# Практическая работа №5

Студент: Болдинов Алексей ЭФМО-02-24

Создание таблиц:

CREATE TABLE public."districts"

(

id bigint,

name character(30),

PRIMARY KEY (id)

);

CREATE TABLE public.property

(

id bigint,

id\_district bigint,

address character,

floor bigint,

rooms bigint,

type bigint,

status bigint,

cost double precision,

description\_property text,

square double precision,

announcement\_date date,

PRIMARY KEY (id),

CONSTRAINT id\_district FOREIGN KEY (id\_district)

REFERENCES public.districts (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

);

CREATE TABLE public.evaluation\_criteria

(

id bigint,

name character,

PRIMARY KEY (id)

);

CREATE TABLE public.evaluation

(

id bigint,

id\_property bigint,

date timestamp without time zone,

id\_criteria bigint,

evaluate double precision,

PRIMARY KEY (id),

CONSTRAINT id\_property FOREIGN KEY (id\_property)

REFERENCES public.property (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_criteria FOREIGN KEY (id\_criteria)

REFERENCES public.evaluation\_criteria (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

);

CREATE TABLE public.realtor

(

id bigint,

lastname character,

firstname character,

secondname character,

t\_number character,

PRIMARY KEY (id)

);

CREATE TABLE public.sale

(

id bigint,

id\_property bigint,

date timestamp without time zone,

id\_realtor bigint,

cost double precision,

PRIMARY KEY (id),

CONSTRAINT id\_property FOREIGN KEY (id\_property)

REFERENCES public.property (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_realtor FOREIGN KEY (id\_realtor)

REFERENCES public.realtor (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

);

1. Вывести адреса объектов недвижимости, у которых стоимость 1 м2 меньше средней стоимости по району.

SELECT p.address

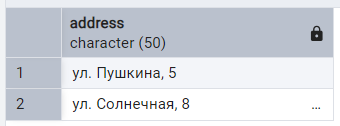
FROM public.property p

WHERE p.cost / p.square <

(SELECT AVG(cost / square)

FROM public.property

WHERE id\_district = p.id\_district);



1. Вывести название районов, в которых количество проданных квартир больше 5.

SELECT d.name

FROM public.districts d

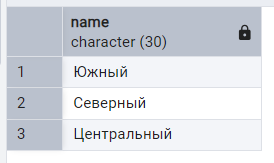
JOIN public.property p ON d.id = p.id\_district

JOIN public.sale s ON p.id = s.id\_property

GROUP BY d.id, d.name

HAVING COUNT(s.id) > 0;

--0 = 5



1. Вывести адреса квартир и название района, средняя оценка которых выше 3,5 баллов.

SELECT p.address, d.name AS district\_name

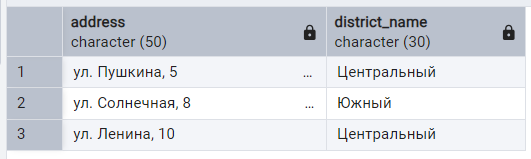
FROM public.property p

JOIN public.districts d ON p.id\_district = d.id

JOIN public.evaluation e ON p.id = e.id\_property

GROUP BY p.id, p.address, d.name

HAVING AVG(e.evaluate) > 3.5;



1. Определить годы, в которых было размещено от 2 до 3 объектов недвижимости.

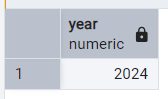
SELECT EXTRACT(YEAR FROM p.announcement\_date) AS year

FROM public.property p

GROUP BY EXTRACT(YEAR FROM p.announcement\_date)

HAVING COUNT(p.id) BETWEEN 0 AND 10;

-- 0 = 2 и 10 = 3



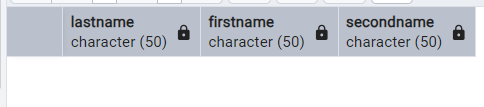
1. Определить ФИО риэлторов, которые ничего не продали в текущем году.

