|  | | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | | |  | |
|  | «Пермский государственный национальный  исследовательский университет» | | | | | | | |
|  | | | |  | | |  | |
|  | | | ОТЧЕТ  О ВЫПОЛНЕНИИ ЗАДАНИЯ №8  ПО ДИСЦИПЛИНЕ БД И СУБД  по теме:  «Права доступа» | | | | |  |
|  | | | |  | | |  | |
|  | |  | | |  | Работу  Выполнил  Студент гр.ПМИ-4-20,  3 курс  Пуховкин В.И. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_\_»\_\_\_\_\_\_\_2023г | | |
|  | | | | Пермь 2023 | | |  | |

1. **Задание**

Для спроектированной ранее базы данных предметной области предложить категории пользователей и их права (как минимум на чтение таблиц/строк/столбцов, на запись в различные таблицы), настроить их на уровне СУБД. Продемонстрировать различие в возможностях созданных пользователей и владельца базы данных.

1. **Теория и реализация**

Для выполнения данной лабораторной я выбрал предметную область - университет

**Схема базы данных:**

Таблица студенты

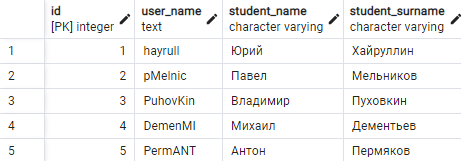


Таблица преподаватели

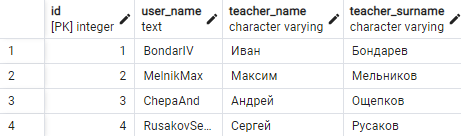
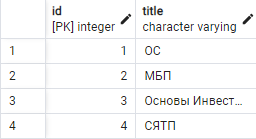


Таблица предметы



Таблица, связующая преподавателя и предмет

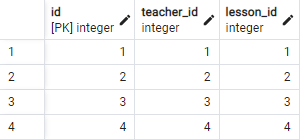
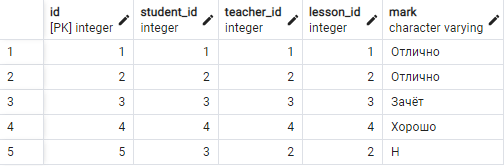


Таблица оценки



**Матрица прав доступа:**



**Описание реализации:**

1) Создаём представления для более удобного вывода данных

2) Создаём роли для реализации прав доступа на разном уровне

3) Создаём описание прав доступа для каждой роли

4) Создаём права доступа на уровне строк

5) Переходим к работе кода

6) Удаляем все данные

**Теория:**

1) Создание и удаление ролей

CREATE ROLE роль [WITH] атрибут [атрибут ...]

В нашем скрипте используем атрибуты:

SUPERUSER суперпользователь

И NOINHERIT – Не позволяет наследовать права той роли, членом которого она является.

(Назначаем сами непосредственно пользователю)

DROP ROLE IF EXISTS <имя роли>

2) Создание и удаление пользователя

CREATE USER <имя пользователя>

DROP USER IF EXISTS <имя пользователя>

3) Создание и удаление прав доступа

Для назначения прав применяется команда GRANT. Например, если в базе данных есть роль Student и таблица show\_marks\_for\_student, право на просмотр таблицы можно дать этой роли так:

GRANT SELECT ON TABLE show\_marks\_for\_student TO Student;

Чтобы дать пользователю (PuhovKin) роль (Student):

GRANT Student to "PuhovKin";

4) Политика защиты строк

В нашем скрипте для защиты строк таблицы, используется:

ALTER TABLE (имя\_таблицы) ENABLE ROW LEVEL SECURITY

Когда для таблицы включается защита строк, все обычные запросы к таблице на выборку или модификацию строк должны разрешаться политикой защиты строк.

Для создания политик предназначена команда CREATE POLICY, для изменения — ALTER POLICY, а для удаления — DROP POLICY. Чтобы включить или отключить защиту строк для определённой таблицы, воспользуйтесь командой ALTER TABLE.

