1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Схема и базовое разграничение доступа**

по дисциплине «Системы управления базами данных»

1. Выполнила
2. студентка гр. 4851003/90801 Кулеева А.Г.

1. Руководитель
2. ассистент Зубков Е.А.
3. Санкт-Петербург
4. 2023

# **Цель**

Получение навыков создания схемы данных и базового разграничения доступа при работе с СУБД.

# Задание

1. Реализовать описанные отношения (рисунок 1), включая их заданные особенности и связи между ними.
2. Разработать матрицу доступа к данным, детализированную до атрибута на основе предложенных прав (другие права и типы пользователей не рассматривать). Права доступа к данным определить из набора CRUD (Create, Read, Update, Delete). Отразить ограничения на работу с атрибутами (ограничения горизонтального доступа к данным, если пользователь имеет доступ не ко всем кортежам). Обязательно выполнение правил доступа, определенных в задании. Остальные роли и не оговоренные атрибуты – на усмотрение студента.
3. Реализовать разграничение доступа при помощи встроенных средств СУБД (права, роли, политики безопасности), для всех ограничений, для которых это возможно.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 ― Вариант задания

# Ход работы

В соответствии с выданным вариантом была реализована логическая схема данных (рисунок 2).

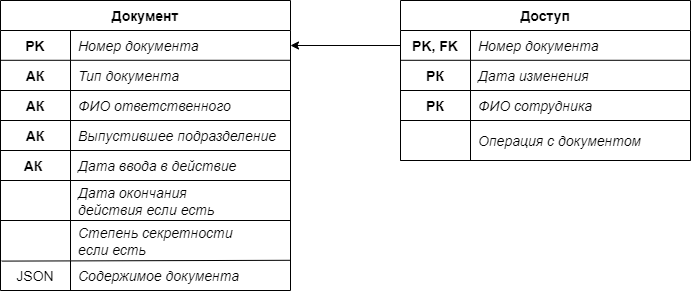


Рисунок 2 ― Реализованные связи

Создадим базу данных для выполнения работы (рисунок 3). Данная база будет состоять из двух таблиц (рисунки 4-5). Код реализации таблиц представлен в Приложении. Также стоит отметить, что для каждой таблицы была создана своя переменная перечислимого типа.

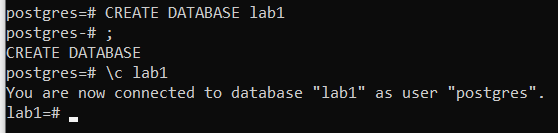


Рисунок 3 ― Требуемые привилегии

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 ― Схема для отношения 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 ― Схема для отношения 2

В соответствии с вариантом была создана матрица доступа (рисунок 6).



Рисунок 6 ― Ролевая матрица доступа

Перед созданием политик заполним таблицы данными. Всего было создано 3 документа, у каждого из них назначен ответственный. Во второй таблице были также прописаны данные: дата изменения, фамилия и операция (Приложение 1).

Было принято решение создать две роли: ответственный и не ответственный. Будем считать, что право на изменение документа есть только у ответственного сотрудника, при чем одним документом могут заниматься несколько человек. Тогда вне зависимости от секретности документа, у любого ответственного есть свой набор прав. А неответственный сотрудник может только видеть несекретные документы.

Сначала дадим ролям права из набора CRUD (Create, Read, Update, Delete). Это делается посредством команды GRANT. Для роли ответственного есть права на чтение и на запись для обеих таблиц. Для роли неответственного — только чтение для одной таблицы. Это вертикальное разграничение доступа, поскольку может быть применено к столбцам.

CREATE ROLE respons;

GRANT SELECT ON TABLE document TO respons;

GRANT INSERT ON TABLE document TO respons;

GRANT UPDATE ("date\_end") ON TABLE document TO respons; //отношение 1

GRANT SELECT ("operation") ON TABLE change\_log TO respons;

GRANT INSERT ("operation") ON TABLE change\_log TO respons;

GRANT UPDATE ("operation") ON TABLE change\_log TO respons; //отношение 2

CREATE ROLE not\_respons;

GRANT SELECT ON TABLE document TO not\_respons;

GRANT INSERT ON TABLE document TO not\_respons;

Теперь настроим горизонтальный доступ на уровне строк. Это делается с помощью политики. Итоговый доступ представляется как пересечение вертикальных и горизонтальных разрешений. Также стоит отметить, что безопасность в PostgreSQL работает по принципу «все, что не разрешено, то запрещено».

