МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАТАРСТАН
Государственное автономное профессиональное образовательное учреждение
«ЗЕЛЕНОДОЛЬСКИЙ МЕХАНИЧЕСКИЙ КОЛЛЕДЖ»
(ГАПОУ «ЗМК»)

09.02.07 «Информационные системы и программирование»

МДК 07.01. УПРАВЛЕНИЕ И АВТОМАТИЗАЦИЯ БАЗ ДАННЫХ

Отчет о практических работах

Исполнитель: Чернов Алексей Александрович
Группа: 227
Преподаватель: Алемасов Евгений Павлович

Дата сдачи 04.11.2024 **Оценка _____**

Подпись преподавателя _____

ЗЕЛЕНОДОЛЬСК – 2024

```
create table specialization(
specialization_id smallserial primary key,
specialization_name varchar(30) not null
);

create table doctor(
doctor_id smallserial primary key,
doctor_name varchar(50) not null,
doctor_firstname varchar(50) not null,
doctor_secondname varchar(50),
doctor_percent numeric(4,3) not null check (doctor_percent >= 0 AND
doctor_percent <= 1),
doctor_specialization_id smallint not null,
foreign key (doctor_specialization_id) references
specialization(specialization_id) on delete cascade on update
cascade
);

create table patient(
patient_id serial primary key,
patient_name varchar(50) not null,
patient_firstname varchar(50) not null,
patient_secondname varchar(50),
patient_birthday date not null,
patient_address varchar(120) not null
);

create table reception(
```

```
reception_id bigserial primary key,
reception_price money not null,
reception_date date not null,
doctor_id smallint not null,
patient_id int not null,
foreign key (doctor_id) references doctor(doctor_id) on delete cascade on update cascade,
foreign key (patient_id) references patient(patient_id) on delete cascade on update cascade
);

insert into specialization values
(default, 'Хирург'),
(default, 'Анестезиолог'),
(default, 'Кардиолог'),
(default, 'Нейрохирург'),
(default, 'Ортопед'),
(default, 'Гинеколог'),
(default, 'Пластический хирург'),
(default, 'Эндокринолог'),
(default, 'Травматолог'),
(default, 'Онколог');

insert into doctor values
(default, 'Нурислам', 'Фахрутдинов', '', 0.38, 1),
(default, 'Алексей', 'Чернов', 'Александрович', 0.21, 2),
(default, 'Максим', 'Петров', 'Георгиевич', 0.12, 3),
(default, 'Булат', 'Сабиров', 'Радикович', 0.15, 4),
(default, 'Андрей', 'Бушев', 'Сергеевич', 0.47, 5),
```

```
(default, 'Никита', 'Гришин', '', 0.09, 6),
(default, 'Ислам', 'Габуров', '', 0.03, 7),
(default, 'Дмитрий', 'Гудихин', 'Сергеевич', 0.1, 8),
(default, 'Джеймс', 'Бонд', '', 0.17, 9),
(default, 'Пит', 'Бред', '', 0.32, 10);

insert into patient values
(default, 'Уил', 'Смит', '', '07/06/06', 'ул. Фрунзе 1'),
(default, 'Пит', 'Бред', '', '07/06/2005', 'ул. Фрунзе 2'),
(default, 'Том', 'Харди', '', '07/06/2004', 'ул. Фрунзе 3'),
(default, 'Джони', 'Деп', '', '07/06/2003', 'ул. Фрунзе 4'),
(default, 'Эмма', 'Вотсон', '', '07/06/2002', 'ул. Фрунзе 5'),
(default, 'Эмили', 'Вилис', '', '07/06/2001', 'ул. Фрунзе 6'),
(default, 'Тони', 'Кларк', '', '07/06/2000', 'ул. Фрунзе 7'),
(default, 'Маргерет', 'Течер', '', '07/06/2007', 'ул. Фрунзе 8'),
(default, 'Тони', 'Старк', '', '07/06/2008', 'ул. Фрунзе 9'),
(default, 'Селена', 'Гомес', '', '07/06/2009', 'ул. Фрунзе 10');

insert into reception values
(default, 2000.0, '15/10/2024', 1, 1),
(default, 2001.0, '16/10/2024', 2, 2),
(default, 2002.0, '17/10/2024', 3, 3),
(default, 2003.0, '18/10/2024', 4, 4),
(default, 2004.0, '18/10/2024', 5, 5),
(default, 2005.0, '17/10/2024', 6, 6),
(default, 2006.0, '16/10/2024', 7, 7),
(default, 2007.0, '15/10/2024', 8, 8),
(default, 2008.0, '19/10/2024', 9, 9),
(default, 2009.0, '19/10/2024', 10, 10);
```

```
create table specialization_dell(
specialization_id smallint primary key,
specialization_name varchar(30) not null
);

create table doctor_dell(
doctor_id smallint primary key,
doctor_name varchar(50) not null,
doctor_firstname varchar(50) not null,
doctor_secondname varchar(50),
doctor_percent numeric(4,3) not null check (doctor_percent >= 0 AND
doctor_percent <= 1),
doctor_specialization_id smallint not null
);

create table patient_dell(
patient_id int primary key,
patient_name varchar(50) not null,
patient_firstname varchar(50) not null,
patient_secondname varchar(50),
patient_birthday date not null,
patient_address varchar(120) not null
);

create table reception_dell(
reception_id bigint primary key,
reception_price money not null,
reception_date date not null,
```

```
doctor_id smallint not null,
patient_id int not null
);

--Мягкое удаление
create or replace function del_specialization()
returns trigger as $$
begin
insert into specialization_dell values
(old.specialization_id, old.specialization_name);
end;
$$ language plpgsql;

create trigger specialization_del
before delete on specialization
for each row
execute function del_specialization();



create or replace function del_doctor()
returns trigger as $$
begin
insert into doctor_dell values
(old.doctor_id, old.doctor_name, old.doctor_firstname,
old.doctor_secondname, old.doctor_percent,
