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

**ЛАБОРАТОРНАЯ РАБОТА № 1  
   
«Базовое разграничение доступа при работе с СУБД**»

1. по дисциплине «Системы управления базами данных»
2. Выполнил
3. студент гр. 4851004/80101 Терещенко Е. А.

<*подпись*>

1. Проверил
2. Ассистент преподавателя Полтавцева М. А.

<*подпись*>

1. Санкт-Петербург
2. 2021

Цель работы

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

Задание

Варианты выполнения лабораторной работы, включая описание схемы данных и требования по разграничению прав доступа представлены ниже.

1. Реализовать описанные отношения, 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).

3. Реализовать разграничение доступа при помощи встроенных средств СУБД (права, роли, политики безопасности), для всех ограничений, для которых это возможно. Если реализация какого-либо права (особенности права доступа) встроенными средствами невозможна, дать пользователю более широкие права и отразить это в отчете.

Вариант задания

В рамках работы был выполнен 7 вариант задания.

Реализованы отношения:

* **Отношение 1**

Код проекта (PK), Название проекта (AK1), Наименование задачи (PK, AK1), ФИО исполнителя (FK), Трудоемкость в часах, Плановая дата выполнения, Реальная дата выполнения если есть, Описание задачи, Отметка о принятии задачи руководителем.

* **Отношение 2**

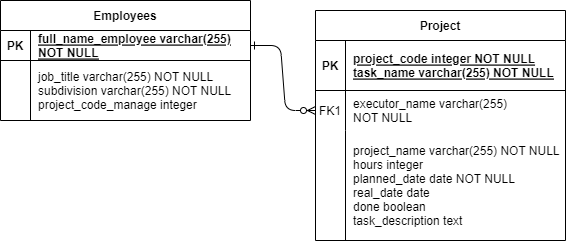
ФИО сотрудника (PK), Должность, Подразделение, Код проекта которым руководит сотрудник, если он есть.

**Описание прав доступа**:

Сотрудник видит все задачи и названия проектов, но без кода проекта и отметки руководителя, на которых он работает. Сотрудник может изменять реальную дату выполнения задачи если он является ее исполнителем. Если сотрудник руководит проектом, он видит все данные по этому проекту в отношении всех задач и всех исполнителей и может корректировать отметку о принятии задачи руководителем.

Часть 1. Реализованные отношения

Логическая (реляционная) схема представлена на рисунке 1.1. Согласно варианту задания, было реализовано два отношения – отношение Employees (Рис. 1.2.), описывающее сотрудников компании, и отношение Projects (Рис. 1.3), описывающее проекты.

  
Рисунок 1.1. – Логическая схема

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.2. – Создание отношения Employees

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 1.3. – Создание отношения Projects

Часть 2. Матрица доступа

Было принято решение разделить сотрудников на два типа – исполнители проекта и руководители.

Разработанные матрицы доступа представлены в таблицах 2.1, 2.2. К таблице Employees и сотрудники-исполнители, и сотрудники-руководители имеют одинаковы дотуп.

Таблица 2.1. Матрица доступа к таблице Projects

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Код проекта | Название проекта | Наименование задачи | ФИО исполнителя | Трудоемкость в часах | Плановая дата выполнения | Реальная дата выполнения | Описание задачи | Отметка о принятии задачи руководителем |
| Исполнитель | CREATE | - | - | - | - | - | - | - | - | - |
| READ | - | + | + | + | + | + | + | + | - |
| UPDATE | - | - | - | - | - | - | + | - | - |
| DELETE | - | - | - | - | - | - | - | - | - |
| Руководитель | CREATE | - | - | - | - | - | - | - | - | - |
| READ | + | + | + | + | + | + | + | + | + |
| UPDATE | - | - | - | - | - | - | - | - | + |
| DELETE | - | - | - | - | - | - | - | - | - |

Таблица 2.2. Матрица доступа к таблице Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ФИО сотрудника | Должность | Подразделение | Код проекта |
| CREATE | - | - | - | - |
| READ | + | + | + | + |
| UPDATE | - | - | - | - |
| DELETE | - | - | - | - |

С помощью встроенных средств удалось реализовать все права доступа согласно матрицам, приведенным выше.

Работа выполнялась в СУБД PostgreSQL. Данная СУБД обладает функциями защиты на уровне столбцов (с помощью команды GRANT) и отдельно на уровне строк (CREATE POLICY). Примеры приведены на рисунках 2.1, 2.2.

Примечательно, что для реализации политик защиты на уровне строк пришлось добавить столбец “login” (очевидно из названия, хранит в себе логин пользователя) в таблицу Employees.

