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

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

«Защита базы данных»

**Выполнил:**

Студент гр. N3347 Нгуен Тхе Вьет



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

(подпись)

**Проверил:**

Салихов Максим Русланович, аспирант ФБИТ 

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

(отметка о выполнении)

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

(подпись)

Содержание

[Содержание 2](#_Toc182298212)

[ВВЕДЕНИЕ 4](#_Toc182298213)

[ХОД РАБОТЫ 5](#_Toc182298214)

[1.1 Выбор СУБД 5](#_Toc182298215)

[1.2 Создание БД 5](#_Toc182298216)

[1.2.1 Код для создания всех таблиц; 7](#_Toc182298217)

[1.2.2 Код для внесения данных в созданные таблицы 7](#_Toc182298218)

[1.2.3 Код одной SQL-команды для модифицирования структуры таблицы 9](#_Toc182298219)

[1.3 Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных. 9](#_Toc182298220)

[1.4 Установить взаимосвязи между таблицами 9](#_Toc182298221)

[1.5 Создать представления, составленные в пункте 5 лабораторной 1. 9](#_Toc182298222)

[1.5.1 Представление для клиентов 9](#_Toc182298223)

[1.5.2 Представление для сотрудников 10](#_Toc182298224)

[1.5.3 Представление для менеджер 11](#_Toc182298225)

[1.6 Дополнительно. Тестовых запросов к вашей БД 12](#_Toc182298226)

[1.6.1 Получить все заказы с информацией о клиентах и сотрудниках. 12](#_Toc182298227)

[1.6.2 Перечислить продукты в конкретной категории. 13](#_Toc182298228)

[1.6.3 Найти заказы, размещенные конкретным клиентом. 13](#_Toc182298229)

[1.6.4 Найти the Most Popular Product 13](#_Toc182298230)

[1.6.5 Расчет общего дохода за определенный период 14](#_Toc182298231)

ВВЕДЕНИЕ

**Цель работы:**

Получение навыков созданию примитивных систем мониторинга, разграничения доступа и шифрования средствами СУБД

**Задание**

## Задачи по мониторингу БД:

* Создайте таблицу-лог, отдельную от ваших основных сущностей БД.
* Создайте для каждой основной таблицы в вашей БД триггер, который срабатывает при любых изменениях в БД (вставка новых данных, изменение существующих записей, удаление кортежей из таблицы). При срабатывании триггер должен вносить в таблицу-лог информацию о том, когда было произведено изменение, со стороны какой роли поступил запрос, какие кортежи поменялись, старые и новые значения.
* Продемонстрируйте работу системы логирования для различных операций и отношений.

## Задачи по шифрованию данных.

* Создайте таблицу с секретными данными, отдельно от ваших основных сущностей. Например, это может быть таблица с токенами или ключами доступа, для каждого класса-пользователей.
* Зашифруйте содержимое данной таблицы, в качестве алгоритма шифрования используйте любой симметричный алгоритм шифрования. Ключ шифрования для данной таблицы не должен храниться в ИС. Ключ шифрования может быть получен из индивидуального пароля для дешифрования суперпользователя (пароль не связан с паролем для входа в СУБД). Индивидуальный пароль суперпользователя и ключ шифрования может быть связан через одностороннюю функцию. Например, пусть индивидуальный пароль комбинация «!stroNgpsw31234», считаем от данного пароля детерминированную хэш-функцию (например, sha-256), полученный хэш-используем как ключ шифрования/дешифрования для симметричного алгоритма шифрования таблицы с секретными данными (например, для AES-256)
* Демонстрируем, что даже обладая полными правами администратора, но без знания индивидуального пароля невозможно получить содержимое таблицы с секретными данными

## Задачи по разграничению доступа в БД:

* Создайте в СУБД как минимум 2 роли (суперпользователь не считается) для каждого из классов потребителей информации;
* С помощью внутренних инструментов СУБД для каждой роли определите набор привилегий по отношению к таблицам вашей БД. Руководствуйтесь принципом минимальных привилегий, если определенному классу потребителей не нужен доступ к определенным таблицам/атрибутам (список задач БД, составленный в рамках 1 ЛР), то доступ к этим таблицам/атрибутам не предоставляется. Разграничиваем доступ к представлениям, созданным в 1 ЛР, а также таблицам логирования (таблицы логирования может просматривать только суперпользователь)
* Продемонстрируйте работу вашей системы разграничения доступа. Зайдите за каждую из ролей и покажите доступные со стороны каждой роли отношения.

ХОД РАБОТЫ

## Задачи по мониторингу БД:

Создадим таблицу-лог «main\_log», состоящую из следующих атрибутов:

CREATE TABLE public.main\_log (

log\_item\_id SERIAL PRIMARY KEY,

operation\_type TEXT,

operation\_date TEXT,

user\_operator TEXT,

changed\_data TEXT

);

Создадим триггерную функцию *logging()*, которая будет срабатывать при изменениях данных и сохранять тип операции, время, пользователя и выполненные изменения.

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

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO public.main\_log (operation\_type, operation\_date, user\_operator, changed\_data) VALUES ('D', now(), current\_user, row\_to\_json(OLD));

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO public.main\_log (operation\_type, operation\_date, user\_operator, changed\_data) VALUES ('U', now(), current\_user, row\_to\_json(NEW));

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO public.main\_log (operation\_type, operation\_date, user\_operator, changed\_data) VALUES ('I', now(), current\_user, row\_to\_json(NEW));

END IF;

RETURN NULL;

END;

$logging$ LANGUAGE plpgsql;

После того как я определил функцию logging(), необходимо создать триггер для каждой из основных таблиц в базе данных cafeteria\_db