**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**

**ФГБОУ ВПО "Пензенский государственный университет"**

**Кафедра "Математическое обеспечение и применение ЭВМ"**

Отчет

по лабораторной работе №5

**«Формирование триггеров в базе данных»**

Выполнили:

Хамзин Х.

Суворов М.

Хусаинов И.

ст. гр. 17ВИ1

Проверил:

доцент Казакова И.А.

**2020**

1. После добавления комнаты добавить туда набор включенных услуг 10 и 6

CREATE OR REPLACE FUNCTION insert\_room() RETURNS TRIGGER AS $body$

DECLARE

mstr varchar(30);

BEGIN

insert into service\_for\_room(id\_room, id\_service) VALUES (new.id\_room, 6);

insert into service\_for\_room(id\_room, id\_service)values (new.id\_room, 10);

return new;

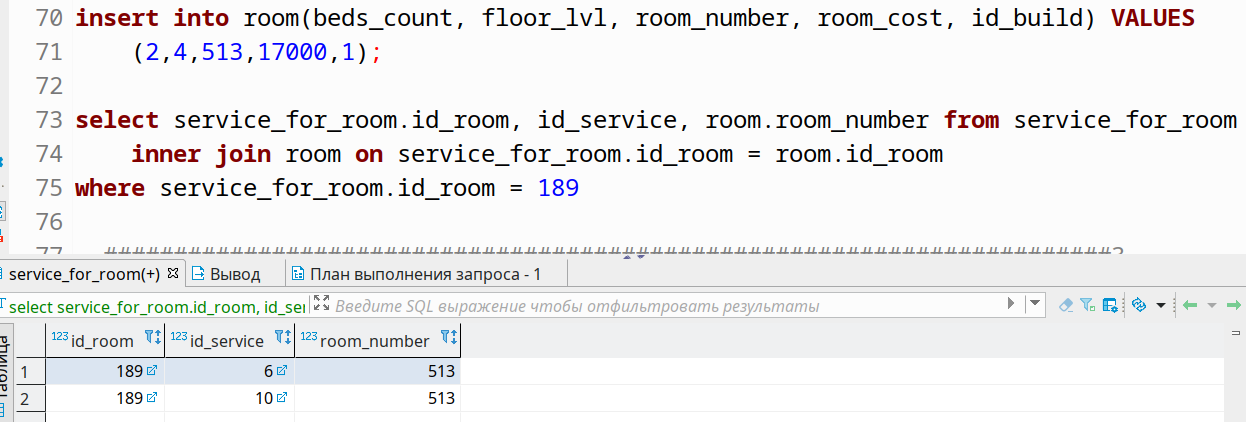
END;

$body$ LANGUAGE plpgsql;

CREATE TRIGGER t\_add\_room

AFTER INSERT ON room FOR EACH ROW EXECUTE PROCEDURE insert\_room();

Проверка работы триггера



2.Добавление купленных номеров в счет

create or replace function room\_in\_bill() RETURNS TRIGGER AS $body$

declare

cost\_room money;

bill\_id int;

bilt\_position\_id integer;

is\_fiz\_person bool;

is\_check integer;

begin

select into bill\_id id\_bill

from contract

inner join booking on booking.id\_contract = contract.id\_contract

where booking.id\_contract = new.id\_contract;

select into is\_check client.id\_client from contract

inner join bill on bill.id\_bill = contract.id\_bill

inner join client on client.id\_client = contract.id\_client

inner join fiz\_person on fiz\_person.id\_client = client.id\_client

where bill.id\_bill = bill\_id;

if( is\_check is NULL) then

is\_fiz\_person = false;

else

is\_fiz\_person = true;

end if;

select into cost\_room room\_cost

from room

where room.id\_room = new.id\_room;

insert into bilt\_position(id\_bill,id\_disscount,without\_vat,without\_disscount, id\_room)

values(bill\_id,7,cost\_room,cost\_room,new.id\_room)

returning id\_bilt\_position into bilt\_position\_id;

perform update\_cost\_bilt\_position(NEW.id\_room,0,7,is\_fiz\_person,bilt\_position\_id);

return NULL;