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.1. Защита на уровне столбцов

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 2.1. Защита на уровне строк

Часть 3. Тестирование

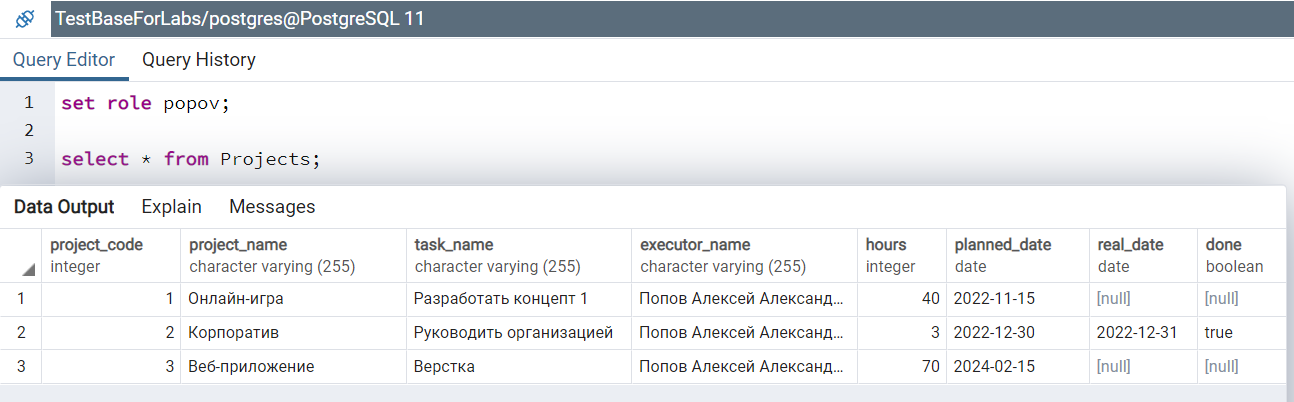
Для того, чтобы протестировать разработанную политику доступа, создадим пользователей, принадлежащих двум разным «группам» (Рис. 3.1).

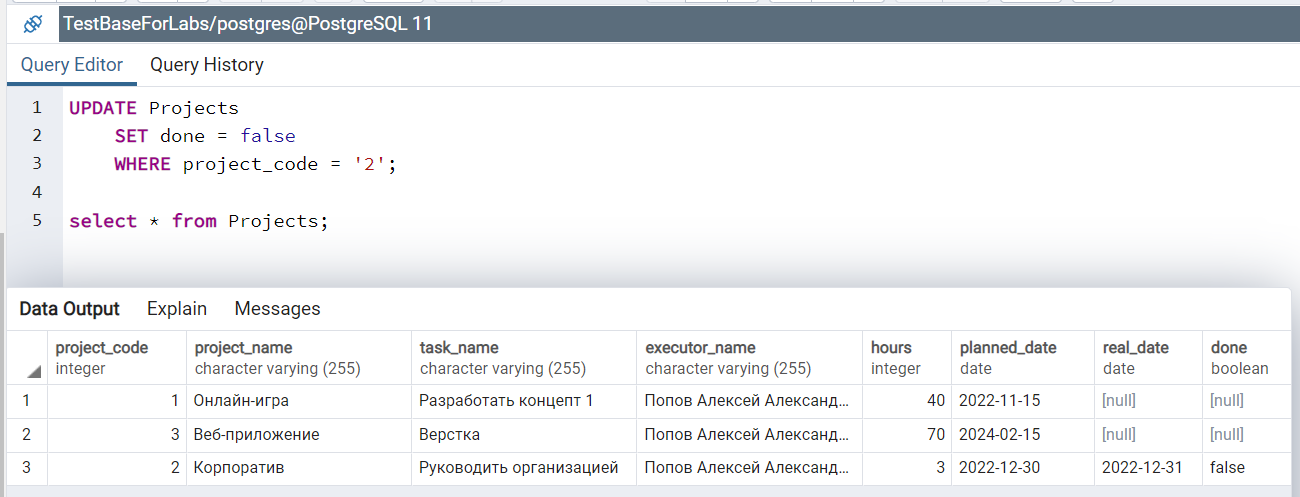
Изображение выглядит как текст, табличка, снимок экрана

Автоматически созданное описание  
Рисунок 3.1. – Создание ролей для тестирования

Напомним, что руководители проекта могут просматривать всю информацию о проекте, а также изменять отметку о принятии задачи в рамках проекта, которым руководят.

