Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Информатика |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | Е. П. Моргунов |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/2Б, 032156940 |  |  |  | Н. А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

1. Вызов функции можно использовать в подзапросе в предложении  
FROM. Напишите такой запрос с подзапросом на примере функции  
generate\_students\_data().

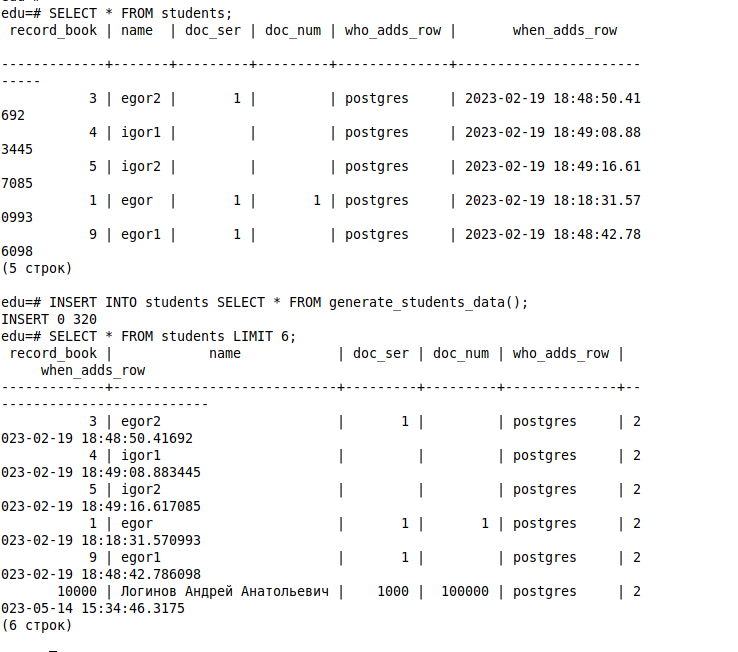


Рисунок 1 – Выполнение задания 1

3. Для чего нужны модификаторы (ключевые слова) IN и OUT перед  
именами параметров функций?

Функция может получать аргументы. Эти аргументы могут иметь  
модификаторы IN и OUT. Первый из них используется по умолчанию и  
означает параметр, значение которого будет изменяться внутри функции, но  
вне функции его новое значение не будет доступно. Параметр с ключевым  
словом OUT используется для того, чтобы возвратить из функции более одного  
значения.

5. Каким образом можно сделать так, чтобы функция возвратила  
табличное значение?

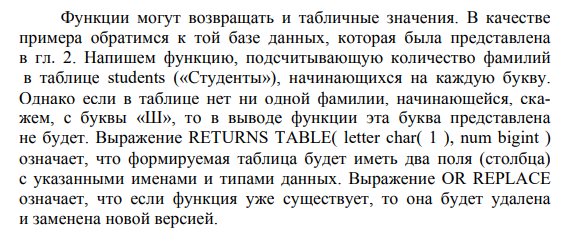


Рисунок 2 – Выполнение задания 5

7. Что такое триггер?

Триггер – это механизм, заставляющий СУБД выполнить конкретную  
функцию, когда выполняется определенный тип операций. Триггеры могут  
быть связаны с таблицами и представлениями (views). Триггеры, связанные с  
таблицами, могут выполняться как ДО (BEFORE), так и ПОСЛЕ (AFTER)  
операций INSERT, UPDATE, DELETE. При этом возможно выполнение  
триггера либо для каждой модифицированной строки таблицы, либо  
однократно для SQL-команды в целом. Если случается конкретное событие,  
приводящее к срабатыванию триггера, то вызывается так называемая  
триггерная функция, которая и обрабатывает это событие.

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

Триггеры могут быть двух типов с точки зрения числа повторных  
вызовов триггерной функции: – функция может вызываться для каждой строки,  
на которую влияет (affected) команда, вызвавшая срабатывание триггера; –  
функция может вызываться только один раз, независимо от числа строк,  
подвергшихся воздействию команды, вызвавшей срабатывание триггера. Даже  
если таких строк не будет ни одной, триггерная функция вызывается все равно.  
Триггеры первого типа называют триггерами уровня строки (row-level), а  
триггеры второго типа – триггерами уровня команды (statement-level).  
Триггерные функции, вызываемые триггерами уровня SQL-команды, должны  
всегда возвращать значение NULL. Триггерные функции, вызываемые  
триггерами уровня строки (row-level), могут возвращать строку таблицы, если  
это необходимо с точки зрения логики этой функции. Триггер уровня строки,  
выполняемый ДО операции, может принимать одно из двух решений: – он  
может возвратить значение NULL, чтобы предотвратить операцию с текущей  
строкой таблицы, для которой и вызван этот триггер (например, вставку,  
обновление или удаление строки таблицы); – при выполнении операций  
вставки или обновления триггер уровня строки может модифицировать  
вставляемую или обновляемую строку таблицы, поскольку именно строка,  
возвращаемая триггером, и будет вставлена в таблицу или обновлена в ней.  
Поэтому триггер такого типа, если не планируется отмена операции или  
модифицирование строки таблицы, должен возвратить неизмененную строку  
таблицы. В случае операции вставки (INSERT) и обновления (UPDATE) это  
будет специальное значение NEW, а в случае операции удаления строки  
(DELETE) это будет специальное значение OLD. Для триггеров уровня строки,  
выполняемых после операции, значение, возвращаемое триггерной функцией,  
просто игнорируется, поэтому они могут возвращать значение NULL.