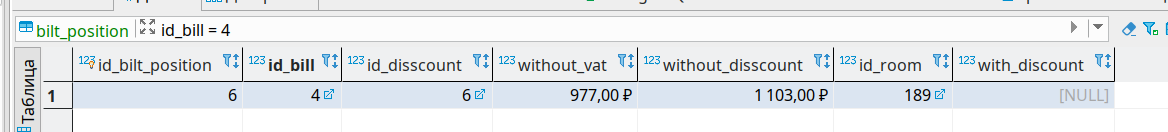
end

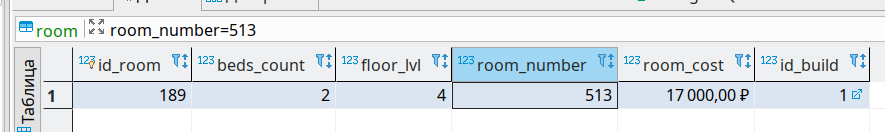
$body$ LANGUAGE plpgsql;

create trigger add\_room\_bill

after insert on booking for each row execute procedure room\_in\_bill();

Проверка работы триггера  
  
select booking\_room('15.05.2020','11.06.2020',513,4,2);





3.Не заселять в номер человека, если там не осталось спальных мест

create or replace function fullnes\_room() RETURNS TRIGGER AS $body$

declare

count\_person int;

room\_size int;

begin

select into room\_size beds\_count

from client\_booking

inner join booking on booking.id\_booking = client\_booking.id\_booking

inner join room on room.id\_room = booking.id\_booking

where client\_booking.id\_booking = new.id\_booking;

raise log 'room\_size - %', room\_size;

select into count\_person count(id\_client)

from client\_booking

where id\_booking = new.id\_booking;

raise log 'count\_person - %', count\_person;

if(count\_person > room\_size) then

raise log 'Номер заполнен';

return null;

else

return new;

end if;

