**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Направление подготовки: 11.03.03**

**Образовательная программа: Безопасность информационных технологий**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

« Защита базы данных »

**Выполнил:**

*A blue logo with a white background

Description automatically generated*Рядовой Т.С., студент группы N3352, поток ИББД.Nб3 1.5

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Салихов Максим Русланович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(дата)

Содержание

[Содержание 2](#_Toc192099756)

[Введение 3](#_Toc192099757)

[1 Ход работы 4](#_Toc192099758)

[1.1 Задача №1 – мониторинг БД 4](#_Toc192099759)

[1.1.1 Создание лог-таблицы 4](#_Toc192099760)

[1.1.2 Создание функции и триггеров 4](#_Toc192099761)

[1.1.3 Пример работы 5](#_Toc192099762)

[1.2 Задача №2 – шифрование данных 6](#_Toc192099763)

[1.2.1 Создание таблицы и шифрование данных 6](#_Toc192099764)

[1.2.2 Пример работы 6](#_Toc192099765)

[1.3 Задача №3 – разграничение доступа к БД 7](#_Toc192099766)

[1.3.1 Создание ролей 7](#_Toc192099767)

[1.3.2 Настройка привилегий 8](#_Toc192099768)

[1.3.3 Пример работы 8](#_Toc192099769)

[Заключение 10](#_Toc192099770)

[Список источников 11](#_Toc192099771)

Введение

Цель работы: получение навыков создания примитивных систем мониторинга, разграничения доступа и шифрования средствами СУБД.

# Ход работы

## Задача №1 – мониторинг БД

Задачи:

* + Создать таблицу-лог для записи изменений в БД;
  + Создать триггеры для основных таблиц, которые будут фиксировать изменения (вставка, обновление, удаление) и записывать их в таблицу-лог;
  + Продемонстрировать работу системы логирования.

### Создание лог-таблицы

1. Создание таблицы

CREATE TABLE public.main\_log (

log\_item\_id SERIAL PRIMARY KEY,

table\_name VARCHAR(50) NOT NULL,

operation\_type VARCHAR(30) NOT NULL,

operation\_date TIMESTAMP,

user\_operator VARCHAR(30) NOT NULL,

changed\_data JSONB

);

### Создание функции и триггеров

1. Создание функции

CREATE OR REPLACE FUNCTION logging() RETURNS TRIGGER AS $$

BEGIN

INSERT INTO public.main\_log (table\_name, operation\_type, operation\_date, user\_operator, changed\_data)

VALUES (

TG\_TABLE\_NAME,

TG\_OP,

NOW(),

current\_user,

row\_to\_json(CASE WHEN TG\_OP = 'DELETE' THEN OLD ELSE NEW END)

);

RETURN CASE WHEN TG\_OP = 'DELETE' THEN OLD ELSE NEW END;

END;

$$ LANGUAGE plpgsql;

1. Создание триггеров

-- Триггер для таблицы "Заказы"

CREATE TRIGGER logging\_orders

AFTER INSERT OR UPDATE OR DELETE ON public.orders

FOR EACH ROW EXECUTE FUNCTION logging();

-- Триггер для таблицы "Клиенты"

CREATE TRIGGER logging\_clients

AFTER INSERT OR UPDATE OR DELETE ON public.clients

FOR EACH ROW EXECUTE FUNCTION logging();

-- Триггер для таблицы "Сотрудники"

CREATE TRIGGER logging\_employees

AFTER INSERT OR UPDATE OR DELETE ON public.employees

FOR EACH ROW EXECUTE FUNCTION logging();

-- Вставка данных в таблицу "Заказы"

INSERT INTO public.orders (creation\_date, completion\_date, status, total\_cost, client\_id, employee\_id)

VALUES ('2024-02-02', NULL, 'Новый', 2500.00, 1, 1);

-- Обновление данных в таблице "Клиенты"

UPDATE public.clients SET email = 'new\_email@mail.ru' WHERE client\_id = 1;

-- Удаление данных из таблицы "Сотрудники"