SELECT r.lastname, r.firstname, r.secondname

FROM public.realtor r

LEFT JOIN public.sale s ON r.id = s.id\_realtor AND EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE)

WHERE s.id IS NULL;



1. Определить ФИО риэлторов, продавших квартиры, более чем в двух районах. Предусмотреть вывод ФИО в следующем формате: Иванов И.И.

SELECT

r.lastname || ' ' || LEFT(r.firstname, 1) || '.' || LEFT(r.secondname, 1) || '.' AS full\_name

SELECT

r.lastname || ' ' || LEFT(r.firstname, 1) || '.' || LEFT(r.secondname, 1) || '.' AS full\_name

FROM public.realtor r

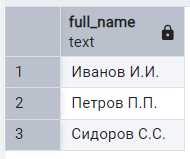
JOIN public.sale s ON r.id = s.id\_realtor

JOIN public.property p ON s.id\_property = p.id

GROUP BY r.id, r.lastname, r.firstname, r.secondname

HAVING COUNT(DISTINCT p.id\_district) > 0;

-- 0 = 2



1. Вывести названия районов, в которых средняя площадь продаваемых квартир больше 30м2.

SELECT d.name

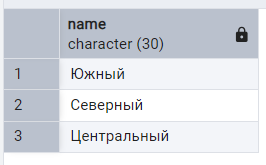
FROM public.districts d

JOIN public.property p ON d.id = p.id\_district

JOIN public.sale s ON p.id = s.id\_property

GROUP BY d.id, d.name

HAVING AVG(p.square) > 30;



1. Вывести для указанного риэлтора (ФИО) года, в которых он продал больше 2 объектов недвижимости.

SELECT EXTRACT(YEAR FROM s.date) AS year

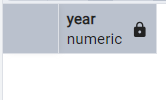
FROM public.sale s

JOIN public.realtor r ON s.id\_realtor = r.id

WHERE r.lastname = 'Иванов' AND r.firstname = 'Иван' AND r.secondname = 'Иванович'

GROUP BY EXTRACT(YEAR FROM s.date)

HAVING COUNT(s.id) > 2;



1. Вывести ФИО риэлторов, которые заработали премию в текущем месяце больше 40000 рублей. Премия рассчитываются по формуле: общая стоимость всех проданных квартир \* 15%.

SELECT r.lastname || ' ' || LEFT(r.firstname, 1) || '.' || LEFT(r.secondname, 1) || '.' AS full\_name

FROM public.realtor r

JOIN public.sale s ON r.id = s.id\_realtor

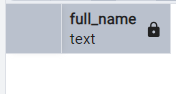
JOIN public.property p ON s.id\_property = p.id

WHERE EXTRACT(MONTH FROM s.date) = EXTRACT(MONTH FROM CURRENT\_DATE)

AND EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY r.id, r.lastname, r.firstname, r.secondname

HAVING SUM(s.cost) \* 0.15 > 40000;



1. Вывести количество однокомнатных и двухкомнатных квартир в указанном районе.

SELECT

CASE

WHEN p.rooms = 1 THEN 'Однокомнатных квартир'

WHEN p.rooms = 2 THEN 'Двухкомнатных квартир'

ELSE 'Больше двух комнат'

END AS apartment\_type,

COUNT(p.id) AS property\_count

FROM public.property p

WHERE p.id\_district = (SELECT id FROM public.districts WHERE name = 'Южный')

GROUP BY

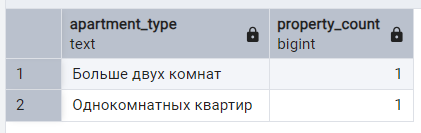
CASE

WHEN p.rooms = 1 THEN 'Однокомнатных квартир'

WHEN p.rooms = 2 THEN 'Двухкомнатных квартир'

ELSE 'Больше двух комнат'

END;



1. Определить среднюю оценку по каждому критерию для указанного объекта недвижимости. Вывести среднюю оценку и эквивалентный текст согласно таблице.