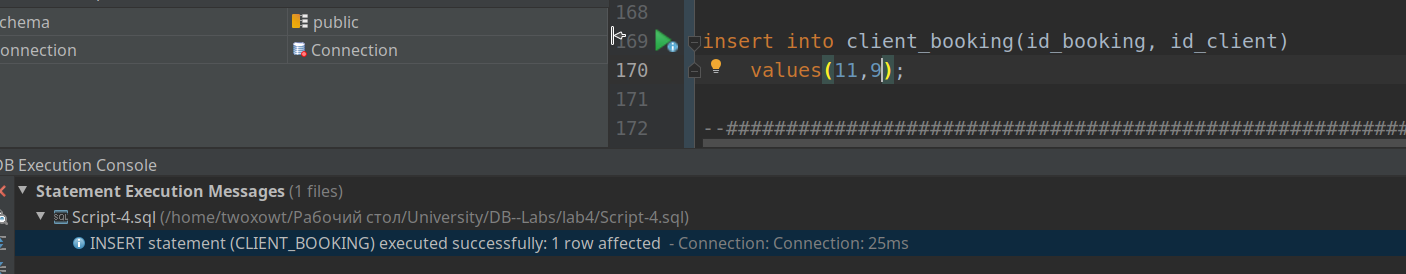
end

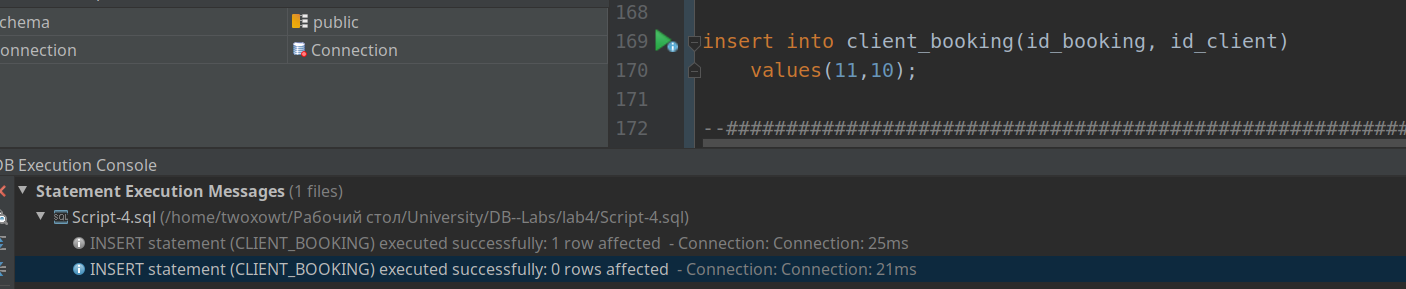
$body$ LANGUAGE plpgsql;

create trigger check\_fullnes\_room

before insert on client\_booking for each row execute procedure fullnes\_room();

Результат работы

добавляем 1 клиента в номер  


добавляемеще одного клиента в тот же номер  
  
0 строк добавлено , тк номер одноместный

4.Не давать пользователю ввести две одинаковые услуги

При попытки изменить имя услуги на уже существующее откатывать транзакцию\*/

CREATE OR REPLACE FUNCTION add\_service() RETURNS TRIGGER AS $body$

DECLARE

mstr varchar(30);

BEGIN

IF TG\_OP = 'INSERT' THEN

IF exists(select service\_name from service where service\_name = NEW.service\_name) then

RAISE EXCEPTION 'Запись уже сущесвтует';

end if;

INSERT INTO service( service\_name, service\_cost) values (new.service\_name, new.service\_cost);

return new;

ELSIF TG\_OP = 'UPDATE' THEN

if new.service\_name <> old.service\_name THEN

IF exists(select service\_name from service where service\_name = NEW.service\_name) then

RAISE EXCEPTION 'Невозмжно навазать имя услуги таким образом';

end if;

end if;

update service set service\_name = new.service\_name,

service\_cost = new.service\_cost

where id\_service = new.id\_service;

return new;

END IF;

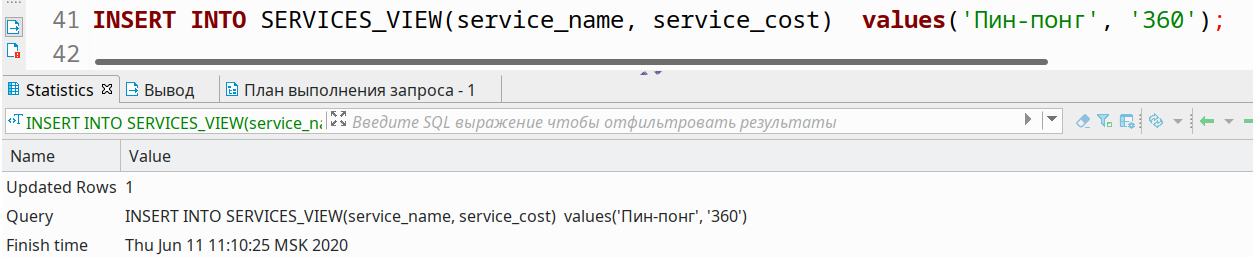
END;

$body$ LANGUAGE plpgsql;