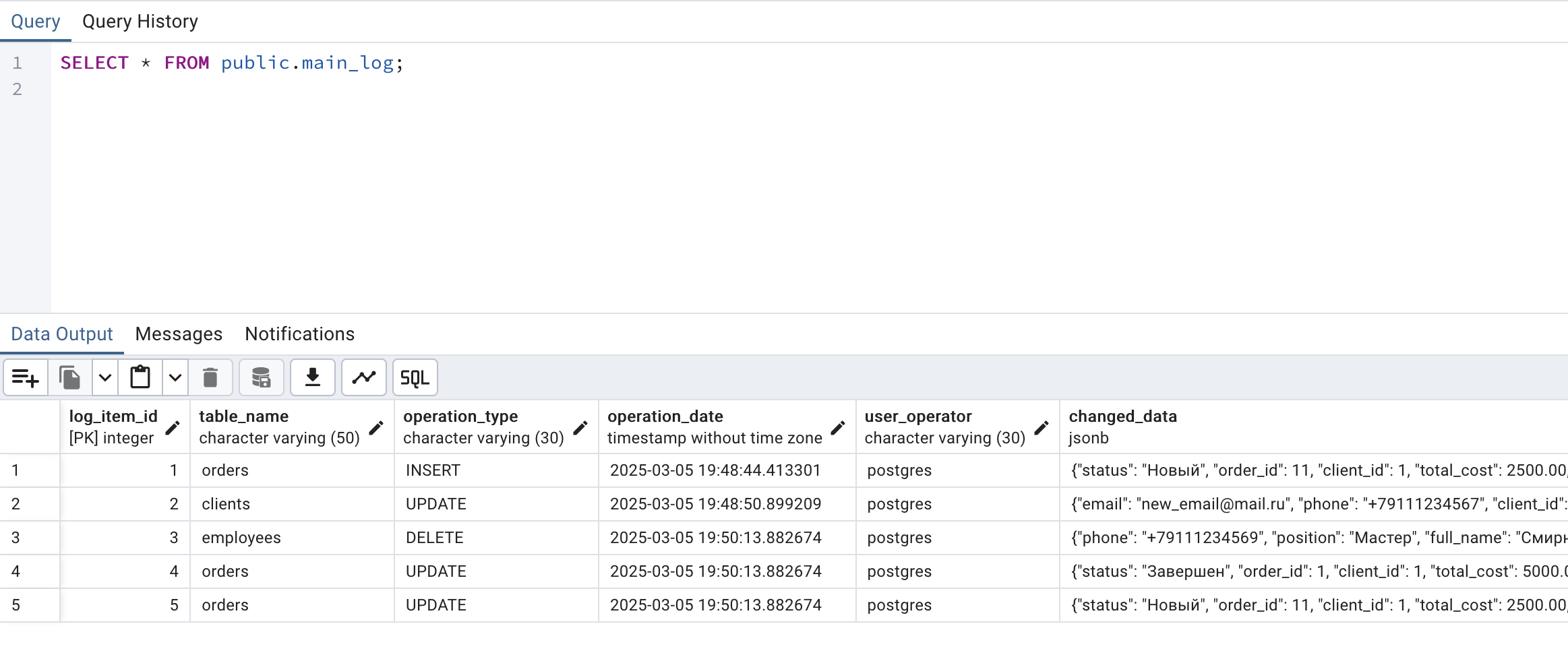
DELETE FROM public.employees WHERE employee\_id = 1;

-- Проверка таблицы-лога

SELECT \* FROM public.main\_log;

### Пример работы

Видим, что также обновились записи в таблице orders, где был удаленный employee\_id.



1. Пример заполнения логов после срабатывания триггеров

A screenshot of a computer

Description automatically generated

1. Обновление таблицы orders после удаление employee\_id

## Задача №2 – шифрование данных

Задачи по шифрованию данных:

* + Создать таблицу с секретными данными;
  + Зашифровать данные в таблице с использованием симметричного алгоритма шифрования (например, AES-256);
  + Продемонстрировать, что без знания ключа шифрования данные недоступны.

### Создание таблицы и шифрование данных

1. Создание таблицы и шифрование

CREATE TABLE public.secret\_data (

id SERIAL PRIMARY KEY,

username VARCHAR(30) NOT NULL,

secret\_token BYTEA NOT NULL

);

-- Установка расширения pgcrypto

CREATE EXTENSION IF NOT EXISTS pgcrypto;

-- Вставка зашифрованных данных

INSERT INTO public.secret\_data (username, secret\_token)

VALUES (

'operator\_1', pgp\_sym\_encrypt('token\_', '9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08')

);

### Пример работы

A screenshot of a computer

Description automatically generated

1. Проверка данных

A screenshot of a computer program

Description automatically generated

1. Попытка расшифровки данных без ключа

A screenshot of a computer

Description automatically generated

1. Успешная попытка расшифровки

## Задача №3 – разграничение доступа к БД

Задачи по разграничению доступа в БД:

* + Создать роли для различных классов пользователей (например, операторы, мастера);
  + Настроить привилегии для каждой роли в соответствии с принципом минимальных привилегий;
  + Продемонстрировать работу системы разграничения доступа.

### Создание ролей

1. Создание ролей

-- Роль для операторов

CREATE ROLE operator\_role WITH

NOLOGIN

NOSUPERUSER

NOCREATEDB

NOCREATEROLE

NOREPLICATION

INHERIT;

-- Роль для мастеров

CREATE ROLE master\_role WITH

NOLOGIN

NOSUPERUSER

NOCREATEDB

NOCREATEROLE

NOREPLICATION

INHERIT;

### Настройка привилегий

1. Настройка привилегий

-- Привилегии для операторов

GRANT SELECT, INSERT, UPDATE ON public.orders TO operator\_role;

GRANT SELECT ON public.clients TO operator\_role;

-- Привилегии для мастеров

GRANT SELECT, INSERT, UPDATE ON public.employees TO master\_role;

GRANT SELECT ON public.materials TO master\_role;

-- Пользователь для оператора

CREATE ROLE operator\_user WITH LOGIN PASSWORD 'operator\_pass';

GRANT operator\_role TO operator\_user;

-- Пользователь для мастера

CREATE ROLE master\_user WITH LOGIN PASSWORD 'master\_pass';

GRANT master\_role TO master\_user;

### Пример работы

A screenshot of a computer

Description automatically generated

1. Успешная попытка доступа

A screenshot of a computer

Description automatically generated

1. Неуспешная попытка доступа

Заключение

В ходе выполнения лабораторной работы были реализованы системы мониторинга, шифрования данных и разграничения доступа в базе данных. Эти механизмы обеспечивают безопасность данных и контроль доступа к информации в рамках СУБД.

Список источников

1. Документация PostgreSQL – Создание ролей [Электронный ресурс]. – URL: https://postgrespro.ru/docs/postgresql/9.6/sql-createrole (Дата обращения: 25.01.2025).