За роль Попова попробуем просмотреть таблицу Projects (рисунок 3.2) и изменить отметку о принятии (рисунок 3.3).

  
Рисунок 3.2 – Демонстрация права на выполнение SELECT

  
Рисунок 3.3 – Демонстрация права на выполнение UPDATE

Теперь авторизируемся за сотрудника-исполнителя и попробуем прочитать всю таблицу (рисунок 3.4), только разрешенные поля (рисунок 3.5) и изменить отметку о принятии (рисунок 3.6).

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 3.4 – Попытка чтения всей таблицы Projects

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 3.5 – Попытка чтения разрешенных полей таблицы Projects

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 3.6 – Попытка изменения поля о принятии задачи

Вывод

В ходе работы были получены навыки создания схемы данных и базового разграничения доступа при работе с СУБД.

СУБД PostgreSQL имеет функции для защиты таблиц и на уровне строк, и на уровне столбцов, что позволяет достаточно гибко реализовать разграничение доступа. Помимо этого, удобным является также механизм ролей – нет необходимости разделять роли на группы и пользователей.

Приложение 1. Реализованные запросы

--------------------------------------------------------------------------------------СОЗДАНИЕ ОТНОШЕНИЙ

create table Employees

(

    full\_name\_employee varchar(255) not null, --фио сотрудника

    job\_title varchar(255) not null, --должность

    subdivision varchar(255) not null, --подразделение

    project\_code\_manage integer, --код проекта, которым руководит сотрудник

    primary key (full\_name\_employee)

);

create table Projects

(

    project\_code integer not null, --код проекта

    project\_name varchar(255) not null, --название проекта

    task\_name varchar(255) not null, --наименование задачи

    executor\_name varchar(255) not null, --фио исполнителя

    hours integer, --трудоемкость в часах

    planned\_date date not null, --плановая дата выполнения

    real\_date date, --реальная дата

    done boolean, --отметка о принятии

    task\_description text, --описание задачи

    primary key (project\_code, task\_name),

    constraint alternative\_key unique (project\_name, task\_name),

    foreign key (executor\_name) references Employees (full\_name\_employee)

);

--------------------------------------------------------------------------------------ЗАПОЛНЕНИЕ ТЕСТОВЫМИ ДАННЫМИ

insert into employees values

('Орлова Екатерина Ивановна', 'Ассистент', 'Отдел HR', NULL),

('Гейнц Лилия Викторовна', 'Менеджер', 'Отдел HR', NULL),

('Федоров Александр Викторович', 'Тестировщик', 'Отдел тестирования', NULL),

('Петров Георгий Афанасьевич', 'Тестировщик', 'Отдел тестирования', NULL),

('Лебедева Мария Ильина', 'Тестировщик', 'Отдел тестирования', NULL),

('Минаев Михаил Богданович', 'Тестировщик', 'Отдел тестирования', NULL),

('Любимова Ольга Владимировна', 'Руководитель отдела тестирования', 'Отдел тестирования', '3'),

('Иванова Ульяна Леонидовна', 'Программист', 'Отдел разработки', NULL),

('Карпов Петр Дмитриевич', 'Программист', 'Отдел разработки', NULL),

('Карпова Анастасия Дмитриевна', 'Руководитель отдела разработки', 'Отдел разработки', '1'),

('Попов Алексей Александрович', 'Дизайнер', 'Отдел дизайна', '2');

insert into projects values

('1', 'Онлайн-игра', 'Разработать концепт 1', 'Попов Алексей Александрович', '40', '15-11-22', NULL, NULL, 'Task description'),

('1', 'Онлайн-игра', 'Реализовать модуль 1', 'Иванова Ульяна Леонидовна', '60', '29-11-22', NULL, NULL, 'Task description'),

('1', 'Онлайн-игра', 'Реализовать модуль 2', 'Иванова Ульяна Леонидовна', '80', '17-10-22', NULL, NULL, 'Task description'),

('1', 'Онлайн-игра', 'Реализовать модуль 3', 'Карпов Петр Дмитриевич', '100', '18-10-22', NULL, NULL, 'Task description'),

('1', 'Онлайн-игра', 'Протестировать', 'Минаев Михаил Богданович', '25', '15-09-22', '15-09-22', true, 'Task description'),

('1', 'Онлайн-игра', 'Руководить проектом', 'Карпова Анастасия Дмитриевна', '94', '01-01-23', NULL, NULL, 'Task description'),

('2', 'Корпоратив', 'Заказать еду', 'Лебедева Мария Ильина', '3', '29-12-22', '29-12-22', NULL, 'Task description'),