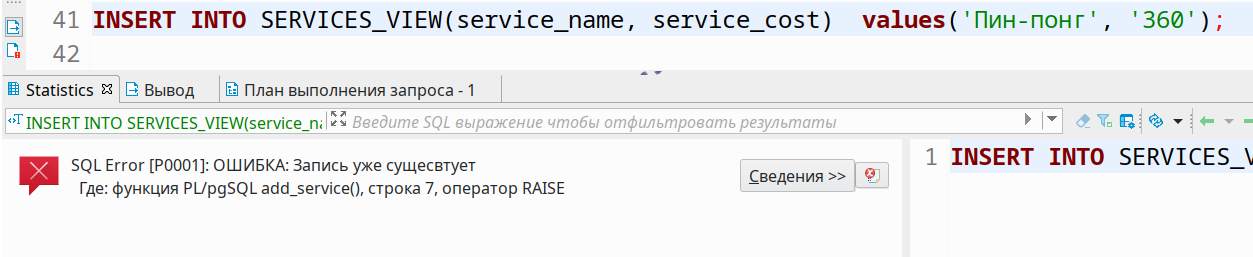
CREATE TRIGGER t\_service

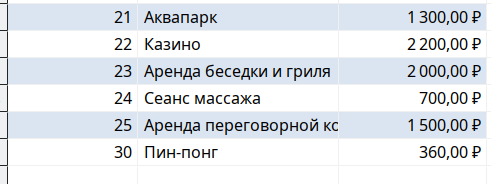
INSTEAD OF INSERT or UPDATE ON SERVICES\_VIEW FOR EACH ROW EXECUTE PROCEDURE add\_service();

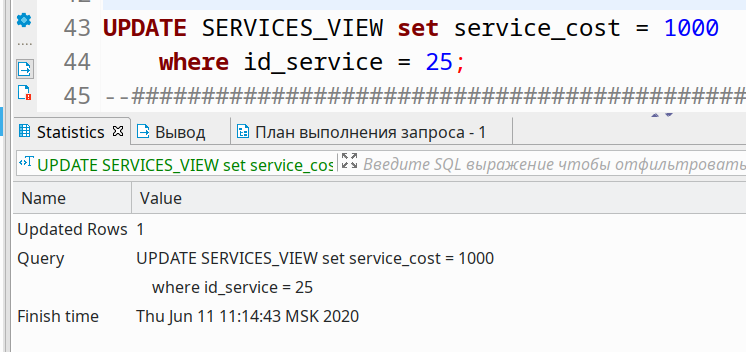
Результат работы

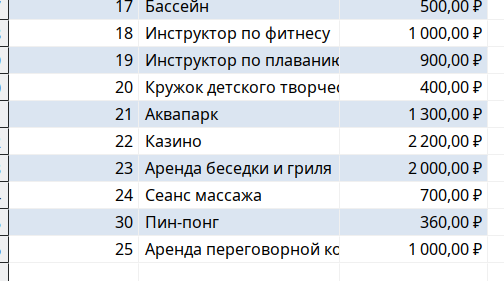
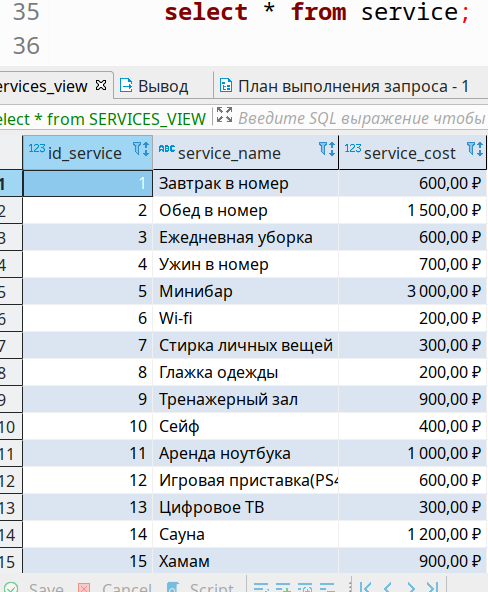
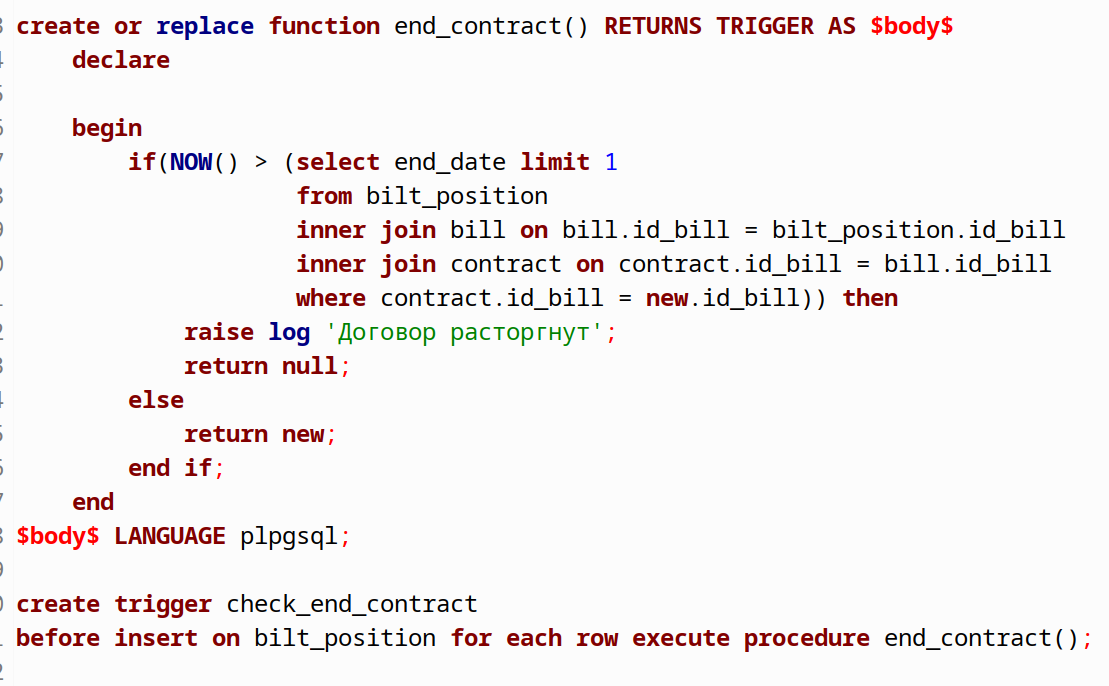