11. В базе данных ais создайте таблицы, функции и триггеры,  
необходимые для изучения метода хранения иерархий в реляционных базах  
данных.

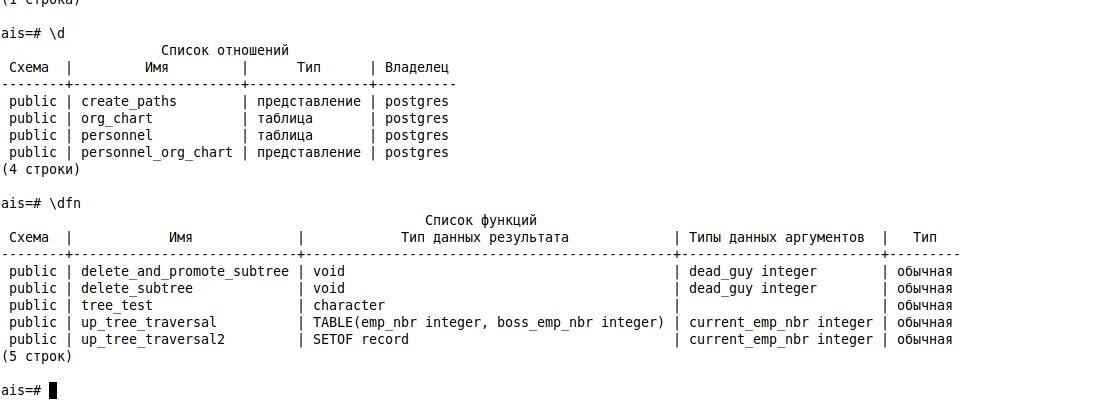


Рисунок 3 – Выполнение задания 11

13. Выполните проверку структуры дерева на предмет отсутствия циклов  
с помощью функции tree\_test().

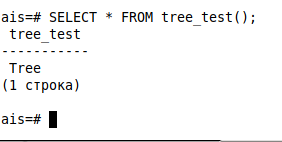


Рисунок 4 – Выполнение задания 13

15. Выполните операцию удаления поддерева с помощью функции  
delete\_subtree(). Параметром функции является код работника.

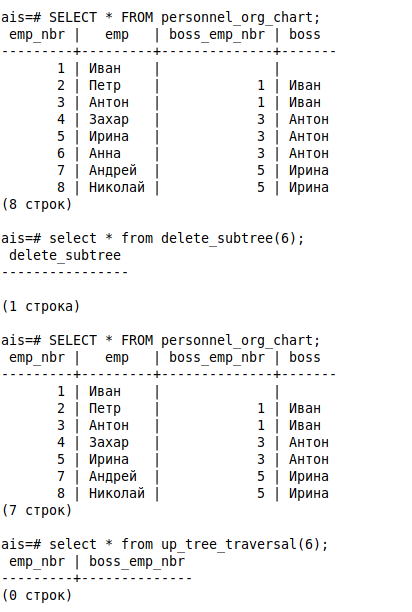


Рисунок 5 – Выполнение задания 15

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

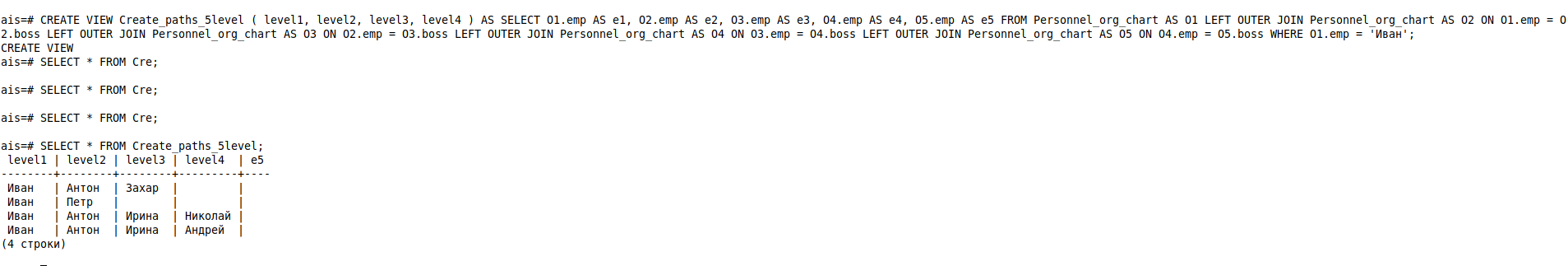


Рисунок 6 – Выполнение задания 17

19. 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».

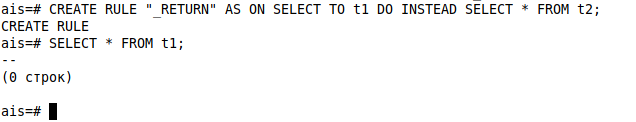


Рисунок 7 – Выполнение задания 19