Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №9**

|  |
| --- |
| Программирование на стороне сервера в среде СУБД PostgreSQL |

тема

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | А. Д. Вожжов |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ23-17/2б, 032320981 |  |  |  | Р. А. Троицкий |
|  | номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2025

Содержание

[1 Цели 3](#_Toc190097585)

[2 Задачи 3](#_Toc190097586)

[3 Ход выполнения 6](#_Toc190097587)

[4 Выводы 12](#_Toc190097588)

1. Цели

Ознакомиться с методами повышения производительности.

1. Задачи

Для выполнения практической работы необходимо выполнить следующие задачи.

1. Вызов функции можно использовать в подзапросе в предложении FROM. Напишите такой запрос с подзапросом на примере функции generate\_students\_data().   
 2. Для чего нужны модификаторы (ключевые слова) IN и OUT перед именами параметров функций?   
3. Каким образом можно сделать так, чтобы функция возвратила табличное значение?

4. Что такое триггер?

5. Чем триггеры уровня строки (row-level) отличаются от триггеров уровня команды (statement-level)?

6. В базе данных ais создайте таблицы, функции и триггеры, необходимые для изучения метода хранения иерархий в реляционных базах данных. Для этого можно поступить следующим образом. Сначала создайте текстовый файл с именем, например, adj\_list.sql, содержащий все команды и определения функций, приведенные в тексте пособия, а затем выполните такую команду в среде операционной системы: psql -d ais -f adj\_list.sql

7. Выполните проверку структуры дерева на предмет отсутствия циклов с помощью функции tree\_test(). SELECT \* FROM tree\_test(); Если вы еще не вносили изменения в таблицу «Организационная структура», то функция покажет отсутствие нарушения структуры дерева. Теперь создайте в таблице «Организационная структура» сначала короткий цикл, а затем длинный цикл. Для каждого из указанных циклов выполните проверку с помощью функции tree\_test().

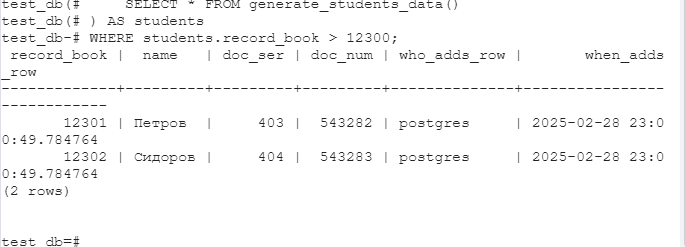
8. Выполните операцию удаления поддерева с помощью функции delete\_subtree(). Параметром функции является код работника. SELECT \* FROM delete\_subtree( 6 ); Аналогично работе с функцией up\_tree\_traversal() используйте подзапрос для получения кода работника по его имени. После удаления поддерева посмотрите, что стало с организационной структурой, с помощью двух представлений Personnel\_org\_chart и Create\_paths.

9. Представление Create\_paths позволяет отобразить только четыре уровня иерархии. Модифицируйте его так, чтобы оно могло работать с пятью уровнями иерархии.

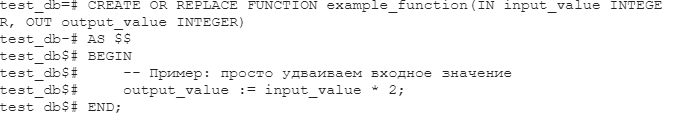
10. Самостоятельно ознакомьтесь с таким средством работы с таблицами базы данных, как правила (rules). Воспользуйтесь технической документацией на PostgreSQL, глава «The Rule System». Напишите правила, позволяющие организовать журнальные таблицы в базе данных «Студенты».

1. Ход выполнения

Выполнил пункт 1 задач.



Выполнил пункт 2 задач.



IN:

* Это модификатор по умолчанию, если не указано иное.
* Указывает, что параметр используется для ввода данных в функцию.
* Значения параметров IN передаются в функцию, и они не могут быть изменены внутри функции. Они используются только для чтения.

