create table student

(

id int GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

user\_name text not null unique ,

student\_name varchar not null,

student\_surname varchar not null

);

ALTER TABLE student ENABLE ROW LEVEL SECURITY;

create table teacher

(

id int generated by default as identity PRIMARY KEY ,

user\_name text not null unique ,

teacher\_name varchar not null ,

teacher\_surname varchar not null

);

ALTER TABLE teacher ENABLE ROW LEVEL SECURITY;

create table lesson

(

id int generated by default as identity PRIMARY KEY ,

title varchar not null

);

ALTER TABLE lesson ENABLE ROW LEVEL SECURITY;

create table marks

(

id int generated by default as identity PRIMARY KEY ,

student\_id int references student(id),

teacher\_id int references teacher (id),

lesson\_id int references lesson(id),

mark varchar not null

);

ALTER TABLE marks ENABLE ROW LEVEL SECURITY;

create table teacher\_lesson

(

id int generated by default as identity PRIMARY KEY,

teacher\_id int references teacher (id),

lesson\_id int references lesson(id)

);

ALTER TABLE teacher\_lesson enable ROW LEVEL SECURITY;

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

values ('Юрий', 'Хайруллин', 'hayrull'),

('Павел','Мельников','pMelnic'),

('Владимир','Пуховкин','PuhovKin'),

('Михаил','Дементьев','DemenMI'),

('Антон','Пермяков','PermANT');

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

values ('Иван','Бондарев', 'BondarIV'),

('Максим','Мельников','MelnikMax'),

('Андрей','Ощепков','ChepaAnd'),

('Сергей','Русаков','RusakovSerge');

insert into lesson (title)

values ('ОС'),('МБП'),('Основы Инвестиций'),('СЯТП');

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

values (1,1),(2,2),(3,3),(4,4);

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

values (1,1,1,'Отлично'),(2,2,2,'Отлично'),(3,3,3,'Зачёт'),(4,4,4,'Хорошо'),(3,2,2,'Н');

--Просмотр оценок для студента

CREATE OR REPLACE VIEW show\_marks\_for\_student AS

SELECT lesson.title, teacher.teacher\_name, teacher.teacher\_surname , mark

FROM marks

JOIN student ON marks.student\_id = student.id

JOIN lesson ON lesson.id = marks.lesson\_id

JOIN teacher ON teacher.id = marks.teacher\_id

WHERE student.user\_name = user;

--Просмотр id преподавателя

create or replace view show\_id\_for\_teacher as

select id from teacher t

where t.user\_name = user;

--Просмотр id студента

CREATE OR REPLACE VIEW show\_student\_id AS

SELECT id

FROM student;

--Просмотр айдишников предметов текущего преподавателя

CREATE OR REPLACE VIEW show\_teacher\_lesson\_id AS

SELECT lesson\_id

FROM teacher\_lesson

JOIN teacher ON teacher\_lesson.teacher\_id = teacher.id

WHERE user\_name = user;

--Просмотр названия предмета

create or replace view show\_name\_lesson as

select lesson.title from lesson

left outer join teacher\_lesson on lesson.id = teacher\_lesson.lesson\_id

join teacher t on teacher\_lesson.teacher\_id = t.id

where user\_name = user;

--Создание ролей

CREATE ROLE Decan SUPERUSER;

CREATE ROLE Student NOINHERIT;

CREATE ROLE Teacher NOINHERIT;

--Права доступа для Студента (Только просмотр оценок)

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

--Права доступа для Преподавателя (Просмотр, изменение оценок, студентов и предмета )

GRANT SELECT,INSERT,UPDATE(student\_id,lesson\_id,mark), DELETE ON TABLE marks TO Teacher;

GRANT SELECT ON TABLE show\_id\_for\_teacher TO Teacher;

GRANT SELECT ON TABLE show\_student\_id to Teacher;

GRANT SELECT ON TABLE show\_teacher\_lesson\_id TO Teacher;

GRANT SELECT ON TABLE show\_name\_lesson TO Teacher;

GRANT USAGE, SELECT ON SEQUENCE marks\_id\_seq TO Teacher;

--Учитель может просматривать все оценки

CREATE POLICY teacher\_see\_marks

ON marks

FOR SELECT TO Teacher

USING (TRUE);

--Определённый учитель может менять только свои строки

CREATE POLICY teacher\_upd\_marks

ON marks

FOR UPDATE TO Teacher

USING (teacher\_id = (SELECT \*

FROM show\_id\_for\_teacher));

CREATE POLICY teacher\_ins\_grades

ON marks

FOR INSERT TO Teacher

WITH CHECK (

teacher\_id = (SELECT \*

FROM show\_id\_for\_teacher)

);

CREATE POLICY teacher\_del\_grades

ON marks

FOR DELETE TO Teacher

USING (teacher\_id = (SELECT \*

FROM show\_id\_for\_teacher));

--1) Работа от лица 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 от лица учителя:

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 ;

--Изменяем значения в разрешённые строки

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 от лица 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');

select \* from marks;

--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;