**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**

**ФГБОУ ВПО "Пензенский государственный университет"**

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

Отчет

по лабораторной работе №3

**«Формирование запросов к базе данных»**

Выполнили:

Хамзин Х.

Суворов М.

Хусаинов И.

ст. гр. 17ВИ1

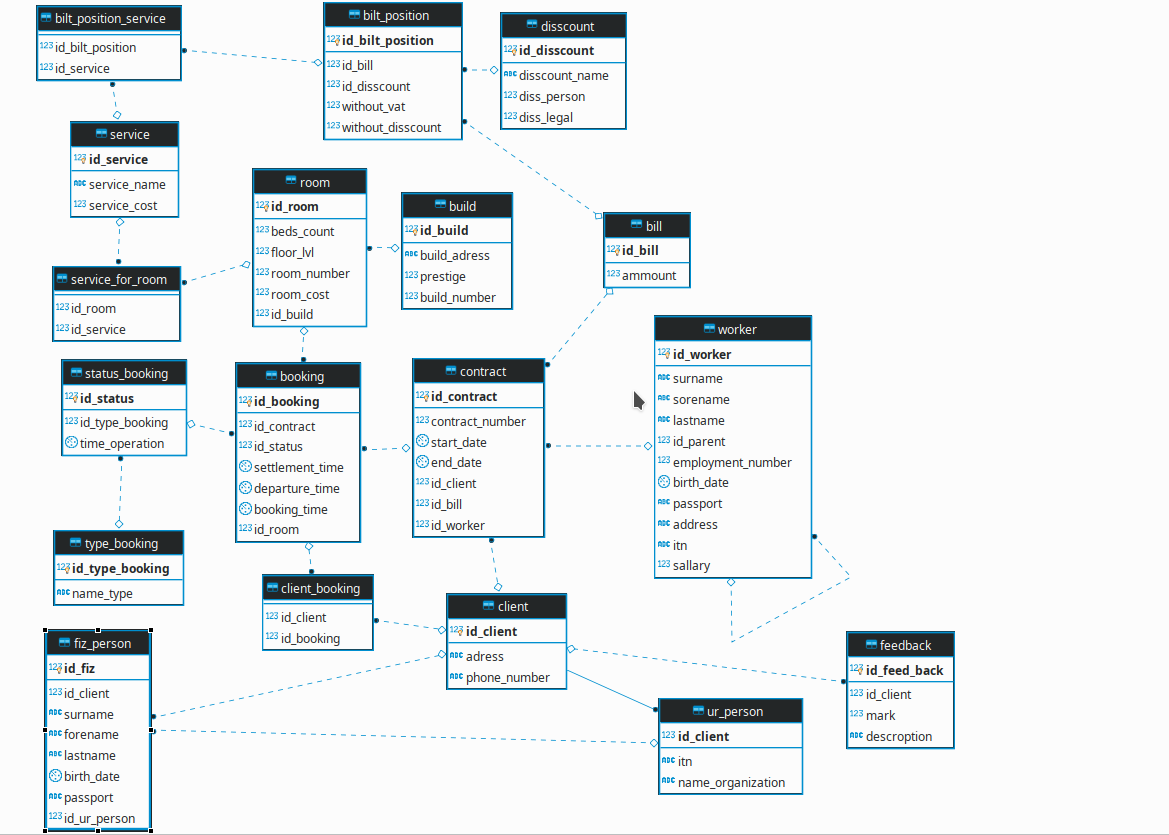
Проверил:

доцент Казакова И.А.