('2', 'Корпоратив', 'Организовать помещение', 'Федоров Александр Викторович', '3', '30-12-22', NULL, NULL, 'Task description'),

('2', 'Корпоратив', 'Руководить организацией', 'Попов Алексей Александрович', '3', '30-12-22', '31-12-22', true, 'Task description'),

('3', 'Веб-приложение', 'Верстка', 'Федоров Александр Викторович', '140', '03-02-24', NULL, NULL, 'Task description'),

('3', 'Веб-приложение', 'Верстка', 'Попов Алексей Александрович', '70', '15-02-24', NULL, NULL, 'Task description'),

('3', 'Веб-приложение', 'Тестирование, руководство', 'Любимова Ольга Владимировна', '94', '10-05-24', NULL, NULL, 'Task description'),

('3', 'Веб-приложение', 'Бэкенд', 'Иванова Ульяна Леонидовна', '80', '15-05-24', NULL, NULL, 'Task description'),

('3', 'Веб-приложение', 'Бэкенд', 'Карпова Анастасия Дмитриевна', '110', '20-12-23', NULL, NULL, 'Task description');

--------------------------------------------------------------------------------------ПОЛИТИКИ

--создание ролей

CREATE ROLE executor\_group NOINHERIT;

CREATE ROLE teamlead\_group NOINHERIT;

CREATE ROLE lyubimova;

CREATE ROLE popov;

CREATE ROLE orlova;

CREATE ROLE lebedeva;

CREATE ROLE petrov;

GRANT teamlead\_group TO lyubimova;

GRANT teamlead\_group TO  popov;

GRANT executor\_group TO orlova;

GRANT executor\_group TO lebedeva;

GRANT executor\_group TO petrov;

--просмотр ролей

SELECT rolname FROM pg\_roles;

--защита на уровне столбцов

GRANT SELECT (project\_name, task\_name, executor\_name, hours, planned\_date, real\_date, task\_description)

    ON Projects

    TO executor\_group;

GRANT SELECT

    ON Projects

    TO teamlead\_group;

GRANT SELECT

    ON Employees

    TO executor\_group;

 GRANT SELECT

    ON Employees

    TO teamlead\_group;

GRANT UPDATE (planned\_date)

    ON Projects

    TO executor\_group;

GRANT UPDATE (done)

    ON Projects

    TO teamlead\_group;

--защита на уровне строк

alter table projects enable row level security;

CREATE POLICY policy\_select\_for\_executor\_group ON Projects

    FOR SELECT

    TO executor\_group

    USING (executor\_name IN (SELECT full\_name\_employee FROM Employees WHERE login = CURRENT\_ROLE));

CREATE POLICY policy\_update\_for\_executor\_group ON Projects

    FOR UPDATE

    TO executor\_group

    USING (executor\_name IN (SELECT full\_name\_employee FROM Employees WHERE login = CURRENT\_ROLE));

CREATE POLICY policy\_select\_for\_teamlead\_group ON Projects

    FOR SELECT

    TO teamlead\_group

    USING (executor\_name IN (SELECT full\_name\_employee FROM Employees WHERE login = CURRENT\_ROLE));

CREATE POLICY policy\_update\_for\_teamlead\_group ON Projects

    FOR UPDATE

    TO teamlead\_group

    USING (executor\_name IN (SELECT full\_name\_employee FROM Employees WHERE login = CURRENT\_ROLE));

--исправляем беду :с

ALTER TABLE Employees ADD COLUMN login varchar(255) not null DEFAULT 'xxx';

DROP POLICY policy\_select\_for\_executor ON Projects;

DROP POLICY policy\_update\_for\_executor\_group ON Projects;

DROP POLICY policy\_select\_for\_teamlead\_group ON Projects;

DROP POLICY policy\_update\_for\_teamlead\_group ON Projects;

UPDATE Employees

    SET login = 'orlova'

    WHERE full\_name\_employee = 'Орлова Екатерина Ивановна';

UPDATE Employees

    SET login = 'lyubimova'

    WHERE full\_name\_employee = 'Любимова Ольга Владимировна';

UPDATE Employees

    SET login = 'popov'

    WHERE full\_name\_employee = 'Попов Алексей Александрович';

UPDATE Employees

    SET login = 'lebedeva'

    WHERE full\_name\_employee = 'Лебедева Мария Ильина';

UPDATE Employees

    SET login = 'petrov'

    WHERE full\_name\_employee = 'Петров Георгий Афанасьевич';