old.doctor_specialization_id);
end;
$$ language plpgsql;
```

```
create trigger doctor_del
before delete on doctor_dell
for each row
execute function del_doctor();



create or replace function del_patien()
returns trigger as $$
begin
insert into patient_dell values
(old.patient_id, old.patient_name, old.patient_firstname,
old.patient_secondname,
old.patient_birthday, old.patient_address);
end;
$$ language plpgsql;

create trigger patient_del
before delete on patient_dell
for each row
execute function del_patien();



create or replace function del_reception()
returns trigger as $$
begin
insert into reception_dell values
```

```
(old.reception_id, old.reception_price, old.reception_date,
old.doctor_id, old.patient_id);

end;

$$ language plpgsql;


create trigger reception_del

before delete on reception_dell

for each row

execute function del_reception();




create or replace function valid_specialization()

returns trigger as $$

begin

if new.specialization_name in ('Хирург', 'Анестезиолог',
'Кардиолог', 'Нейрохирург',

'Ортопед', 'Гинеколог', 'Пластический хирург', 'Эндокринолог',
'Травматолог', 'Онколог') then

return new;

else

raise notice 'Недопустимая специализация: %',
new.specialization_name;

return null;

end if;

end;

$$ language plpgsql;


create trigger specialization_valid

before insert or update on specialization
```

```
for each row
execute function valid_specialization();

create or replace function valid_doctor()
returns trigger as $$
begin
if (new.doctor_secondname is null) then
if left(new.doctor_name, 1) != upper(left(new.doctor_name, 1)) then
raise notice 'Имя должно начинаться с большой буквы';
return null;
end if;

if left(new.doctor_firstname, 1) != upper(left(new.doctor_firstname,
1)) then
raise notice 'Фамилия должно начинаться с большой буквы';
return null;
end if;

if new.doctor_specialization_id not in (select specialization_id
from specialization) then
raise notice 'Укажите доставерный id специальности';
return null;
end if;

else
if left(new.doctor_name, 1) != upper(left(new.doctor_name, 1)) then
raise notice 'Имя должно начинаться с большой буквы';
return null;
end if;
```

```
if left(new.doctor_firstname, 1) != upper(left(new.doctor_firstname, 1)) then
raise notice 'Фамилия должно начинаться с большой буквы';
return null;
end if;

if left(new.doctor_secondname, 1) != upper(left(new.doctor_secondname, 1)) then
raise notice 'Отчество должно начинаться с большой буквы';
return null;
end if;

if new.doctor_specialization_id not in (select specialization_id from specialization) then
raise notice 'Укажите доставерный id специальности';
return null;
end if;

end if;
return new;
end;
$$ language plpgsql;

create trigger doctor_valid
before insert or update on doctor
for each row
execute function valid_doctor();
```

```
create or replace function valid_patient()

returns trigger as $$

begin

if (new.patient_secondname is null) then

if left(new.patient_name, 1) != upper(left(new.patient_name, 1))
then

raise notice 'Имя должно начинаться с большой буквы';

return null;

end if;


if left(new.patient_firstname, 1) !=
upper(left(new.patient_firstname, 1)) then

raise notice 'Фамилия должно начинаться с большой буквы';

return null;

end if;


if (new.patient_birthday < 04/11/1914 and new.patient_birthday >
now()) then

raise notice 'Укажите доставерную дату рождения';

return null;

end if;


else

if left(new.patient_name, 1) != upper(left(new.patient_name, 1))
then

raise notice 'Имя должно начинаться с большой буквы';

return null;

end if;
```

```
if left(new.patient_firstname, 1) !=
upper(left(new.patient_firstname, 1)) then

raise notice 'Фамилия должно начинаться с большой буквы';

return null;

end if;


if left(new.patient_secondname, 1) !=
upper(left(new.patient_secondname, 1)) then

raise notice 'Отчество должно начинаться с большой буквы';

return null;

end if;


if (new.patient_birthday < 04/11/1914 and new.patient_birthday >
now()) then

raise notice 'Укажите доставерную дату рождения';

return null;

end if;


end if;

return new;

end;

$$ language plpgsql;


create trigger patient_valid

before insert or update on patient

for each row

execute function valid_patient();


create or replace function valid_reception()
```

```
returns trigger as $$
begin
if (new.reception_date > now() and new.reception_date < 04/11/1994)
then --Предположим что больнице 30 лет
raise notice 'Укажите доставерную дату приёма';
return null;
end if;

if (new.doctor_id not in (select doctor_id from doctor)) then
raise notice 'Укажите существующий id врача';
return null;
end if;

if (new.patient_id not in (select patient_id from patient)) then
raise notice 'Укажите существующий id пациента';
return null;
end if;

end;
$$ language plpgsql;

create trigger reception_valid
before insert or update on reception
for each row
execute function valid_reception();
```