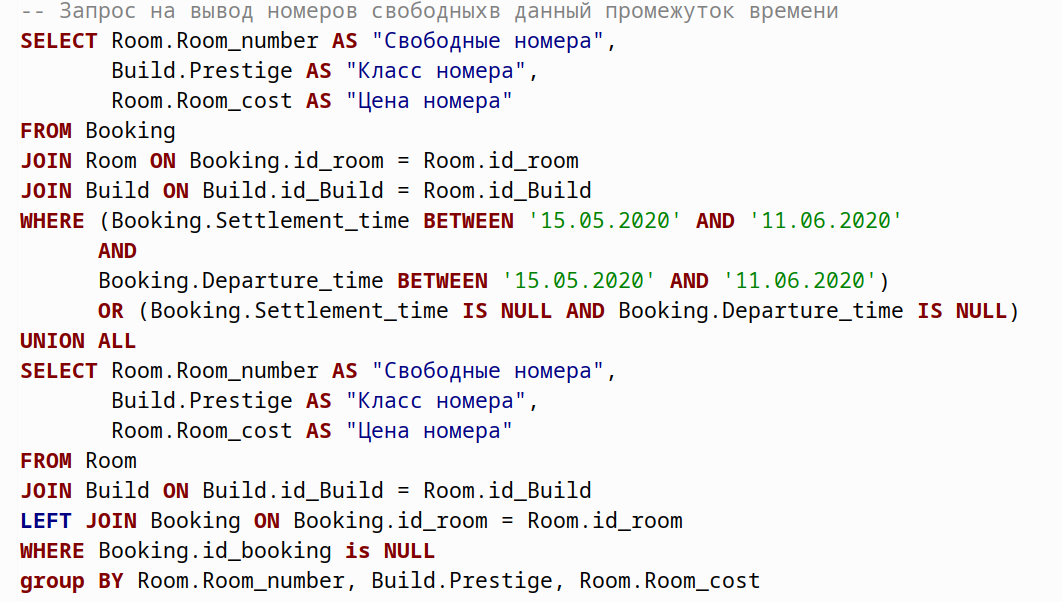
**2020**

**Лабораторная работа №3**

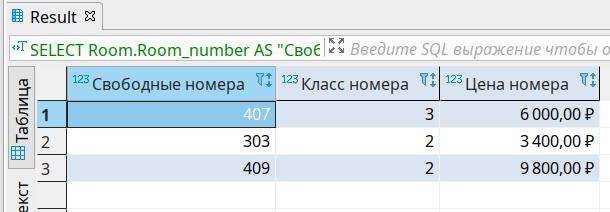
Диаграмма базы данных гостиничного комплекса

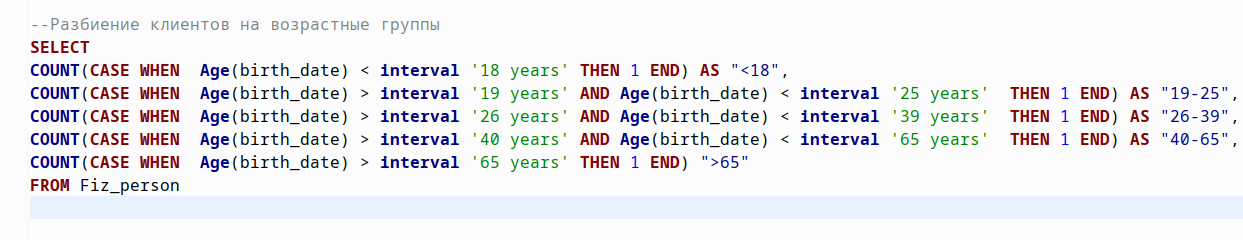


1. Запрос на вывод номеров свободных в данный промежуток времени

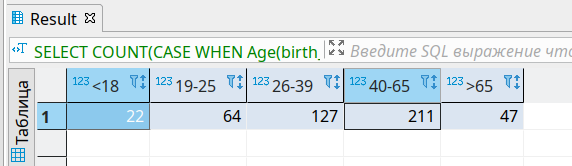


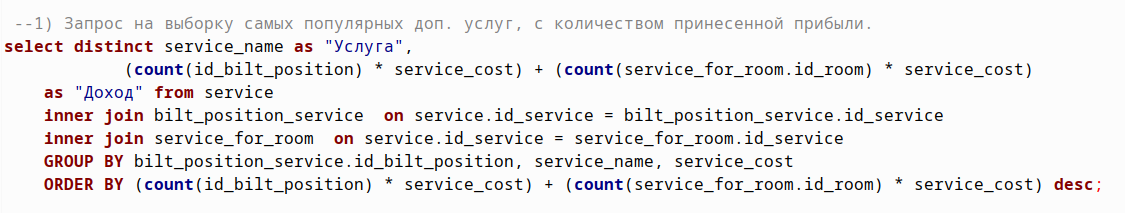
Результат работы запроса:



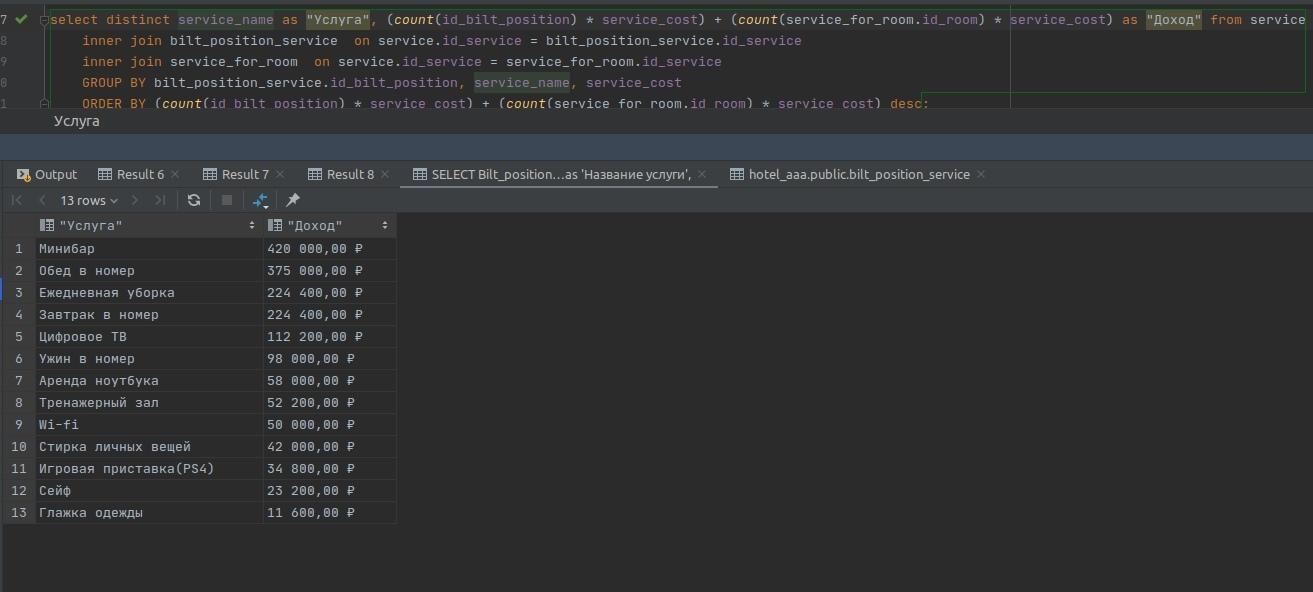
2.Разбиение клиентов на возрастные группы

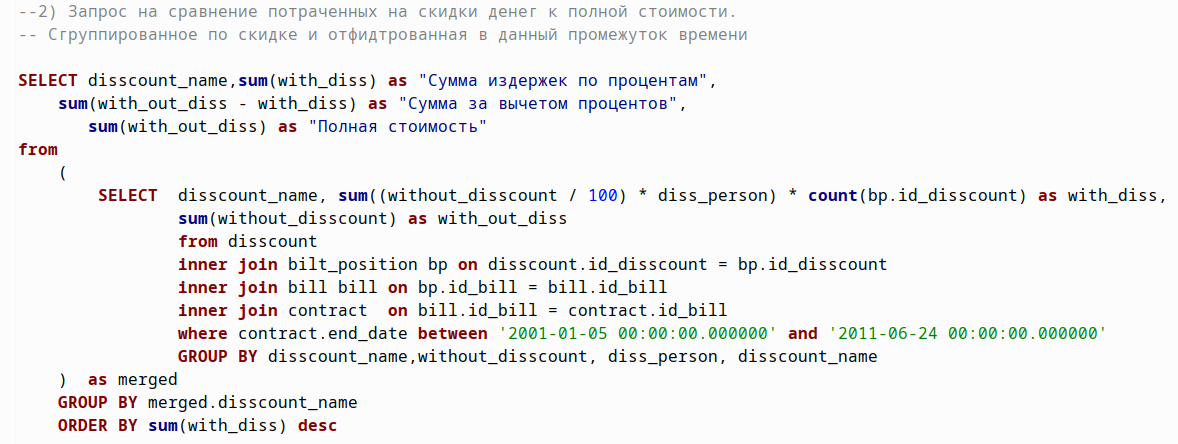
Результат работы запроса:



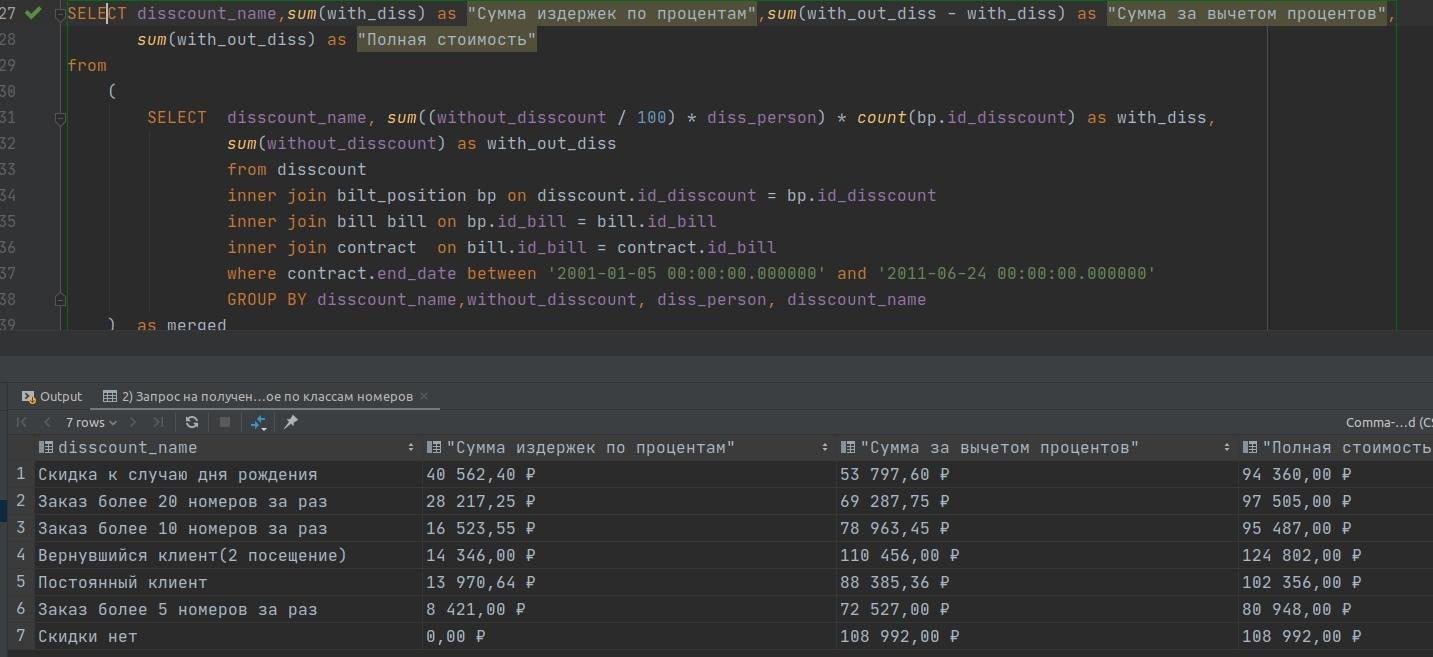
3.Запрос на выборку самых популярных доп. услуг, с количеством принесенной прибыли.