1. **Тестирование**

До начала тестирования уже создана БД, подключены все права доступа и ограничения на строки и созданы следующие роли:

CREATE ROLE Decan SUPERUSER;

CREATE ROLE Student NOINHERIT;

CREATE ROLE Teacher NOINHERIT;

**1) Работа с БД в роли студента**

--Студент Владимир

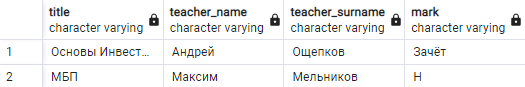
create user "PuhovKin";

GRANT Student to "PuhovKin";

set role "PuhovKin";

--Выводим оценки для пользователя PuhovKin

SELECT \* from show\_marks\_for\_student;

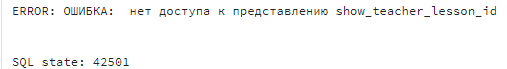


--Пытаемся вывести данные недоступных нам таблиц (Получим Ошибку)

select \* from show\_teacher\_lesson\_id;

select \* from show\_student\_id;

select \* from show\_id\_for\_teacher;



--Вывести данные не получится, так как мы не имеем доступ

reset role;

**2) Работа с БД в роли преподавателя**

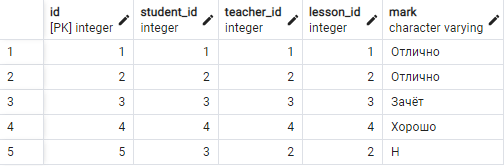
create user "RusakovSerge";

GRANT Teacher to "RusakovSerge";

set role "RusakovSerge";

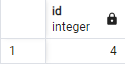
--Просмотрим таблицу оценок

select \* from marks;



--Выведем свой id

select \* from show\_id\_for\_teacher;



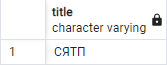
--Выведем id своего предмета

select \* from show\_teacher\_lesson\_id;



--Выведем название своего предмета

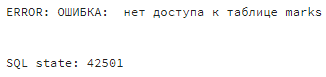
select \* from show\_name\_lesson;



--Попробуем изменить столбцы, разрешение на которых нет в marks

update marks set id = 2 ;

update marks set teacher\_id = 2 ;

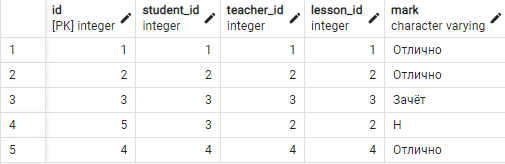


update marks set student\_id = 4 ;

update marks set lesson\_id = 4;

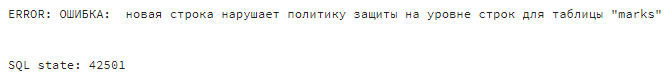
update marks set mark = 'Отлично';

select \* from marks;



--Вставляем значения, которые мы не можем вставлять (id не нашего учителя)

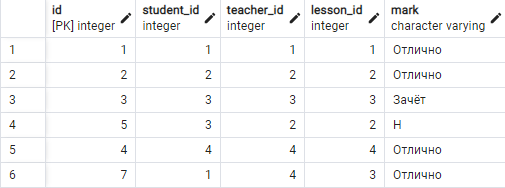
insert into marks (student\_id, teacher\_id, lesson\_id, mark) VAlUES (1,1,1,'Зачёт');



--Вставляем значения, но используя уже id текущего преподавателя

insert into marks (student\_id, teacher\_id, lesson\_id, mark) VALUES (1,4,3,'Отлично');

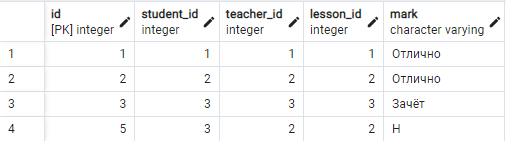
select \* from marks;



--Удаляем значения в таблице (Удалятся только данные связанные с нашем учителем)

delete from marks;

select \* from marks;



reset role ;

