create table classes(

id integer primary key,

id\_classroom\_teacher integer Unique,

id\_type integer Unique,

students\_quantity integer,

letter char,

description json,

Constraint classe\_fk Foreign Key(id\_type) References classes\_types (id))

create table classes\_types(

id integer primary key,

name varchar(50))

create table students(

full\_name varchar(150) primary key,

birthday date,

gender varchar(20),

address varchar(100),

father\_full\_name varchar(150),

mather\_full\_name varchar(150),

id\_class integer,

add\_inform varchar(500),

Constraint students\_fk Foreign Key(id\_class) References classes (id))

create table employees(

id integer primary key,

full\_name varchar(150),

age integer,

gender varchar(20),

phone varchar(20),

post varchar(150),

Constraint employees\_classes\_fk Foreign Key(id) References classes (id\_classroom\_teacher),

Constraint employees\_subject\_fk Foreign Key(id) References subject (id))

create table subject(

id integer primary key,

name varchar(100),

description varchar(500),

id\_employee integer)

create table schedule(

ddate date primary key,

week\_day varchar(50),

id\_class integer,

id\_subject integer,

start\_time time,

end\_time time,

Constraint schedule\_fk Foreign Key(id\_subject) References subject (id))

UPDATE classes SET description = '{"Обучение":7,"Создание":2016}' WHERE id=1;

UPDATE classes SET description = '{"Обучение":5,"Создание":2018}' WHERE id=2;

UPDATE classes SET description = '{"Обучение":10,"Создание":2013}' WHERE id=3;

UPDATE classes SET description = '{"Обучение":2,"Создание":2021}' WHERE id=4;

UPDATE classes SET description = '{"Обучение":8,"Создание":2015}' WHERE id=5;

**Представления:**

CREATE VIEW subject\_list AS SELECT name as subjects FROM subject s, employees e

WHERE s.id\_employee=e.id and e.full\_name='Воронин Максим Артемьевич'

CREATE VIEW schedule\_7B AS select schedule.ddate, schedule.week\_day, classes.id\_type, classes.letter, subject.name, schedule.start\_time, schedule.end\_time

from schedule, classes, subject

where schedule.id\_class=classes.id and schedule.id\_subject=subject.id and id\_class =(select id from classes where letter='Б' and id\_type=7) and week\_day='Вторник'

CREATE VIEW schedule\_Voronin AS select schedule.ddate, schedule.week\_day, classes.id\_type, classes.letter, subject.name, schedule.start\_time, schedule.end\_time

from schedule, classes, subject

where schedule.id\_class=classes.id and schedule.id\_subject=subject.id and id\_subject in

(SELECT s.id FROM subject s, employees e WHERE s.id\_employee=e.id and e.full\_name='Воронин Максим Артемьевич')and ddate>='2023-12-11' and ddate<='2023-12-16'

CREATE VIEW inform\_7B AS select full\_name,to\_char(birthday, 'dd.mm.yyyy'), gender, address, father\_full\_name, mather\_full\_name, classes.id\_type, classes.letter, add\_inform

from students,classes

where students.id\_class=classes.id and id\_class=(select id from classes where letter='Б' and id\_type=7)

CREATE VIEW qty\_subjects\_7B AS select count(\*) as qty\_7Б from schedule

where id\_class =(select id from classes where letter='Б' and id\_type=7)

CREATE VIEW qty\_in\_classes AS SELECT classes.id\_type, classes.letter, COUNT(\*) FROM students, classes

where students.id\_class=classes.id

group by id\_type,letter

**Функции:**

CREATE FUNCTION teacher (subject\_name varchar(100))

RETURNS SETOF text AS $$

SELECT full\_name FROM employees WHERE id = (select id\_employee from subject where name=subject\_name);

$$ LANGUAGE SQL;

select \* from teacher('ОБЖ')

CREATE FUNCTION teacher\_schedule (teacher varchar(100), dat date)

RETURNS SETOF text AS $$

SELECT subject.name FROM subject, schedule, employees

WHERE subject.id\_employee=employees.id and schedule.id\_subject=subject.id and employees.full\_name=teacher and schedule.ddate=dat

$$ LANGUAGE SQL;

select \* from teacher\_schedule('Нечаева Варвара Марковна','2023-12-12')

CREATE FUNCTION lesson\_qty (lesson varchar(100))

RETURNS integer AS $$

select count(\*) from (select subject.name from subject, schedule

where subject.id=schedule.id\_subject and date\_part('week',schedule.ddate)=date\_part('week',now()) and name=lesson) as lessons

$$ LANGUAGE SQL;

select \* from lesson\_qty('Физика')

CREATE FUNCTION students\_age (down integer, up integer)

RETURNS integer AS $$

select count(\*) from (select date\_part('year',now())-date\_part('year',birthday) as age from students) as count

where age>=down and age<=up

$$ LANGUAGE SQL;

select \* from students\_age(10,15)

**Хранимые процедуры:**

CREATE PROCEDURE new\_schedule\_test(data date, day varchar, class integer, sub integer, s\_time time, e\_time time)