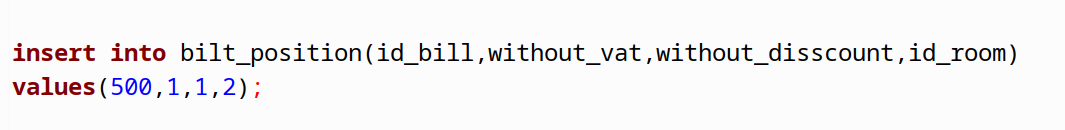
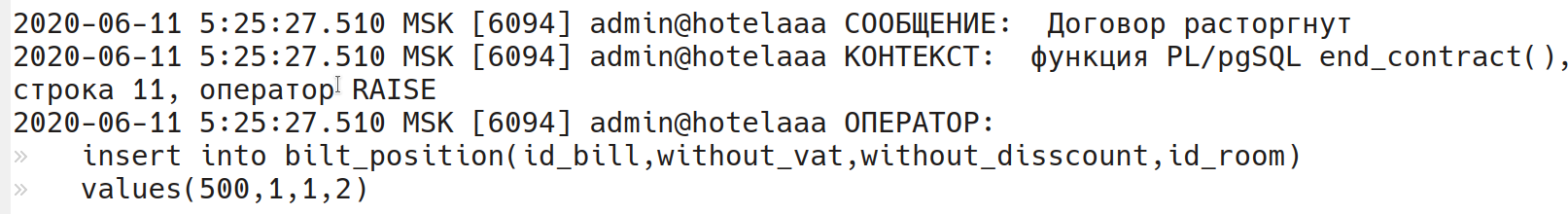
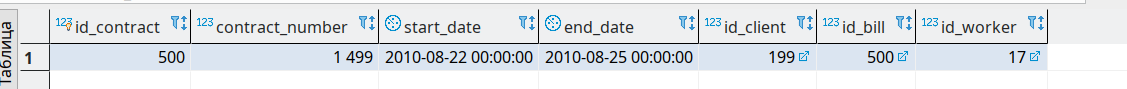
попытка еще раз вызвать insert







  
5. Запретить осуществлять запись в договора, которые уже расторгнуты  
  


Результат работы  
  
  


действие договора закончилось в 2010 году