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,

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

create table schedule(

ddate date primary key,

week\_day varchar(50),

id\_class integer,

id\_subject integer Unique, (попробовать primary здесь для связи нескольких предметов в 1 день)

start\_time time,

end\_time time)

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 FROM subject s, employees e

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

CREATE VIEW schedule\_7B AS select\* from schedule

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

CREATE VIEW schedule\_Voronin AS select \* from schedule

where id\_subject in (SELECT s.id FROM subject s, employees e WHERE s.id\_employee=e.id and e.full\_name='Воронин Максим Артемьевич')

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

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

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

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

CREATE VIEW qty\_in\_classes AS SELECT id\_class, COUNT(\*) FROM students

GROUP BY id\_class;

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 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);