**3) Работа с БД в роли декана**

insert into teacher(teacher\_name, teacher\_surname,user\_name)

values ('Андрей','Геннадьевич','IasamiyGlavniy');

insert into lesson (title)

values ('Введение в Специальность');

insert into teacher\_lesson(teacher\_id, lesson\_id)

values (5,5);

insert into marks(student\_id, teacher\_id, lesson\_id, mark)

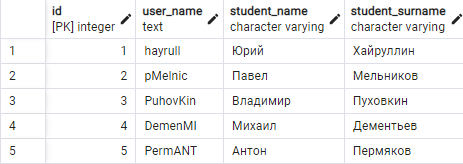
VALUES (3,5,5,'Отлично') ;

create user "IasamiyGlavniy";

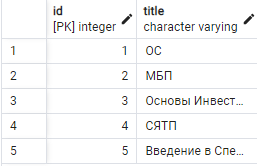
GRANT Decan to "IasamiyGlavniy";

set role Decan;

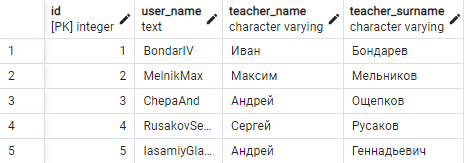
select \* from student;



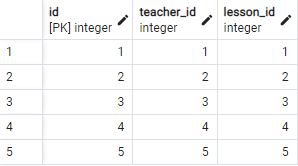
select \* from lesson;



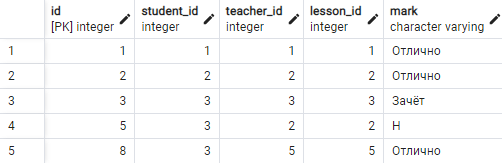
select \* from teacher;



select \* from teacher\_lesson;



select \* from marks;



--Обновление всех срок сразу

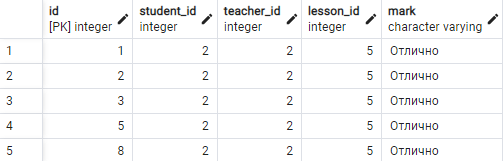
update marks set student\_id = 2 ;

update marks set teacher\_id = 2 ;

update marks set lesson\_id = 5 ;

update marks set mark = 'Отлично';

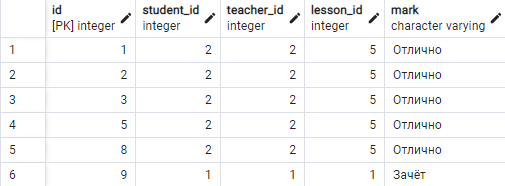
select \* from marks;



--Вставка значений любого преподавателя

insert into marks(student\_id, teacher\_id, lesson\_id, mark)

values (1,1,1,'Зачёт');



--Удаление таблиц и строк

DELETE FROM marks;

DELETE FROM student;

DELETE FROM teacher\_lesson;

DELETE FROM teacher;

DELETE FROM lesson;



reset role ;

--DROP USERS

DROP USER IF EXISTS "PuhovKin";

DROP USER IF EXISTS "RusakovSerge";

DROP USER IF EXISTS "IasamiyGlavniy";

--REVOKE PRIVILEGES

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM Decan;

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM Teacher;

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM Student;

--DROP POLICIES

DROP POLICY IF EXISTS teacher\_del\_grades ON marks;

DROP POLICY IF EXISTS teacher\_upd\_marks ON marks;

DROP POLICY IF EXISTS teacher\_ins\_grades ON marks;

drop table student cascade ;

drop table lesson cascade ;

drop table teacher cascade ;

drop table teacher\_lesson cascade ;

drop table marks cascade ;

drop view show\_teacher\_lesson\_id;

drop view show\_id\_for\_teacher;

drop view show\_marks\_for\_student;

drop view show\_student\_id;

--DROP ROLES

DROP ROLE IF EXISTS Decan;

DROP ROLE IF EXISTS Teacher;

DROP ROLE IF EXISTS Student;