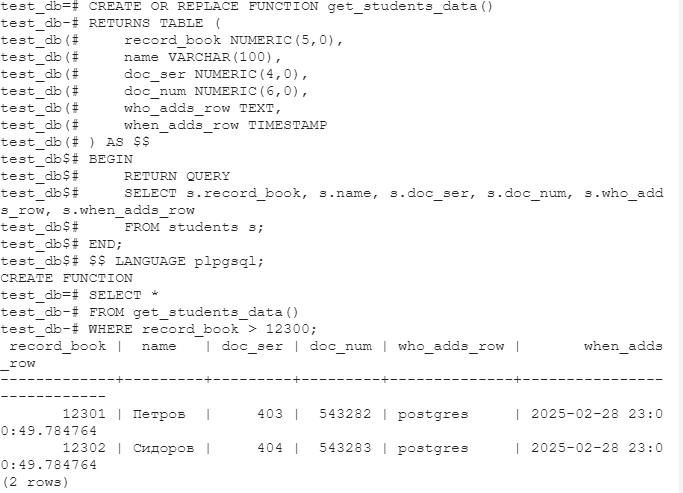
OUT:

* Указывает, что параметр используется для вывода данных из функции.
* Параметры OUT позволяют функции возвращать несколько значений. Они действуют как дополнительные возвращаемые значения.
* Значения параметров OUT устанавливаются внутри функции, и они возвращаются вызывающему коду.

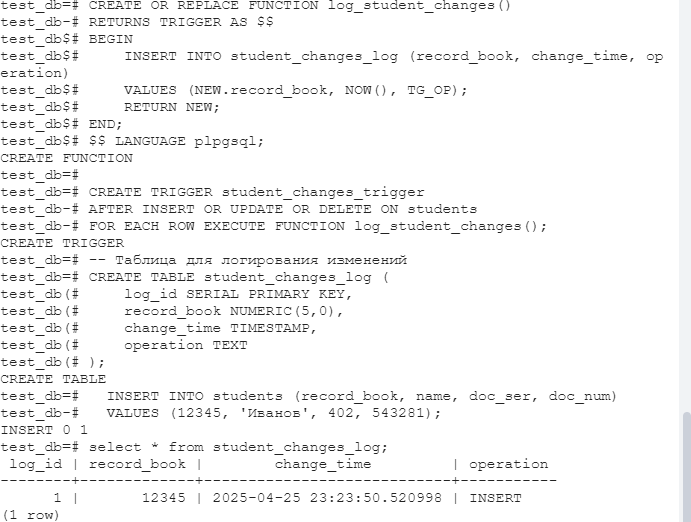
INOUT:

* Комбинирует поведение IN и OUT.
* Параметры INOUT принимают начальное значение при вызове функции и могут быть изменены внутри функции. Изменённые значения возвращаются вызывающему коду.

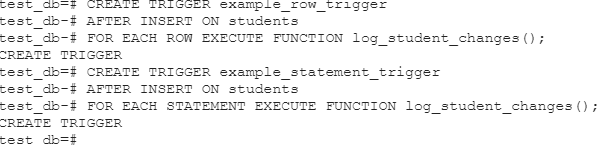
Выполнил пункт 3 задач

  
  
Чтобы функция в PostgreSQL возвращала табличное значение, вы можете использовать ключевое слово RETURNS TABLE в определении функции.

Выполнил пункт 4 задач



Триггер в базе данных — это специальный объект, который автоматически выполняет заданный набор действий (обычно это SQL-операторы) в ответ на определённые события, происходящие в таблице или представлении. Эти события могут включать операции вставки (INSERT), обновления (UPDATE) или удаления (DELETE) строк.

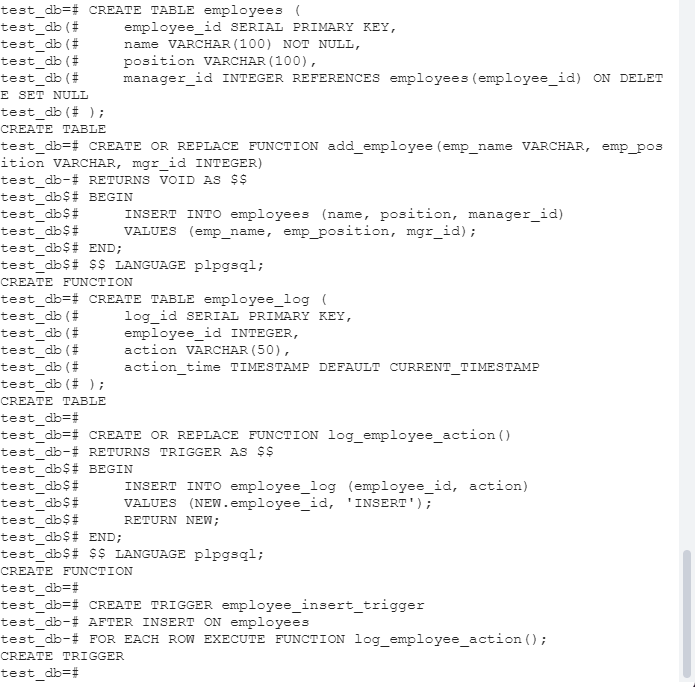
Выполнил пункт 5 задачТриггеры уровня строки (Row-Level Triggers)