SELECT

ec.name AS criterion,

AVG(e.evaluate) AS average\_score,

CASE

WHEN AVG(e.evaluate) >= 4.5 THEN 'превосходно'

WHEN AVG(e.evaluate) >= 4.0 THEN 'очень хорошо'

WHEN AVG(e.evaluate) >= 3.5 THEN 'хорошо'

WHEN AVG(e.evaluate) >= 3.0 THEN 'удовлетворительно'

ELSE 'неудовлетворительно'

END AS evaluation\_text

FROM public.evaluation e

JOIN public.evaluation\_criteria ec ON e.id\_criteria = ec.id

WHERE e.id\_property = (SELECT id FROM public.property WHERE address = 'ул. Пушкина, 5')

GROUP BY ec.name;



1. Добавить новую таблицу «Структура объекта недвижимости» с колонками: Объект недвижимости, Тип комнаты, Площадь. Установите ограничение-проверку площади, которая должна быть больше нуля и типа комнаты (1, 2, 3, 4), где 1 – кухня, 2 – зал, 3 – спальня, 4 – санузел.

CREATE TABLE public.property\_structure

(

id\_property bigint,

room\_type smallint CHECK (room\_type IN (1, 2, 3, 4)),

area double precision CHECK (area > 0),

PRIMARY KEY (id\_property, room\_type),

CONSTRAINT fk\_property FOREIGN KEY (id\_property) REFERENCES public.property (id)

);

1. Вывести информацию о комнатах для объекта недвижимости.

SELECT

p.address,

ps.room\_type,

CASE

WHEN ps.room\_type = 1 THEN 'Кухня'

WHEN ps.room\_type = 2 THEN 'Зал'

WHEN ps.room\_type = 3 THEN 'Спальня'

WHEN ps.room\_type = 4 THEN 'Санузел'

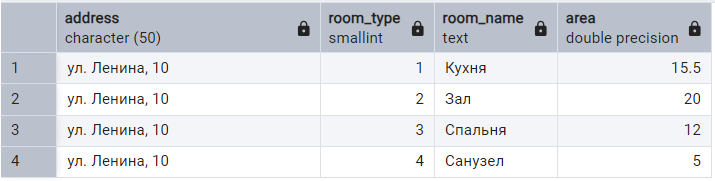
END AS room\_name,

ps.area

FROM public.property\_structure ps

JOIN public.property p ON ps.id\_property = p.id

WHERE ps.id\_property = 1;



1. Рассчитайте какой процент составляет площадь каждого типа комнаты объекта недвижимости от общей площади.

SELECT

p.address,

ps.room\_type,

CASE

WHEN ps.room\_type = 1 THEN 'Кухня'

WHEN ps.room\_type = 2 THEN 'Зал'

WHEN ps.room\_type = 3 THEN 'Спальня'

WHEN ps.room\_type = 4 THEN 'Санузел'

END AS room\_name,

ps.area,

(ps.area / total\_areas.total\_area) \* 100 AS percentage\_of\_total

FROM public.property\_structure ps

JOIN public.property p ON ps.id\_property = p.id

JOIN (

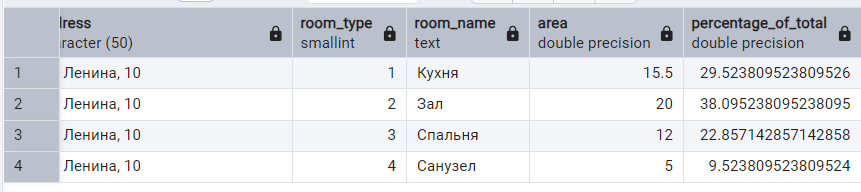
SELECT id\_property, SUM(area) AS total\_area

FROM public.property\_structure

GROUP BY id\_property

) AS total\_areas ON ps.id\_property = total\_areas.id\_property

WHERE ps.id\_property = 1;



1. Вывести количество объектов недвижимости по каждому району, общая площадь которых больше 40 м2. Использовать таблицу «Структура объекта недвижимости».