CREATE POLICY oper\_pol ON document

FOR ALL TO respons

USING (

(fio\_respons = current\_user));

CREATE POLICY resp\_pol ON change\_log

FOR ALL TO respons

USING (

(fio = current\_user));

CREATE POLICY not\_resp ON document

FOR ALL TO not\_respons

USING (

(secret = ‘1’));

Чтобы применить все созданные ограничения введем команду:

ALTER TABLE document ENABLE ROW LEVEL SECURITY;

На рисунке 7 представлен результат команды \d к обеим таблицам.

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 — Результат применения политик

Создадим пользователей и наделим их роллями (рисунок 8).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 — Создание пользователей

Проверим доступ для пользователей различных ролей (рисунки 9-10).

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 — Проверка доступа для ответственного пользователя

Изображение выглядит как текст, снимок экрана, экран, закрыть

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 — Проверка доступа для неответственного пользователя

# Вывод

В ходе выполнения данной лабораторной работы были получены навыки создания схемы данных и базового разграничения доступа при работе с СУБД. Были созданы две таблицы и пользователи с различными привилегиями. Разграничение доступа реализовано с помощью ролей и политик безопасности строк. После тестирования разработанной модели все условия задания оказались выполненными, то есть, итоговая матрица доступа не отличается от теоретической.

ПРИЛОЖЕНИЕ 1

CREATE DATABASE lab1;

\c lab1

CREATE TYPE sec\_lvl AS ENUM ('1', '2', '3');

CREATE TABLE document (

num\_doc BIGSERIAL NOT NULL PRIMARY KEY,

type\_doc VARCHAR(50) NOT NULL UNIQUE,

fio\_respons VARCHAR(100) NOT NULL UNIQUE,

department VARCHAR(50) NOT NULL UNIQUE,

date\_start DATE NOT NULL UNIQUE,

date\_end DATE,

secret sec\_lvl,

content json);

CREATE TYPE oper AS ENUM ('c', 'r', 'u', ‘d’);

CREATE TABLE change\_log (

num\_doc BIGSERIAL NOT NULL UNIQUE REFERENCES document (num\_doc),

date\_change DATE NOT NULL UNIQUE,

fio VARCHAR(100) NOT NULL UNIQUE,

operation oper,

PRIMARY KEY (num\_doc, date\_change, fio));

//////////////////////////////////////////////////////

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘docx’, ‘Ivanov’, ‘IT’, ’2022-01-01’, ’2025-01-01’, ‘2’);

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘pdf’, ‘Smirnov’, ‘HR’, ’2020-03-01’, ’2024-01-01’, ‘1’);

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘txt’, ‘Kuleeva’, ‘IC’, ’2022-11-11’, ’2022-02-02’, ‘2’);

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘xml’, ‘test’, ‘admin’, ’2022-12-12’, ’2022-03-03’, ‘2’);

-----------------------------

INSERT INTO change\_log (

date\_change, fio, operation)

VALUES (‘2022-05-01’, ‘Kuleeva’, ‘u’);

INSERT INTO change\_log (

date\_change, fio, operation)

VALUES (‘2021-05-01’, ‘Smirnov’, ‘r’);

INSERT INTO change\_log (

date\_change, fio, operation)

VALUES (‘2022-05-05’, ‘Ivanov’, ‘d’);

INSERT INTO change\_log (

date\_change, fio, operation)

VALUES (‘2022-06-06’, ‘test’, ‘d’);

///////////////////////////////////////////////////////

CREATE ROLE respons;

GRANT SELECT ON TABLE document TO respons;

GRANT INSERT ON TABLE document TO respons;

GRANT UPDATE ("date\_end") ON TABLE document TO respons; //отношение 1

GRANT SELECT ("operation") ON TABLE change\_log TO respons;

GRANT INSERT ("operation") ON TABLE change\_log TO respons;

GRANT UPDATE ("operation") ON TABLE change\_log TO respons; //отношение 2

CREATE POLICY oper\_pol ON document

FOR ALL TO respons

USING (

(fio\_respons = current\_user));

CREATE POLICY resp\_pol ON change\_log

FOR ALL TO respons

USING (

(fio = current\_user));

//////////////////////////////////////////////////

CREATE ROLE not\_respons;

GRANT SELECT ON TABLE document TO not\_respons;

GRANT INSERT ON TABLE document TO not\_respons;

CREATE POLICY not\_resp ON document

FOR ALL TO not\_respons

USING (

(secret = ‘1’));

//////////////////////////////////////////////////

CREATE USER Kuleeva PASSWORD ‘kuleeva’;

CREATE USER Ivanov PASSWORD ‘ivanov’;

GRANT respons TO Kuleeva, Ivanov;