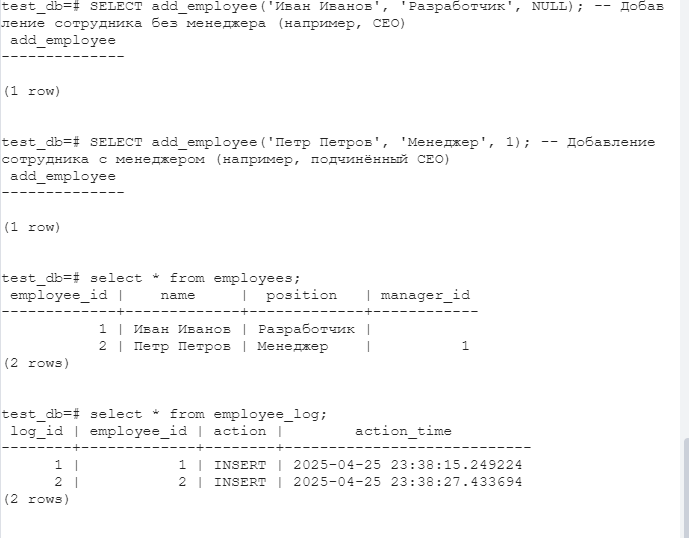
* Срабатывание: Триггер уровня строки срабатывает для каждой отдельной строки, затронутой SQL-операцией (INSERT, UPDATE, DELETE).
* Использование: Обычно используется, когда необходимо выполнить действие для каждой изменяемой строки, например, логирование изменений или валидация данных.
* Доступ к данным: В теле триггера доступны специальные переменные NEW и OLD, которые содержат значения новой и старой строки соответственно.

Триггеры уровня команды (Statement-Level Triggers)

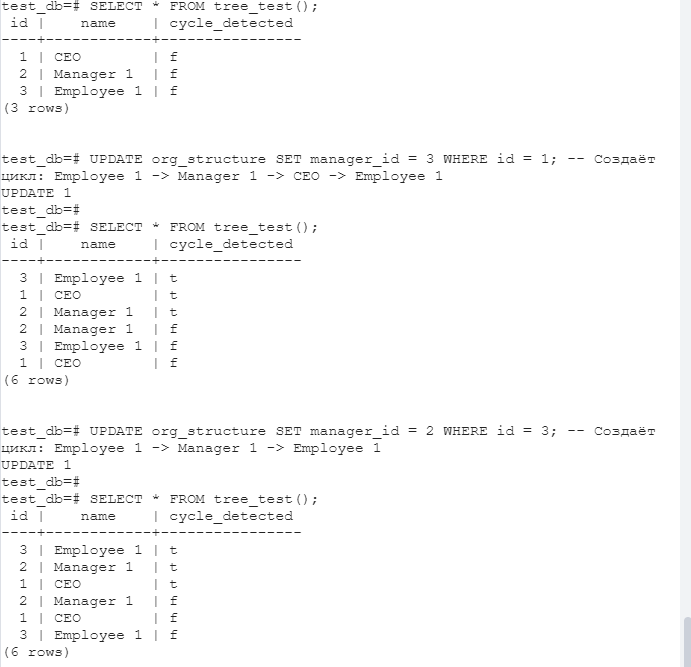
* Срабатывание: Триггер уровня команды срабатывает один раз для всей SQL-операции, независимо от того, сколько строк было затронуто.
* Использование: Подходит для задач, которые должны выполняться один раз за операцию, например, обновление агрегированных данных или выполнение действий, не зависящих от конкретных строк.
* Доступ к данным: В теле триггера нет доступа к NEW и OLD, так как он не привязан к конкретным строкам.