LANGUAGE plpgsql AS $$

begin

if (select exists (select start\_time from schedule where ddate=data and id\_subject=sub))=true

then raise notice 'Урок у учителя';

else

if (select exists (select start\_time from schedule where ddate=data and id\_class=class))=true

then raise notice 'Урок у класса';

else

CASE

WHEN class>=1 AND class<=4

THEN if (select count(\*) from schedule where ddate=data and id\_class=class)>4

then raise notice 'Превышен лимит уроков';

else insert into schedule values(data,day,class,sub,s\_time, e\_time);

end if;

WHEN class>=5 AND class<=9

THEN if (select count(\*) from schedule where ddate=data and id\_class=class)>6

then raise notice 'Превышен лимит уроков';

else insert into schedule values(data,day,class,sub,s\_time, e\_time);

end if;

WHEN class>=10 AND class<=11

THEN if (select count(\*) from schedule where ddate=data and id\_class=class)>8

then raise notice 'Превышен лимит уроков';

else insert into schedule values(data,day,class,sub,s\_time, e\_time);

end if;

END case;

end if;

end if;

end;

$$;

CALL new\_schedule\_test('2023-12-22','Пятница',5, 11, '10:40','11:25');

CREATE PROCEDURE update\_student(student\_name varchar(100), class integer)

LANGUAGE plpgsql AS $$

begin

if (SELECT COUNT(\*) FROM students where id\_class=class)<25

then UPDATE students SET id\_class=class where full\_name=student\_name;

else RAISE NOTICE 'Класс переполнен';

end if;

end;

$$;

CALL update\_student('Михеев Степан Денисович',5);

**Процедура с транзакцией:**

CREATE PROCEDURE add\_schedule(data date, day varchar, class integer, sub integer, s\_time time, e\_time time)

LANGUAGE plpgsql AS $$

begin

insert into schedule values(data,day,class,sub,s\_time, e\_time);

if(select data>now())

then commit;

raise notice 'Расписание обновлено';

else rollback;

raise notice 'Отмена ввода';

end if;

end;

$$;

CALL add\_schedule('2023-12-23','Пятница',5, 12, '10:40','11:25');

**Триггер:**

CREATE OR REPLACE FUNCTION add\_schedule\_fun()

RETURNS TRIGGER AS $$

declare

dat\_int integer;

dat varchar;

id\_sub integer;

s\_time time;

e\_time time;

BEGIN

SELECT EXTRACT(DOW FROM (SELECT ddate FROM schedule ORDER BY ddate DESC LIMIT 1)) into dat\_int;

SELECT id\_subject FROM schedule ORDER BY ddate DESC LIMIT 1 into id\_sub;

CASE WHEN dat\_int=1 THEN dat='Понедельник';

WHEN dat\_int=2 THEN dat='Вторник';

WHEN dat\_int=3 THEN dat='Среда';

WHEN dat\_int=4 THEN dat='Четверг';

WHEN dat\_int=5 THEN dat='Пятница';

WHEN dat\_int=6 THEN dat='Суббота';

END case;

CASE WHEN id\_sub=1 THEN s\_time='8:30'; e\_time='9:15';

WHEN id\_sub=2 THEN s\_time='9:25'; e\_time='10:10';

WHEN id\_sub=3 THEN s\_time='8:30'; e\_time='9:15';

WHEN id\_sub=4 THEN s\_time='10:30'; e\_time='11:15';

WHEN id\_sub=5 THEN s\_time='11:30'; e\_time='12:15';

WHEN id\_sub=6 THEN s\_time='8:30'; e\_time='9:15';

WHEN id\_sub=7 THEN s\_time='11:30'; e\_time='12:15';

WHEN id\_sub=8 THEN s\_time='13:10'; e\_time='10:10';

WHEN id\_sub=9 THEN s\_time='12:25'; e\_time='13:10';

WHEN id\_sub=10 THEN s\_time='9:25'; e\_time='10:10';

END case;

UPDATE schedule SET week\_day = dat WHERE ddate=(SELECT ddate FROM schedule ORDER BY ddate DESC LIMIT 1);

UPDATE schedule SET start\_time = s\_time, end\_time = e\_time WHERE ddate=(SELECT ddate FROM schedule ORDER BY ddate DESC LIMIT 1);

return new;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER add\_inf\_schedule

after INSERT ON schedule

FOR EACH ROW

EXECUTE PROCEDURE add\_schedule\_fun();

**Роли:**

CREATE ROLE Директор LOGIN;

CREATE ROLE Завуч LOGIN;

CREATE ROLE Диспетчер LOGIN;

GRANT CREATE, USAGE ON SCHEMA public to Директор;

GRANT SELECT ON employees TO Завуч;

GRANT UPDATE ON schedule,subject,classes,classes\_types,students to Завуч;

GRANT SELECT ON subject,classes,classes\_types TO Диспетчер;

GRANT UPDATE ON schedule to Завуч;

GRANT UPDATE(name,id\_employee) ON subject TO Завуч;

https://sysadminium.ru/privilegii\_v\_postgresql/