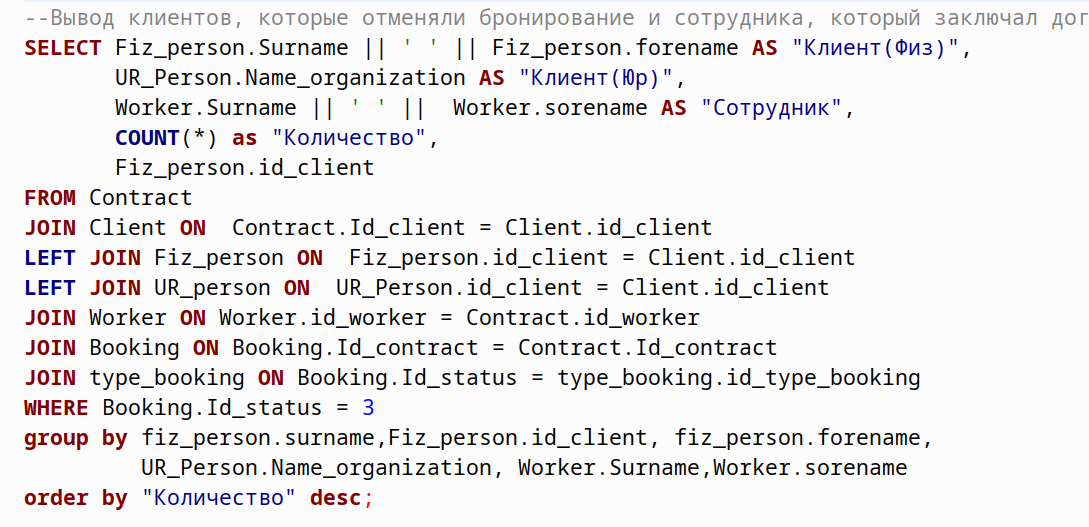
Результат работы запроса:



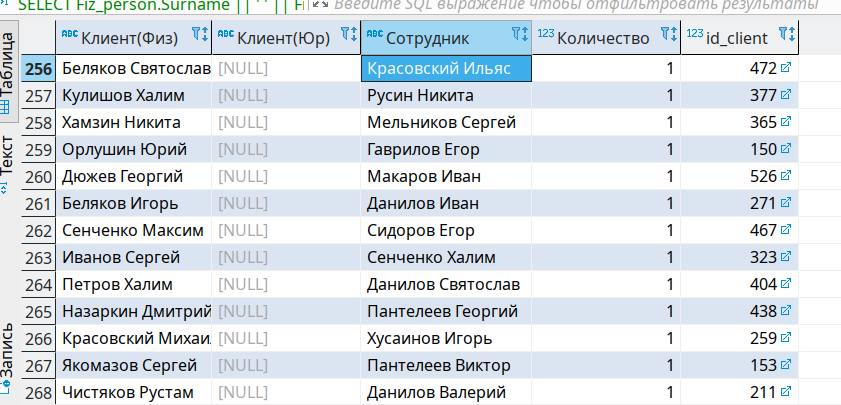
4.Запрос на сравнение потраченных на скидки денег к полной стоимости услуги, сгруппированное по скидке и отсортированная по убыванию.  
В данный промежуток времени

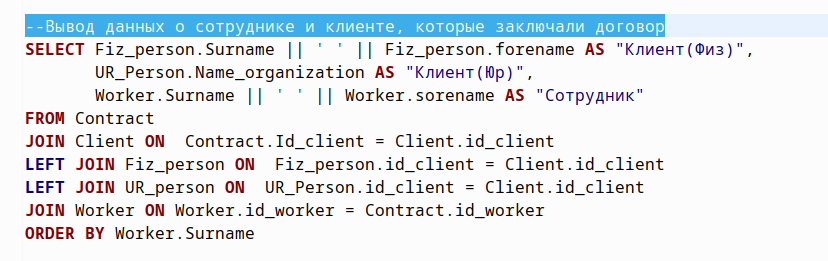
Результат работы запроса:



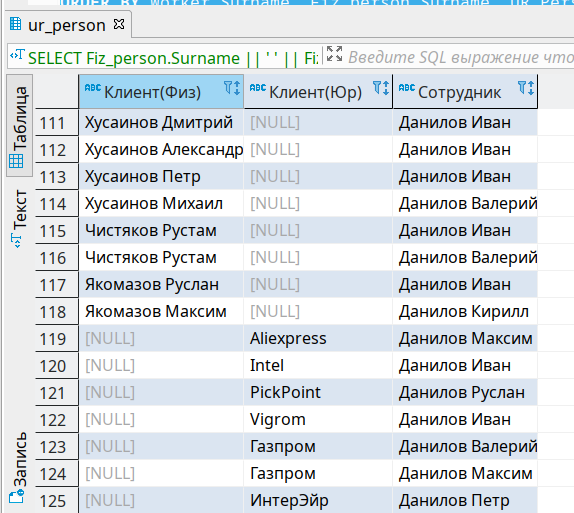
5.Вывод клиентов, которые отменяли бронирование и сотрудника, который заключал договор с клиентом

Результат работы запроса:



6.Вывод данных о сотруднике и клиенте, которые заключали договор

Результат работы запроса:



Представления:

1.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;

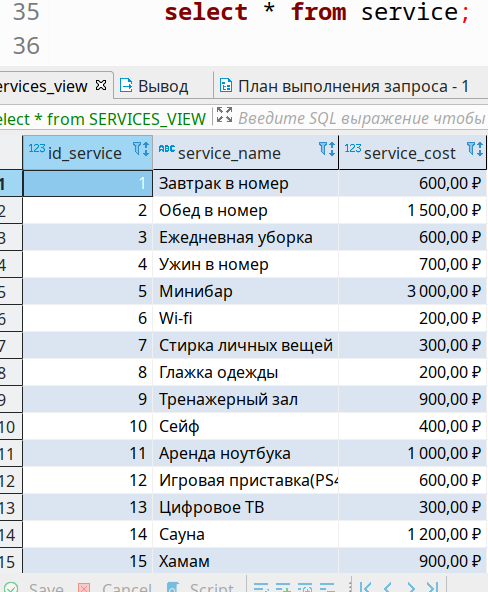
$body$ LANGUAGE plpgsql;

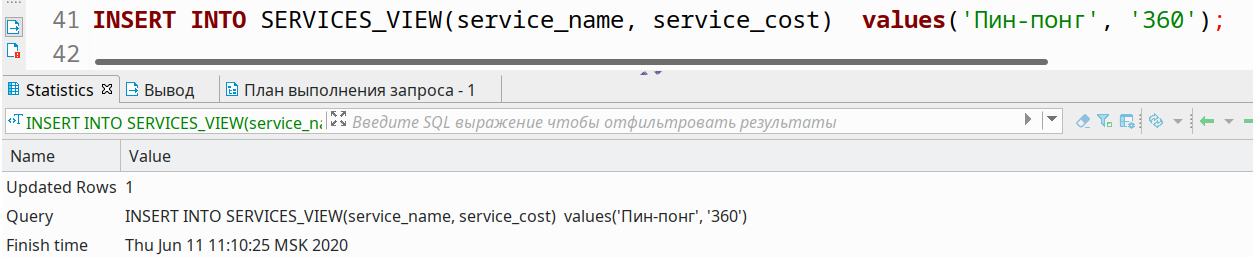
CREATE TRIGGER t\_service

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

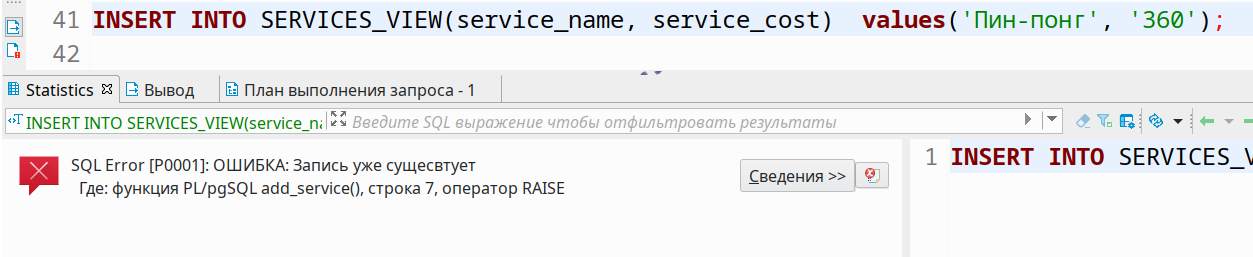
CREATE VIEW SERVICES\_VIEW AS

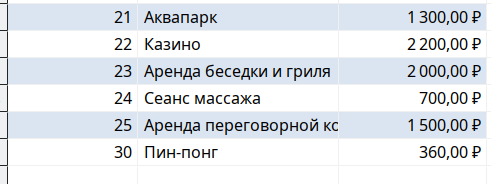
select \* from service;

