

**Создание таблицы claim для заявок клиентов**

create table claim

(

id integer primary key ,

id\_account integer not null,

id\_tarif integer not null,

priority integer not null constraint c\_pr check (priority in (1,2,3)),

text\_claim text not null

)

**Добавление в таблицу claim данных**

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(10,10, 10, 1, 'Таким образом реализация намеченных плановых заданий позволяет оценить значение новых предложений.');

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(20,10, 20, 2, 'Равным образом постоянный количественный рост и сфера нашей активности играет важную роль в формировании системы обучения кадров, соответствует насущным потребностям.');

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(30,30, 10, 3, 'Разнообразный и богатый опыт консультация с широким активом обеспечивает широкому кругу.');

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(40,50, 10, 1, 'С другой стороны укрепление и развитие структуры обеспечивает участие в формировании систем массового участия.');

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(50,10, 10, 3, 'Значимость этих проблем настолько очевидна, что консультация с широким активом играет важную роль в формировании новых предложений.');

INSERT INTO public.claim

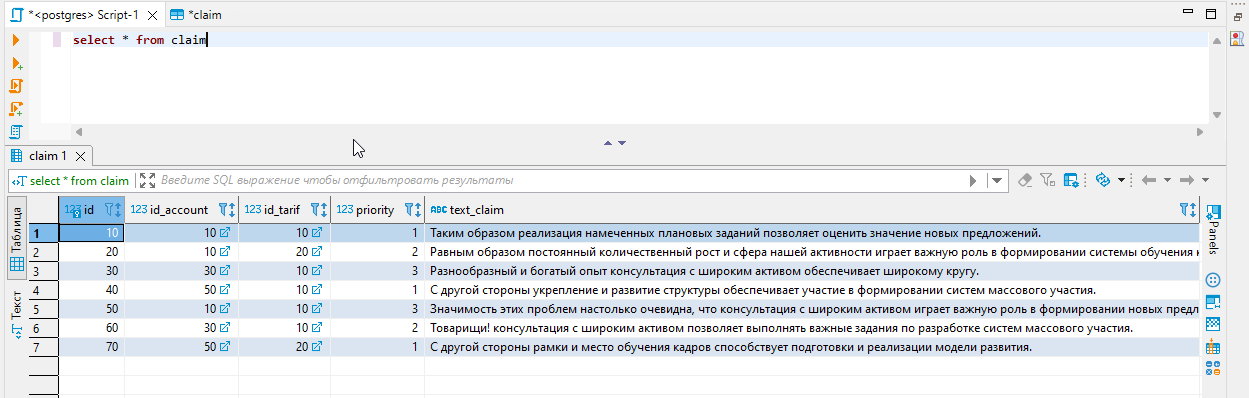
(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(60,30, 10, 2, 'Товарищи! консультация с широким активом позволяет выполнять важные задания по разработке систем массового участия.');

INSERT INTO public.claim

(id, id\_account, id\_tarif, priority, text\_claim)

VALUES(70,50, 20, 1, 'С другой стороны рамки и место обучения кадров способствует подготовки и реализации модели развития.');



**Добавление внешнего ключа между таблицами claim и account**

alter table claim add constraint claim\_acc

foreign key(id\_account) references account(id)

**Добавление внешнего ключа между таблицами claim и tarif**

alter table claim add constraint claim\_tarif

foreign key(id\_tarif) references tarif(id)

**Создание таблицы tarif\_audit для записи изменений таблицы tarif**

create table tarif\_audit

(

modified\_date TIMESTAMP,

modified\_type VARCHAR,

id integer primary key ,

name varchar not null ,

cost\_per\_day numeric not null

)

**Триггерная функция trg\_fnc\_audit\_tarif на таблицу tarif**

CREATE OR REPLACE FUNCTION public.trg\_fnc\_audit\_tarif()

RETURNS trigger

LANGUAGE plpgsql

AS $$

BEGIN

if (TG\_OP = 'INSERT') then

insert into public.tarif\_audit(modified\_date, modified\_type, id, "name", cost\_per\_day)

values (now(), TG\_OP, new.id, new."name", new.cost\_per\_day);

elseif (TG\_OP = 'UPDATE') then

insert into public.tarif\_audit(modified\_date, modified\_type, id, "name", cost\_per\_day)

values (now(), TG\_OP, old.id, old."name", old.cost\_per\_day);

else

insert into public.tarif\_audit(modified\_date, modified\_type, id, "name", cost\_per\_day)

values (now(), TG\_OP, old.id, old."name", old.cost\_per\_day);

end if;

return null;

END;

$$;

**Триггер trg\_audit\_tarif для таблицы tarif**

create trigger trg\_audit\_tarif after

insert

or

delete

or

update

on

public.tarif for each row execute function trg\_fnc\_audit\_tarif()

**Проверка записи в таблицу tarif\_audit изменений таблицы tarif**

INSERT INTO public.tarif (id, "name", cost\_per\_day)

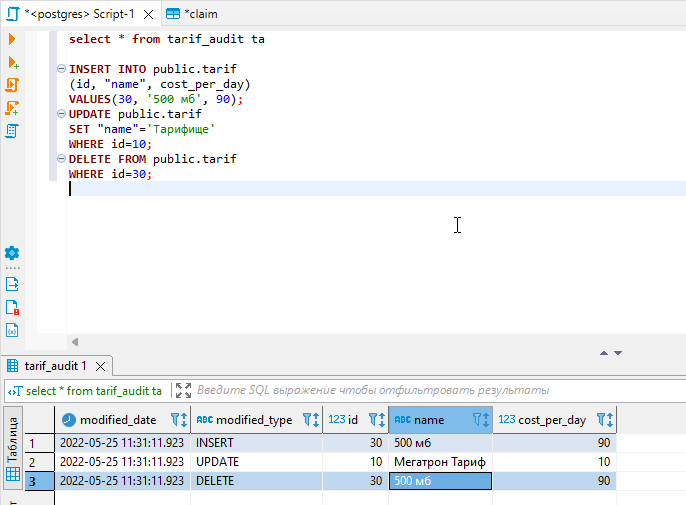
VALUES(30, '500 мб', 90);

UPDATE public.tarif SET "name"='Тарифище'

WHERE id=10;

DELETE FROM public.tarif

WHERE id=30;



**TOP-3 пользователей, у которых имеется наибольшее количество запросов**

with top\_account as (select id\_account , count(id\_account) as count\_claim from claim

group by id\_account

order by count\_claim desc

limit 3

)

select account.email as email, count\_claim from top\_account

inner join account on top\_account.id\_account = account.id