SELECT

d.name AS district\_name,

COUNT(p.id) AS property\_count

FROM

public.property p

JOIN

public.districts d ON p.id\_district = d.id

JOIN

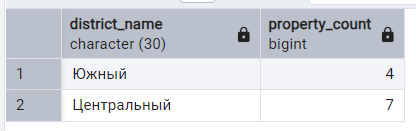
public.property\_structure ps ON p.id = ps.id\_property

GROUP BY

d.name

HAVING

SUM(ps.area) > 40;



1. Вывести квартиры, которые были проданы не позже 4 месяцев после размещения объявления о их продаже.

SELECT

p.address,

p.announcement\_date,

s.date AS sale\_date,

EXTRACT(MONTH FROM (s.date - p.announcement\_date)) AS months\_diff

FROM

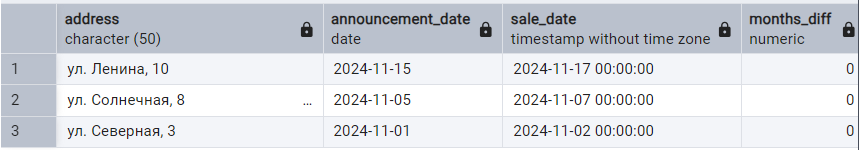
public.property p

JOIN

public.sale s ON p.id = s.id\_property

WHERE

s.date <= p.announcement\_date + INTERVAL '4 months';



1. Вывести адреса и статус объектов недвижимости, стоимость 1м2 которых меньше средней всех объектов недвижимости по району, объявления о которых были размещены не более 4 месяцев назад.

SELECT

p.address,

CASE

WHEN p.status = 1 THEN 'в продаже'

WHEN p.status = 0 THEN 'продано'

END AS status

FROM

public.property p

JOIN

public.districts d ON p.id\_district = d.id

WHERE

(p.cost / p.square) < (

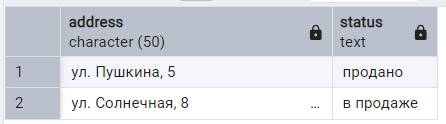
SELECT AVG(cost / square)

FROM public.property

WHERE id\_district = p.id\_district

)

AND p.announcement\_date >= CURRENT\_DATE - INTERVAL '4 months';



1. Вывести информацию о количество продаж в предыдущем и текущем годах по каждому району, а также процент изменения.

SELECT

d.name AS district\_name,

COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) THEN 1 ELSE 0 END), 0) AS current\_year\_sales,

COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) - 1 THEN 1 ELSE 0 END), 0) AS previous\_year\_sales,

CASE

WHEN COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) - 1 THEN 1 ELSE 0 END), 0) = 0 THEN NULL

ELSE ROUND(

((COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) THEN 1 ELSE 0 END), 0) -

COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) - 1 THEN 1 ELSE 0 END), 0)) \* 100.0) /

COALESCE(SUM(CASE WHEN EXTRACT(YEAR FROM s.date) = EXTRACT(YEAR FROM CURRENT\_DATE) - 1 THEN 1 ELSE 0 END), 0), 2)

END AS percentage\_change

FROM

public.sale s

JOIN

public.property p ON s.id\_property = p.id

JOIN

public.districts d ON p.id\_district = d.id

GROUP BY

d.name

ORDER BY

d.name;



1. Сформировать статистику по продажам за указанный год. Входные параметры: год.

WITH sales\_stats AS (

SELECT

p.type AS property\_type,

COUNT(s.id) AS sales\_count,

SUM(s.cost) AS total\_cost

FROM

public.sale s

JOIN

public.property p ON s.id\_property = p.id

WHERE

EXTRACT(YEAR FROM s.date) = 2024

GROUP BY

p.type

)

SELECT

CASE

WHEN property\_type = 1 THEN 'Квартиры'

WHEN property\_type = 2 THEN 'Дома'

WHEN property\_type = 3 THEN 'Апартаменты'

ELSE 'Другие'

END AS property\_type,

sales\_count,

ROUND((sales\_count \* 100.0 / (SELECT COUNT(\*) FROM public.sale s WHERE EXTRACT(YEAR FROM s.date) = 2024)), 2) AS percentage\_of\_total\_sales,

total\_cost

FROM

sales\_stats

ORDER BY

property\_type;

