корректности выполнения запросов буду использовать метод написание дополнительных запросов для проверки количества строк.

Тестируемый SELECT-запрос:

*-- 8. Джойн (JOIN)*

*-- Вывод всех студентов в группе "Экономика"*

*SELECT s.last\_name AS Фамилия, s.first\_name AS Имя*

*FROM students s*

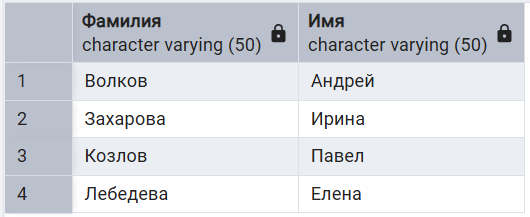
*INNER JOIN groups g ON s.group\_id = g.group\_id*

*WHERE g.faculty = 'Экономика'*

*ORDER BY s.last\_name*

В данном примере проверяется корректность выполнения JOIN запроса. Из таблицы «Студенты», с помощью присоединенной таблицы «Группы», выводятся все студенты из группы «Экономика».

Результат запроса выглядит так:



При заполнении таблицы «Студенты» данными, из всех студентов я указала 4 студента из группы «Экономика», то есть с group\_id = 3, поэтому ожидаемое число строк в проверочном запросе должно соответствовать четырем.

Проверочный запрос для сравнения количества:

*SELECT*

*(SELECT COUNT(\*)*

*FROM students s*

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

*WHERE g.faculty = 'Экономика'*

*) as actual\_count,*

*4 as expected\_count; -- Ожидается 4 строки*

При выполнении проверочного запроса вышла таблица с фактическим и ожидаемым числом строк:

Изображение выглядит как текст, Шрифт, линия, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Числа соответствуют друг другу, следовательно тестируемый SELECT-запрос написан корректно.

Также для проверки корректности запроса можно воспользоваться техникой визуального сравнения.

Для этого:

* 1. Выведу все записи из таблицы «Студенты»: *select \* from students;*
  2. Посчитаю число строк с group\_id равным 3;
  3. Зафиксирую значения имен и фамилий в этих полях;
  4. Зафиксирую эти данные в алфавитном порядке фамилий, т.к. в запросе есть сортировка ORDER BY.

Тестируемый INSERT-запрос:

*-- 2. Добавление данных в таблицу без указания столбцов*

*-- Добавление новой группы*

*INSERT INTO groups*

*VALUES (6, 'ПМ-301', 'Прикладная математика', 3);*

Для тестирования INSERT-запроса воспользуюсь техникой визуального сравнения. Для этого с помощью запроса *select \* from groups* выведу все записи и верю корректно ли добавилась новая запись.

В результате выполнения запроса вышла такая таблица:

Изображение выглядит как текст, Шрифт, число, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Новая запись успешно добавлена в таблицу, все значения соответствуют ожидаемому результату, что свидетельствует о том, что запрос INSERT выполнен корректно.

Тестируемый UPDATE-запрос:

*-- 6. Внешние ключи (FOREIGN KEY)*

*-- Изменение номера группы на несуществующий номер*

*UPDATE groups*

*SET group\_id = 14*

*WHERE group\_id = 2*

Нужно проверить, что изменение поля, которое является внешним ключом в другой таблице, вызовет ошибку.

Таблицы «Студенты» и «Группы» связаны внешним ключом group\_id. В группе с group\_id = 2 присутствуют студенты, в таком случаи ID группы изменить нельзя.

При выполнении запроса возникает ошибка:

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Здесь сообщается, что ключ (group\_id)=(2) всё ещё используется из таблицы «students». Данная ошибка соответствует ожидаемому результату.

Тестируемый DELETE-запрос:

*-- 3. Обработка условий (WHERE)*

*-- Удаление студента с несуществующей группой*

*DELETE FROM students*

*WHERE group\_id = 20*

При выполнении данного запроса нужно проверить, что если условие WHERE не совпало ни с одной строкой, то ни одна запись не должна быть удалена.

В таблице «Студенты» значения *group\_id = 20* не существует и не одна запись не должна быть удалена.

Для проверки корректности выполнения запросов также буду использовать метод написание дополнительных запросов для проверки количества строк:

*select COUNT(\*) from students;*

Результат равен 16. Далее выполню запрос на удаление строки и заново выполню запросов для проверки количества строк. При выполнении последнего запроса число не изменилось, что соответствует ожидаемому результату и корректности запроса на удаление с несуществующим полем.

# Результаты тестирования

Из 28 проверок, при прохождении по чек-листу, было выявлено 3 дефекта:

* Была допущена логическая ошибка в SELECT-запросе №3: вместо = NULL необходимо использовать is NULL, так как для проверки значения NULL (пустого значения) не используется оператор «равно». Фактический результат: вывод пустой таблицы. Ожидаемый результат: таблица со строками, где email пустой.
* Допущена математическая ошибка в SELECT-запросе №7: SUM() должно быть COUNT(), так как SUM() вычисляет сумму значений, то есть будет считать сумму чисел student\_id, а COUNT() считает количество строк, то есть будет выводить количество студентов. Фактический результат: 504. Ожидаемый результат: 16.
* Допущена синтаксическая ошибка в UPDATE-запросе №2: некорректно указано название таблицы. Вместо students указано student. Фактический результат: ERROR: relation "student" does not exist. Ожидаемый результат: UPDATE 1, данные успешно обновлены.

# Заключение

В данной работе я тестировала простые запросы в базе данных Университета. Для этого создала базу данных Университета, заполнила данными о студентах, предметах и группах, написала различные запросы и составила чек-листы для проверки этих запросов.

По итогам тестирования из 28 проверок было выявлено 3 дефекта. Для их устранения необходимо исправить логику запроса, математические и синтаксические ошибки.