Выполнил пункт 6 задач

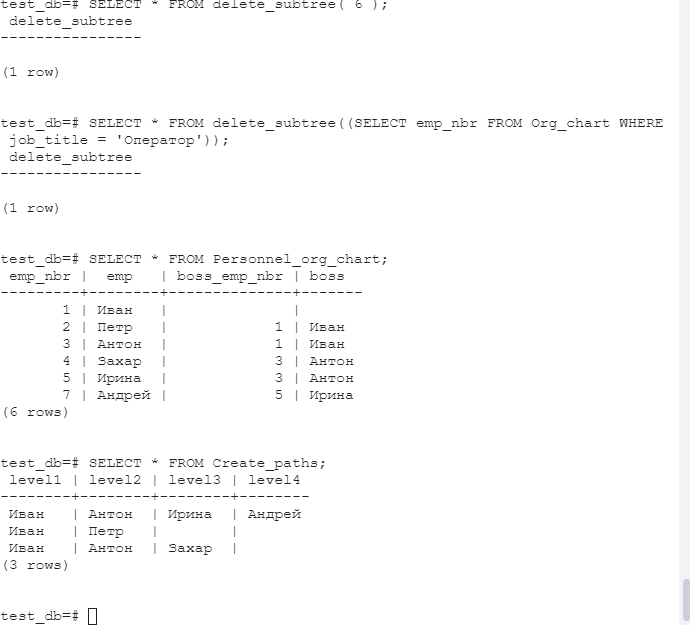




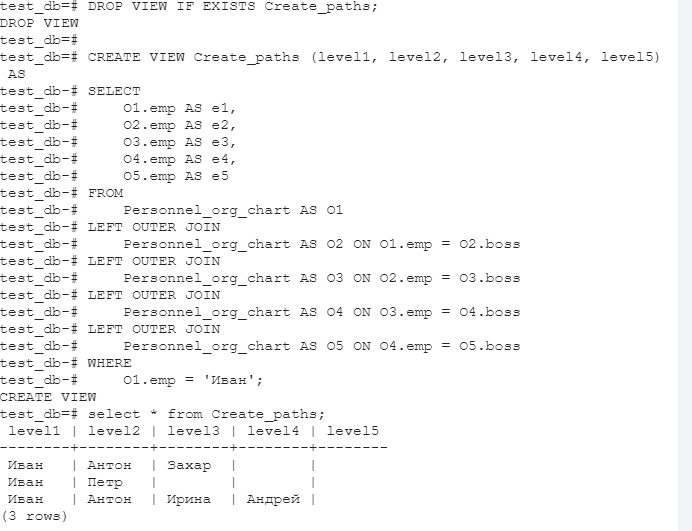
Выполнил пункт 7 задач



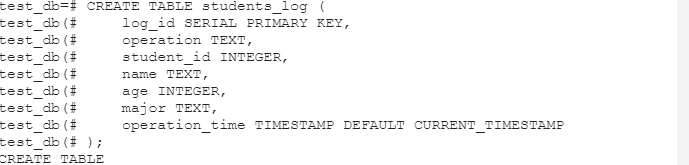
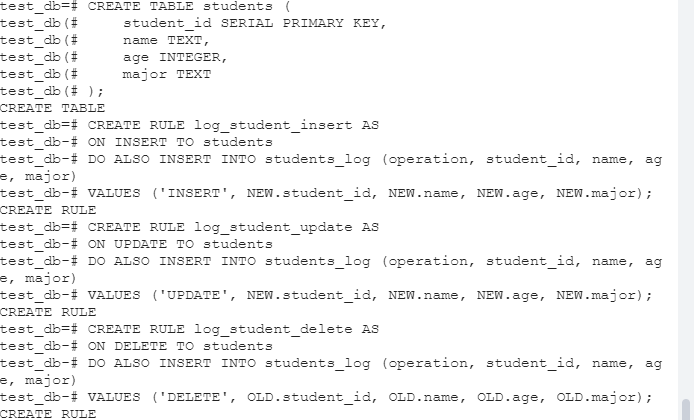
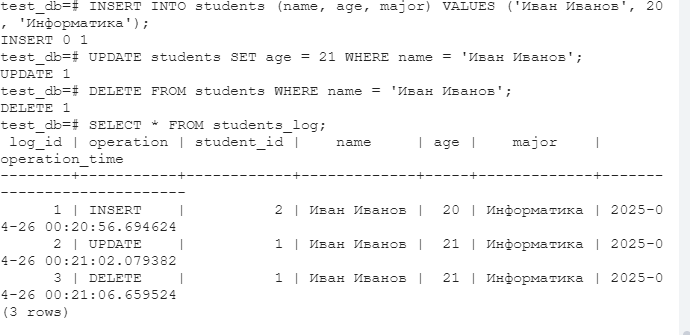
Выполнил пункт 8 задач



Выполнил пункт 9 задач



Выполнил пункт 10 задач

4 Выводы

Выполнив задание, было изучено программирование на стороне сервера в среде СУБД PostgreSQL.