МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
 РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра математического и аппаратного обеспечения   
 информационных систем

Лабораторная работа №4 по дисциплине “Безопасность систем баз данных”

Тема:Серверное программирование:функции,процедуры

Выполнил: студент гр. ИВТ 21-22

Иванов Н.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил

Старший преподаватель

Марков А.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Цель работы:**

1. Изучить способы создания, синтаксис и исполнение функций в PostgreSQL, создать запросы согласно заданию.
2. Изучить способы создания, синтаксис и исполнение процедур в PostgreSQL, создать запросы согласно заданию.

**Задание:**

1. Создать и запустить скрипт create\_functions.sql, который создаёт следующие функции:

- функция, возвращающая сумму двух чисел;

- функция, возвращающая дату, которая будет послезавтра;

- функция, возвращающая сумму всех проведённых платежей;

- функция с параметром id платежа, которая создаёт 'противоположный' платёж;

- функция с параметрами id клиента, дата1, дата2, возвращающая все исходящие платежи данного клиента (дата, сумма, кому) между указанными датами;

- функция с параметрами id клиента1, id клиента2, возвращающая количество платежей между этими клиентами;

- функция с параметрами дата, сумма, id плательщика, id получателя, которая создаёт запись в таблице payments;

- функция с параметрами сумма, id плательщика, id получателя, которая создаёт запись в таблице payments и меняет поле баланс в таблице Счета на значение суммы для счета плательщика и счёта получателя;

- функция с параметрами дата и количество(n), которая создаёт n записей в таблице payments, с датой из параметра дата, а остальные поля генерирует случайным образом(но поддерживая целостность ссылок), для генерации случайных значений использовать функцию random() - о функции здесь: https://postgrespro.ru/docs/postgresql/14/functions-math, https://oracleplsql.ru/random-postgresql.html;

2. Создать и запустить скрипт perform\_functions, который выполняет функции из п.1 с параметрами, демонстрирующими работоспособность и правильность функций.

3. Создать и запустить скрипт drop\_functions .sql, который удаляет функции из п.1.

1. Создать и запустить скрипт create\_procedures.sql, который создаёт следующие процедуры:

- процедура с IN параметрами дата, сумма, id плательщика, id получателя, которая создаёт запись в таблице payments;

- процедура с IN параметрами сумма, id плательщика, id получателя и OUT параметром new\_id, которая создаёт запись в таблице payments, меняет поле баланс в таблице Счета на значение суммы для счета плательщика и счёта получателя, а также возвращает id новой записи;

- процедура с IN параметрами сумма, id плательщика, id получателя, дата платежа, и OUT параметром new\_id, которая создаёт запись в таблице payments, меняет поле баланс в таблице Счета на значение суммы для счета плательщика и счёта получателя, возвращает id новой записи, а также создаёт такую же запись в таблице archive\_payments, если дата платежа более чем на 2 недели старше сегодня;

- процедура с IN параметрами дата и количество(n), которая создаёт n записей в таблице payments, с датой из параметра дата, а остальные поля генерирует случайным образом(но поддерживая целостность ссылок), для генерации случайных значений использовать функцию random() - о функции здесь: https://postgrespro.ru/docs/postgresql/14/functions-math, https://oracleplsql.ru/random-postgresql.html;

- процедура с IN параметром id клиента и OUT параметрами is\_correct и delta, которая проверяет, корректно ли заполнено поле баланс для всех счетов данного клиента и возвращает FALSE или TRUE в is\_correct и разницу в полученных значениях баланса в delta;

- процедура с IN параметрами Название, id родителя, которая создаёт запись в таблице pages(если id родителя NULL, то и в создаваемой записи id родителя NULL) и OUT параметром new\_id, с помощью процедуры создать несколько записей в таблице pages, в соответствии с иерархией

- Главная

- Платежи

- Счета

- Исходящие платежи

- Входящие платежи

- Архивные счета

- Офисы

- Клиенты

- процедуры с IN параметром Название и OUT параметром root\_id, которая возвращает самый верхний уровень в иерархии для данного названия, например, для Исходящие платежи – Главная, для Клиенты – Офисы, для Счета – Главная, для Офисы – Офисы и т.д.

2. Создать и запустить скрипт perform\_procedures, который выполняет функции из п.1 с параметрами, демонстрирующими работоспособность и правильность функций.

3. Создать и запустить скрипт drop\_procedures .sql, который удаляет функции из п.1.

**Результат:**  
https://github.com/bwl-andromeda/db/tree/lab-4

**Вывод по работе:**

В ходе выполнения лабораторной работы №4 мы глубоко погрузились в мир серверного программирования PostgreSQL. Мы изучили два важных инструмента: функции и процедуры.

Начали мы с функций - этаких мини-программ внутри базы данных. Научились их создавать, вызывать и даже удалять. Оказалось, что функции - это очень гибкий инструмент, который можно использовать для самых разных задач: от простого сложения чисел до сложных операций с данными.

Затем мы перешли к процедурам. Хотя они похожи на функции, у них есть свои особенности. Например, они не возвращают значения напрямую, но могут изменять данные в базе. Мы научились создавать процедуры для различных задач, включая обработку платежей и работу с иерархическими данными.

Практическая часть работы была особенно полезной. Мы не просто читали теорию, а своими руками создавали функции и процедуры, тестировали их работу, а потом аккуратно удаляли. Это дало нам реальное понимание того, как работает серверное программирование в PostgreSQL.

Важно отметить, что эти инструменты не просто игрушки для разработчиков. Они реально помогают обеспечивать безопасность баз данных. Например, используя функции и процедуры, мы можем лучше контролировать доступ к данным и выполнять сложные операции, не рискуя целостностью базы.

В целом, эта лабораторная работа дала нам практические навыки, которые пригодятся в реальной работе с базами данных. Мы научились не только создавать функции и процедуры, но и понимать, когда и зачем их использовать. Это важный шаг в понимании того, как можно эффективно и безопасно управлять данными в PostgreSQL.