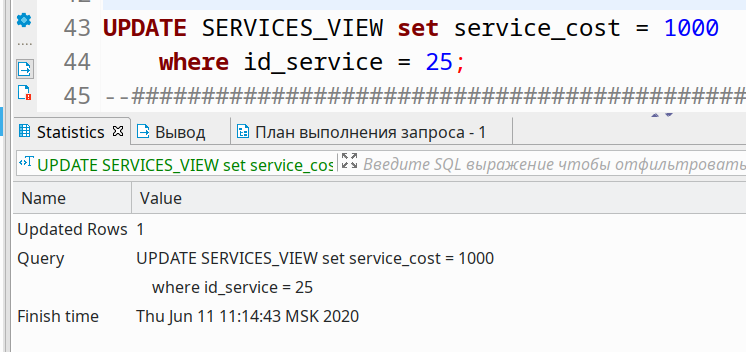


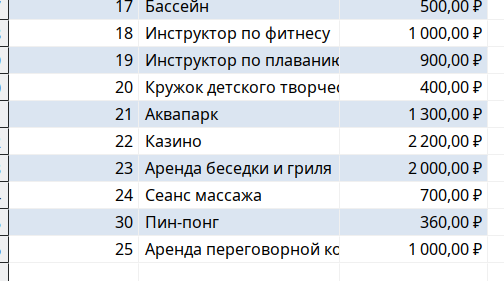


попытка еще раз вызвать insert









2. Сортированные по убыванию прибыли сумма всех доходв принесенная номерами , сгруперованное по престижу

CREATE VIEW MOST\_VALUABLE AS

SELECT Prestig as "Престиж", sum(Summer) as "Сумма" from (

SELECT prestige as Prestig,

sum(room\_cost \* EXTRACT(EPOCH FROM b.departure\_time - b.settlement\_time) /

3600) as Summer

from build

inner join room on build.id\_build = room.id\_build

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

inner join status\_bookiКопия Новый документng sb on b.id\_status = sb.id\_status

inner join type\_booking tb on sb.id\_type\_booking = tb.id\_type\_booking

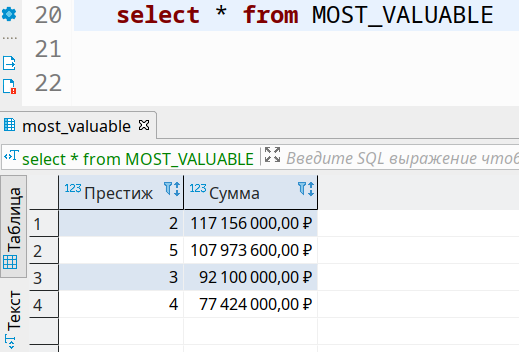
WHERE sb.id\_type\_booking = 1

GROUP BY prestige

) as PS

GROUP BY Prestig

ORDER BY sum(Summer) desc ;



3.Анализ целевой аудитории  
create view target audience as

SELECT

COUNT(CASE WHEN Age(birth\_date) < interval '18 years' THEN 1 END) AS "<18",

COUNT(CASE WHEN Age(birth\_date) > interval '19 years' AND Age(birth\_date) < interval '25 years' THEN 1 END) AS "19-25",

COUNT(CASE WHEN Age(birth\_date) > interval '26 years' AND Age(birth\_date) < interval '39 years' THEN 1 END) AS "26-39",

COUNT(CASE WHEN Age(birth\_date) > interval '40 years' AND Age(birth\_date) < interval '65 years' THEN 1 END) AS "40-65",

COUNT(CASE WHEN Age(birth\_date) > interval '65 years' THEN 1 END) ">65"

FROM Fiz\_person

Результат:

