



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт перспективных технологий и индустриального программирования
(ИПТИП)

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ
по дисциплине
«Базы данных и анализ промышленных данных»
Практическая работа №5

Выполнил студент группы ЭФМО-02-23

Мурадов Н.Н.

Москва 2023

СОДЕРЖАНИЕ

Задача.....	3
Решение.....	7
Результаты	17

Задача

1. Создание

таблиц Таблица

«Районы»

Код района	bigint, primary key
Название района	character

Таблица «Объект недвижимости»

Код объекта	bigint, primary key
Район	bigint, внешний ключ (Районы)
Адрес	character
Этаж	bigint
Количество комнат	bigint
Тип	bigint, внешний ключ (Тип)
Статус (1 – в продаже, 0 – продана)	bigint
Стоимость	double precision
Описание объекта	text
Площадь	double precision
Дата объявления	timestamp

Таблица «Критерии оценки»

Код критерия	bigint, primary key
Название критерия	character

Таблица «Оценки»

Код оценки	bigint, primary key
Код объекта	bigint, внешний ключ (Объекты недвижимости)
Дата оценивания	timestamp
Код критерия	bigint, внешний ключ (Критерии оценки)

Оценка	double precision
--------	------------------

Таблица «Риэлтор»

Код риэлтора	bigint, primary key
Фамилия	character
Имя	character
Отчество	character
Контактный телефон	character

Таблица «Продажа»

Код продажи	bigint, primary key
Код объекта	bigint, внешний ключ (Объекты недвижимости)
Дата продажи	timestamp
Код риэлтора	bigint, внешний ключ (Риэлтор)
Стоимость	double precision

2. Выполнение запроса

1. Вывести адреса объектов недвижимости, у которых стоимость 1 м² меньше средней стоимости по району.
2. Вывести название районов, в которых количество проданных квартир больше 5.
3. Вывести адреса квартир и название района, средняя оценка которых выше 3,5 баллов.
4. Определить годы, в которых было размещено от 2 до 3 объектов недвижимости.
5. Определить ФИО риэлторов, которые ничего не продали в текущем году.
6. Определить ФИО риэлторов, продавших квартиры, более чем в двух

районах. Предусмотреть вывод ФИО в следующем формате: Иванов И.И.

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

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

9. Вывести ФИО риэлторов, которые заработали премию в текущем месяце больше 40000 рублей. Премия рассчитывается по формуле:

*общая стоимость всех проданных квартир * 15%*

10. Вывести количество однокомнатных и двухкомнатных квартир в указанном районе. Формат вывода:

Вид квартиры	Количество объектов недвижимости
Однокомнатных квартир	2
Двухкомнатных квартир	5
Больше двух комнат	3

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

Диапазон	Эквивалентный текст
от 90% до 100%	превосходно
от 80% до 90%	очень хорошо
от 70% до 80%	хорошо
от 60% до 70%	удовлетворительно
до 60 %	неудовлетворительно

Образец вывода результатов:

Критерий	Средняя оценка	Текст
Экология	5 из 5	превосходно
Чистота	4 из 5	очень хорошо
Соседи	3,5 из 5	хорошо

12. 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.

13. Вывести информацию о комнатах для объекта недвижимости.

Тип комнаты	Площадь
Зал	20
Кухня	5

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

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

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

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

Адрес	Статус
ул. Академика Янгеля 40	продано
Ул. Тверская 20	в продаже

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

Название района	2023	2022	Разница в %
Бутово	20	40	100
Чертаново	30	20	-33,33

19. Сформировать статистику по продажам за указанный год.

Входные параметры: год.

Тип объекта недвижимости	Количество продаж	Процент от общего количества все проданных ОН	Общая сумма
Квартир	5	16%	15000000
Домов	10	34%	18500000
Апартаменты	15	50%	20000000

Решение

Листинг кода:

Создание БД:

1:

```
CREATE DATABASE realty;
```

```
\c realty;
```

```
CREATE TABLE AREA (  
    ID BIGINT PRIMARY KEY,  
    NAME CHARACTER(30)  
);
```

```
INSERT INTO AREA VALUES
```

```
(0, 'Центр'),
```

```
(1, 'Таганский'),
```

```
(2, 'Тушино')
```

```
RETURNING ID;
```

2:

```
CREATE TABLE REALTY_OBJECT (  
    ID BIGINT PRIMARY KEY,  
    AREA_ID BIGINT REFERENCES AREA (ID),  
    ADDRESS CHARACTER(30),  
    FLOOR BIGINT,  
    COUNT_ROOMS BIGINT,  
    TYPE_OBJECT BIGINT,  
    STATE_OBJECT BIGINT,  
    PRICE double precision,  
    DESCRIPTION TEXT,  
    THE_AREA double precision,  
    DATE_OF_ANNOUNCEMENT timestamp  
);
```

```
INSERT INTO REALTY_OBJECT VALUES
```

```
(3, 0, 'г. Москва, ул. Донская, д. 8 стр. 1', 1, 5, 0, 0, 12000000, 'Уютная студия с  
открытой планировкой, минималистичным дизайном и современными удобствами.  
Имеется небольшая кухня, спальная зона и ванная комната. Отличный вариант для одного  
или двух человек.', 25, '2016-07-22'),
```

```
(4, 1, 'г. Москва, ул. Тверская, д. 1', 2, 6, 1, 1, 17000000, 'Просторная квартира с  
двумя отдельными комнатами, гостиная, полностью оборудованная кухня и ванная  
комната. Имеется балкон с видом на город. Отличный вариант для семьи или группы  
друзей.', 50, '2019-01-22'),
```

```
(5, 2, 'г. Москва, ул. Снайперская д. 4', 15, 10, 0, 1, 20000000, 'Роскошный пентхаус  
с просторными и роскошными интерьерами. Три спальни, просторная гостиная,  
полностью оборудованная кухня, несколько ванных комнат и большая терраса с  
потрясающим видом на город. Идеальный выбор для проживания в стиле VIP.', 100, '2021-  
12-22')
```

```
RETURNING ID;
```

```

3:
CREATE TABLE ASSESSMENT_CRITERION (
    ID BIGINT PRIMARY KEY,
    NAME CHARACTER(30)
);

INSERT INTO ASSESSMENT_CRITERION VALUES
(6, 'Площадь'),
(7, 'Цена'),
(8, 'Дата')
RETURNING ID;

4:
CREATE TABLE SCORE (
    ID BIGINT PRIMARY KEY,
    REALTY_OBJECT_ID BIGINT REFERENCES REALTY_OBJECT (ID),
    DATE_OF_SCORE timestamp,
    ASSESSMENT_CRITERION_ID BIGINT REFERENCES
ASSESSMENT_CRITERION (ID),
    VALUE double precision
);

INSERT INTO SCORE VALUES
(9, 3, '2016-07-22', 6, 10),
(10, 4, '2019-01-22', 8, 7),
(11, 5, '2021-12-22', 7, 2)
RETURNING ID;

5:
CREATE TABLE REALTOR (
    ID BIGINT PRIMARY KEY,
    LAST_NAME CHARACTER(15),
    FIRST_NAME CHARACTER(15),
    PATRONYMIC CHARACTER(15),
    PHONE CHARACTER(15)
);

INSERT INTO REALTOR VALUES
(12, 'Дорохов', 'Владимир', 'Александрович', '+7 (495) 727 77 77'),
(13, 'Зубенко', 'Михаил', 'Петрович', '+7 (926) 333 31 10'),
(14, 'Мякина', 'Наталья', 'Валерьевна', '+7 (999) 878 78 78')
RETURNING ID;

6:
CREATE TABLE SALE (
    ID BIGINT PRIMARY KEY,
    REALTY_OBJECT_ID BIGINT REFERENCES REALTY_OBJECT (ID),
    DATE_OF_SALE timestamp,
    REALTOR_ID BIGINT REFERENCES REALTOR (ID),
    PRICE double precision
);

```



```
INSERT INTO SALE VALUES
(15, 3, '2016-07-23', 12, 11999999),
(16, 4, '2019-01-23', 13, 16999999),
(17, 5, '2021-12-23', 14, 19999999)
RETURNING ID;
```

Поправки:

```
ALTER TABLE REALTY_OBJECT ALTER COLUMN ADDRESS TYPE
CHARACTER(40);
```

```
ALTER TABLE REALTOR ALTER COLUMN PHONE TYPE CHARACTER(20);
```

```
ALTER TABLE REALTY_OBJECT ALTER COLUMN ADDRESS TYPE
CHARACTER(80);
```

Тест:

```
INSERT INTO AREA
SELECT (i + 17), (faker.street_title() || ' sdf') from generate_series(1,5) i
RETURNING *;
```

```
delete FROM AREA WHERE ID>15;
```

Запросы:

1:

Дополнительная генерация данных:

```
SELECT faker.faker('ru_RU');
```

```
INSERT INTO REALTY_OBJECT
SELECT (i + 22), 1, 'г. Москва, ' || faker.address(), rand(1, 25), rand(1, 10), rand(),
0, rand(100000000, 200000000), faker.text(), rand(10, 150),
faker.date_between('-50y')::timestamp from generate_series(1,1) i
RETURNING *;
```

```
delete FROM REALTY_OBJECT WHERE ID>22;
```

Запрос:

```
--SELECT * FROM (SELECT a, r FROM REALTY_OBJECT r, AREA a WHERE
r.AREA_ID = a.ID) x;
```

```
WITH x as (SELECT a, r, RND((r).PRICE / (r).THE_AREA, 2) p_to_m2 FROM
REALTY_OBJECT r, AREA a WHERE r.AREA_ID = a.ID),
x1 as (SELECT a, RND(AVG((r).PRICE / (r).THE_AREA), 2) avg FROM x
GROUP BY a)
SELECT x1, (r).ADDRESS, p_to_m2 FROM x NATURAL INNER JOIN x1
WHERE p_to_m2 < x1.avg GROUP BY x1, r, p_to_m2;
```

2:

```
SELECT a, COUNT(r) FROM REALTY_OBJECT r, AREA a WHERE r.AREA_ID =  
a.ID AND (r).STATE_OBJECT = 0 GROUP BY a HAVING COUNT(r) > 5;
```

3:

Поправка:

```
SELECT * FROM SCORE;
```

```
UPDATE SCORE SET VALUE = rand(1, 5) WHERE true;
```

Дополнительная генерация данных:

```
--SELECT x.id FROM (SELECT r.id FROM REALTY_OBJECT r ORDER BY  
random() LIMIT 1) x(id);
```

```
--SELECT * FROM (SELECT r.id FROM REALTY_OBJECT r) x ORDER BY  
random() LIMIT 1;
```

```
--SELECT * FROM (VALUES (1), (2), (3)) x ORDER BY random() LIMIT 1;
```

```
SELECT faker.faker('ru_RU');
```

```
INSERT INTO SCORE SELECT  
    (i + 23), (SELECT r.id FROM REALTY_OBJECT r WHERE i=i ORDER BY  
random() LIMIT 1),  
    faker.date_between('-50y')::timestamp, rand(6, 8), rand(2, 5) FROM  
generate_series(1,100) i  
RETURNING *;
```

```
delete FROM SCORE WHERE ID>22;
```

Запрос:

```
WITH x as (SELECT s.REALTY_OBJECT_ID realId, RND(AVG(s.VALUE), 2) avg, r,  
a FROM Score s, REALTY_OBJECT r, AREA a WHERE s.REALTY_OBJECT_ID = r.ID  
AND r.AREA_ID = a.ID GROUP BY s.REALTY_OBJECT_ID, r, a)  
SELECT x.realId, x.avg, (x.a).NAME, (x.r).ADDRESS FROM x WHERE x.avg  
> 3.5;
```

4:

Дополнительная генерация данных(REALTY_OBJECT):

```
SELECT faker.faker('ru_RU');
```

```
INSERT INTO REALTY_OBJECT  
SELECT (i + 123), 1, 'г. Москва, ' || faker.address(), rand(1, 25), rand(1, 10), rand(),  
0, rand(100000000, 200000000), faker.text(), rand(10, 150),  
faker.date_between('-20y')::timestamp from generate_series(1,20) i  
RETURNING *;
```

```
delete FROM REALTY_OBJECT WHERE ID>23;
```

Запрос:

```
WITH x AS (SELECT to_char(r.DATE_OF_ANNOUNCEMENT, 'yyyy') yr FROM
REALTY_OBJECT r),
x1 as (SELECT x.yr, COUNT(*) cnt FROM x GROUP BY x.yr)
SELECT * FROM x1 WHERE cnt BETWEEN 2 AND 3;
```

5:

Поправка:

```
UPDATE SALE SET REALTY_OBJECT_ID = 18 WHERE ID = 16;
UPDATE SALE SET REALTY_OBJECT_ID = 19 WHERE ID = 17;
```

```
INSERT INTO SALE VALUES
(144, 20, '2022-07-23', 12, 11999999),
(145, 21, '2023-01-23', 12, 16999999)
RETURNING *;
```

Поправка #2:

```
INSERT INTO REALTOR VALUES
(507, 'Дорохов', 'Михаил', 'Александрович', '+7 (495) 727 77 77')
RETURNING ID;
```

```
INSERT INTO SALE VALUES
(508, 20, '2022-07-23', 507, 11999999)
RETURNING *;
```

Запрос:

```
WITH x AS (SELECT to_char(s.DATE_OF_SALE, 'yyyy')::INT yr, (r.LAST_NAME || '
' || r.FIRST_NAME || ' ' || r.PATRONYMIC) fio FROM SALE s, REALTOR r WHERE
s.REALTOR_ID = r.ID),
x1 as (SELECT x.fio, array_agg(x.yr) arrYr FROM x GROUP BY x.fio)
SELECT * FROM x1 WHERE 2023 != ALL(x1.arrYr);
```

6:

Дополнительная генерация данных:

```
SELECT faker.faker('ru_RU');
```

```
INSERT INTO REALTOR
SELECT (i + 145), faker.last_name_female(), faker.first_name_female(),
faker.middle_name_female(), faker.phone_number() from generate_series(1,7) i
RETURNING *;
```

```
INSERT INTO AREA SELECT
```

```

        (i + 152), faker.street_title() from generate_series(1,4) i
RETURNING *;

WITH idA AS (SELECT id FROM AREA a)
INSERT INTO REALTY_OBJECT SELECT
    (i + 156), (SELECT * FROM idA WHERE i=i ORDER BY random() LIMIT 1),
    'г. Москва, ' || faker.address(), rand(1, 25), rand(1, 10),rand(), 0,
    rand(100000000, 200000000), faker.text(), rand(10, 150),
    faker.date_between('-50y')::timestamp from generate_series(1,30) i
RETURNING *;

delete FROM REALTY_OBJECT WHERE ID>156;

WITH idRO AS (SELECT id FROM REALTY_OBJECT r WHERE r.STATE_OBJECT
= 0),
    idRR AS (SELECT id FROM REALTOR r)
INSERT INTO SALE SELECT
    (i + 186), (SELECT * FROM idRO WHERE i=i ORDER BY random() LIMIT 1),
    faker.date_between('-10y')::timestamp,
    (SELECT * FROM idRR WHERE i=i ORDER BY random() LIMIT 1),
rand(100000000, 200000000)
    from generate_series(1,100) i
RETURNING *;

delete FROM SALE WHERE ID>186;

Запрос:

WITH x AS (SELECT DISTINCT (rR.LAST_NAME || ' ' || LEFT(rR.FIRST_NAME, 1)
|| ' ' || LEFT(rR.PATRONYMIC, 1) || '.') fio, rO.AREA_ID idROA FROM SALE s, REALTOR
rR, REALTY_OBJECT rO WHERE s.REALTOR_ID = rR.ID AND s.REALTY_OBJECT_ID =
rO.ID),
    x1 as (SELECT x.fio, COUNT(x.idROA) cnt, array_agg(x.idROA) arrIdROA
FROM x GROUP BY x.fio)
    SELECT * FROM x1 WHERE cnt > 2;

7:

SELECT a.NAME, rnd(AVG(r.THE_AREA), 2) FROM REALTY_OBJECT r, AREA a
WHERE r.AREA_ID = a.ID AND r.STATE_OBJECT = 1 GROUP BY a.NAME;

8:

WITH x AS (SELECT (rR.LAST_NAME || ' ' || rR.FIRST_NAME || ' ' ||
rR.PATRONYMIC) fio, to_char(s.DATE_OF_SALE, 'yyyy')::INT d, s.id idS FROM SALE s,
REALTOR rR WHERE s.REALTOR_ID = rR.ID),
    x1 as (SELECT x.fio, x.d, COUNT(x.d) cnt, array_agg(x.idS) arrIdS FROM x
GROUP BY x.fio, x.d)
    SELECT * FROM x1 WHERE cnt > 2;

9:

```

Дополнительная генерация данных:

```
WITH idRO AS (SELECT id FROM REALTY_OBJECT r WHERE r.STATE_OBJECT
= 0),
      idRR AS (SELECT id FROM REALTOR r)
INSERT INTO SALE SELECT
      (i + 286), (SELECT * FROM idRO WHERE i=i ORDER BY random() LIMIT 1),
      faker.date_between('-1y')::timestamp,
      (SELECT * FROM idRR WHERE i=i ORDER BY random() LIMIT 1),
      rand(100000000, 200000000)
      from generate_series(1,100) i
RETURNING *;
```

Запрос:

```
WITH x AS (SELECT (LAST_NAME || ' ' || FIRST_NAME || ' ' || PATRONYMIC) fio,
to_char(DATE_OF_SALE, 'mm')::INT d, s.id idS, PRICE FROM SALE s, REALTOR rR
WHERE s.REALTOR_ID = rR.ID),
      x1 as (SELECT fio, d, COUNT(d) cnt, RND(SUM(PRICE) * 0.001, 1) sumR,
array_agg(idS) arrIdS FROM x GROUP BY fio, d)
SELECT * FROM x1 WHERE x1.d = 11 AND SUMR > 40000;
```

10:

Поправка:

```
UPDATE REALTY_OBJECT SET COUNT_ROOMS = rand(1, 5) WHERE TRUE;
```

Запрос:

```
--SELECT ARRAY[1, 2, 3, 4];
```

```
WITH x as (SELECT ARRAY['Однокомнатных квартир', 'Двухкомнатных квартир',
'Больше двух комнат'] des),
      x1 as (SELECT NAME, ARRAY[COUNT(COUNT_ROOMS = 1 OR NULL),
COUNT(COUNT_ROOMS = 2 OR NULL), COUNT(COUNT_ROOMS > 2 OR NULL)] cnt,
array_agg(r.id) filter(where COUNT_ROOMS = 1) arr1 FROM REALTY_OBJECT r, AREA a
WHERE AREA_ID = a.ID AND NAME = 'Таганский' GROUP BY NAME)
SELECT unnest(des) "Вид квартиры", unnest(cnt) "Количество объектов
недвижимости" FROM x, x1;
```

```
--SELECT a.NAME, r.id, r.ADDRESS FROM REALTY_OBJECT r, AREA a WHERE
r.AREA_ID = a.ID AND r.COUNT_ROOMS BETWEEN 1 AND 2;
```

```
--SELECT * FROM (VALUES ('Однокомнатных квартир'), ('Двухкомнатных
квартир'), ('Больше двух комнат')) x;
```

11:

```
WITH x as (SELECT NAME, RND(AVG(VALUE), 1) avg FROM Score s,
REALTY_OBJECT r, ASSESSMENT_CRITERION asCr WHERE REALTY_OBJECT_ID =
```

```
20 AND REALTY_OBJECT_ID = r.ID AND ASSESSMENT_CRITERION_ID = asCr.ID  
GROUP BY NAME)
```

```
SELECT NAME "Критерий", (avg || ' из 5') "Средняя оценка",  
getText11(avg/5) "Текст" FROM x;
```

12:

Новая таблица:

```
CREATE TABLE STRUCTURE_REALTY_OBJECT (  
    ID BIGINT PRIMARY KEY,  
    REALTY_OBJECT_ID BIGINT REFERENCES REALTY_OBJECT (ID),  
    TYPE BIGINT CHECK (TYPE BETWEEN 1 AND 4),  
    THE_AREA BIGINT CHECK (THE_AREA > 0)  
);
```

```
INSERT INTO STRUCTURE_REALTY_OBJECT VALUES  
    (387, 3, 1, 50),  
    (388, 4, 2, 50),  
    (389, 5, 3, 50)  
RETURNING *;
```

13:

Дополнительная генерация данных:

```
WITH idR AS (SELECT id FROM REALTY_OBJECT r)  
INSERT INTO STRUCTURE_REALTY_OBJECT SELECT  
    (i + 386), (SELECT * FROM idR WHERE i=i ORDER BY random() LIMIT 1),  
    rand(1, 4), rand(1, 50)  
from generate_series(1,100) i  
RETURNING *;
```

```
delete FROM STRUCTURE_REALTY_OBJECT WHERE ID>386;
```

Запрос:

```
--SELECT REALTY_OBJECT_ID, array_agg(TYPE) FROM  
STRUCTURE_REALTY_OBJECT s GROUP BY REALTY_OBJECT_ID;
```

```
SELECT getTypeText(TYPE) "Тип комнаты", THE_AREA "Площадь" FROM  
STRUCTURE_REALTY_OBJECT s WHERE REALTY_OBJECT_ID = 184;
```

14:

```
WITH x as (SELECT getTypeText(TYPE) typ, THE_AREA FROM  
STRUCTURE_REALTY_OBJECT s WHERE REALTY_OBJECT_ID = 184),  
x1 as (SELECT SUM(THE_AREA) sum FROM x)  
SELECT typ "Тип комнаты", THE_AREA "Площадь", rnd((THE_AREA / sum,  
2) * 100 "Процент площади" FROM x, x1;
```

15:

```

WITH x as (SELECT NAME, REALTY_OBJECT_ID, SUM(s.THE_AREA) sum
FROM STRUCTURE_REALTY_OBJECT s, REALTY_OBJECT r, AREA a WHERE
REALTY_OBJECT_ID = r.ID AND AREA_ID = a.ID GROUP BY NAME,
REALTY_OBJECT_ID)
SELECT NAME "Название района", COUNT(REALTY_OBJECT_ID) "Кол-во
объектов", array_agg(REALTY_OBJECT_ID) "Объекты" FROM x WHERE sum > 40
GROUP BY NAME;

```

16:

```

SELECT r.ID "Квартира", DATE_OF_ANNOUNCEMENT "Дата объявления",
DATE_OF_SALE "Дата продажи" FROM SALE s, REALTY_OBJECT r WHERE
REALTY_OBJECT_ID = r.ID AND getDiffSec(DATE_OF_SALE,
DATE_OF_ANNOUNCEMENT) BETWEEN 1 AND 10368000 GROUP BY r.ID,
DATE_OF_ANNOUNCEMENT, DATE_OF_SALE;

```

17:

Дополнительная генерация данных:

```

WITH idA AS (SELECT id FROM AREA a)
INSERT INTO REALTY_OBJECT SELECT
(i + 486), (SELECT * FROM idA WHERE i=i ORDER BY random() LIMIT 1),
'r. Москва, ' || faker.address(), rand(1, 25), rand(1, 10), rand(), rand(),
rand(10000000, 20000000), faker.text(), rand(10, 150),
faker.date_between('-4M')::timestamp from generate_series(1,20) i
RETURNING *;

```

```

delete FROM REALTY_OBJECT WHERE ID>486;

```

Запрос:

```

--SELECT r.ID "Квартира", DATE_OF_ANNOUNCEMENT "Дата объявления"
FROM REALTY_OBJECT r WHERE getDiffSec(now()::timestamp,
DATE_OF_ANNOUNCEMENT) BETWEEN 1 AND 10368000 GROUP BY r.ID,
DATE_OF_ANNOUNCEMENT;

```

```

WITH x as (SELECT a, r, RND(PRICE / THE_AREA, 2) m2 FROM
REALTY_OBJECT r, AREA a WHERE r.AREA_ID = a.ID),
x1 as (SELECT a, RND(AVG((r).PRICE / (r).THE_AREA), 2) avg FROM x
GROUP BY a)
SELECT avg "Средняя по району", (r).ID, (r).ADDRESS "Адрес", m2,
getTypeTextSt((r).STATE_OBJECT) "Статус" FROM x NATURAL INNER JOIN x1 WHERE
m2 < avg AND getDiffSec(now()::timestamp, (r).DATE_OF_ANNOUNCEMENT) BETWEEN
1 AND 10368000 GROUP BY avg, r, m2;

```

18:

```

--SELECT NAME FROM SALE s, REALTY_OBJECT r, AREA a WHERE
to_char(DATE_OF_SALE, 'yyyy')::INT > 2021 AND REALTY_OBJECT_ID = r.ID AND
AREA_ID = a.ID GROUP BY NAME;

```

```

WITH x as (SELECT NAME, to_char(DATE_OF_SALE, 'yyyy')::INT yr FROM SALE
s, REALTY_OBJECT r, AREA a WHERE REALTY_OBJECT_ID = r.ID AND AREA_ID =
a.ID GROUP BY NAME, DATE_OF_SALE),
x1 as (SELECT NAME, COUNT(yr = 2022 OR NULL)::FLOAT cnt22,
COUNT(yr = 2023 OR NULL) cnt23 FROM x GROUP BY NAME)
SELECT NAME "Название района", cnt23 "2023", cnt22 "2022", rnd((cnt22-
cnt23)/cnt23, 4) * 100 "Разница в %" FROM x1;

```

19:

Поправка:

```
UPDATE REALTY_OBJECT SET TYPE_OBJECT = rand(0, 2) WHERE TRUE;
```

Запрос:

```

WITH x as (SELECT TYPE_OBJECT, s.PRICE, to_char(DATE_OF_SALE,
'yyyy')::INT yr FROM SALE s, REALTY_OBJECT r WHERE REALTY_OBJECT_ID = r.ID
GROUP BY TYPE_OBJECT, s.PRICE, DATE_OF_SALE),
x1 as (SELECT * FROM x WHERE yr = 2023),
x2 as (SELECT COUNT(*)::float FROM x1),
x3 as (SELECT TYPE_OBJECT FROM x1 GROUP BY TYPE_OBJECT),
x4 as (SELECT TYPE_OBJECT, COUNT(*) cnt FROM x1 GROUP BY
TYPE_OBJECT),
x5 as (SELECT TYPE_OBJECT, SUM(PRICE) sum FROM x WHERE yr =
2023 GROUP BY TYPE_OBJECT)
SELECT getTypeTextRO(TYPE_OBJECT) "Тип объекта недвижимости", cnt
"Количество продаж", (rnd(cnt/count * 100, 0) || '%') "% от общего колва все проданных
ОН", x5.sum "Общая сумма" FROM x2, x4 NATURAL INNER JOIN x5;

```


Результаты

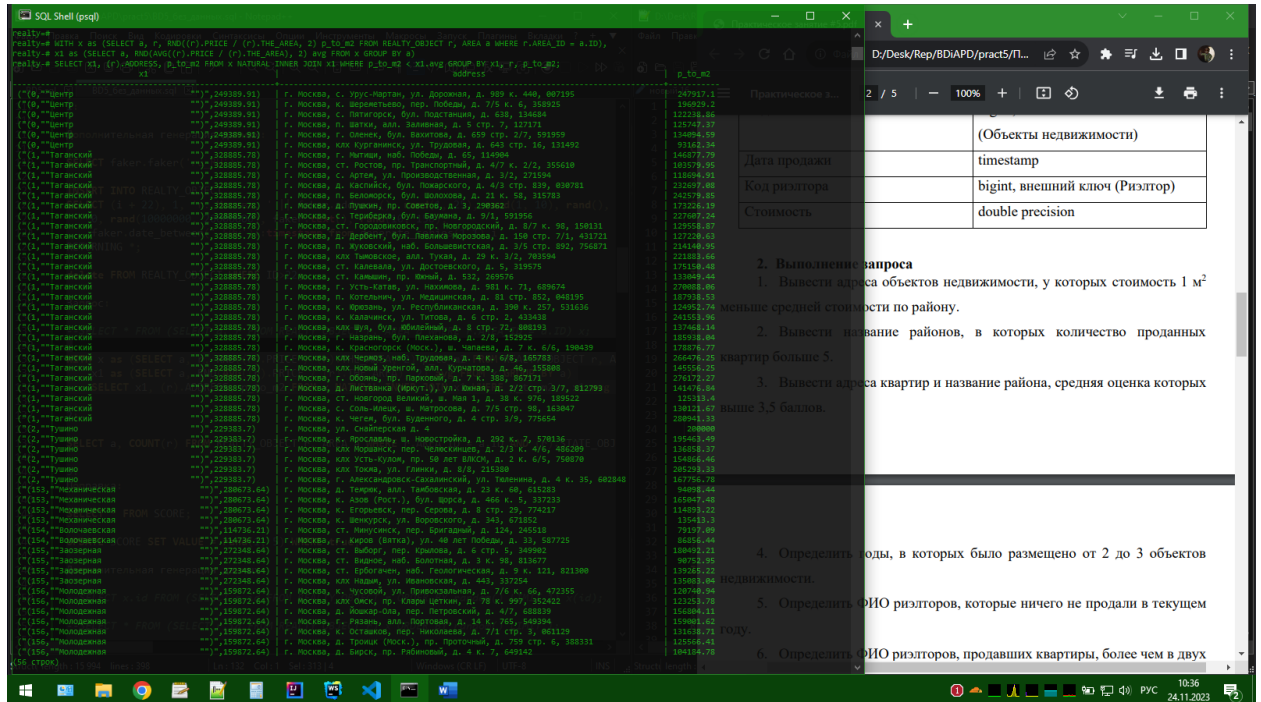


Рисунок 1 – Результат к 1 запросу

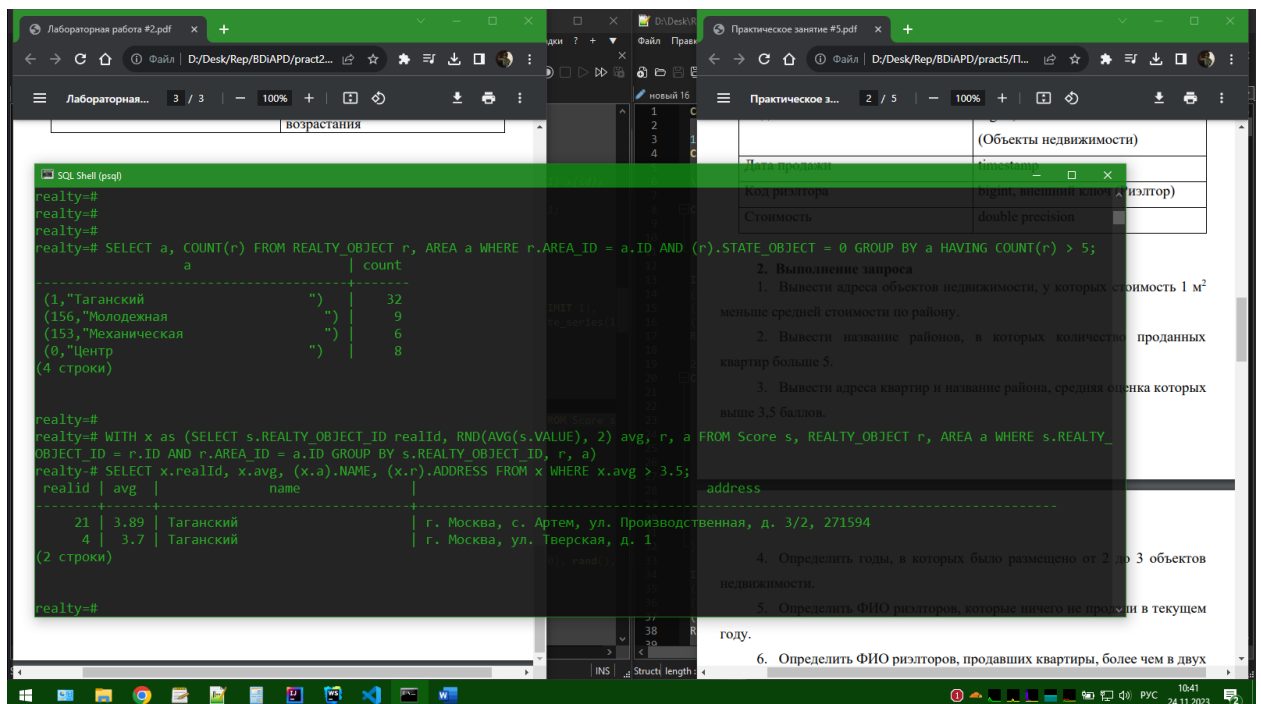


Рисунок 2 – Результат к 2 и 3 запросу

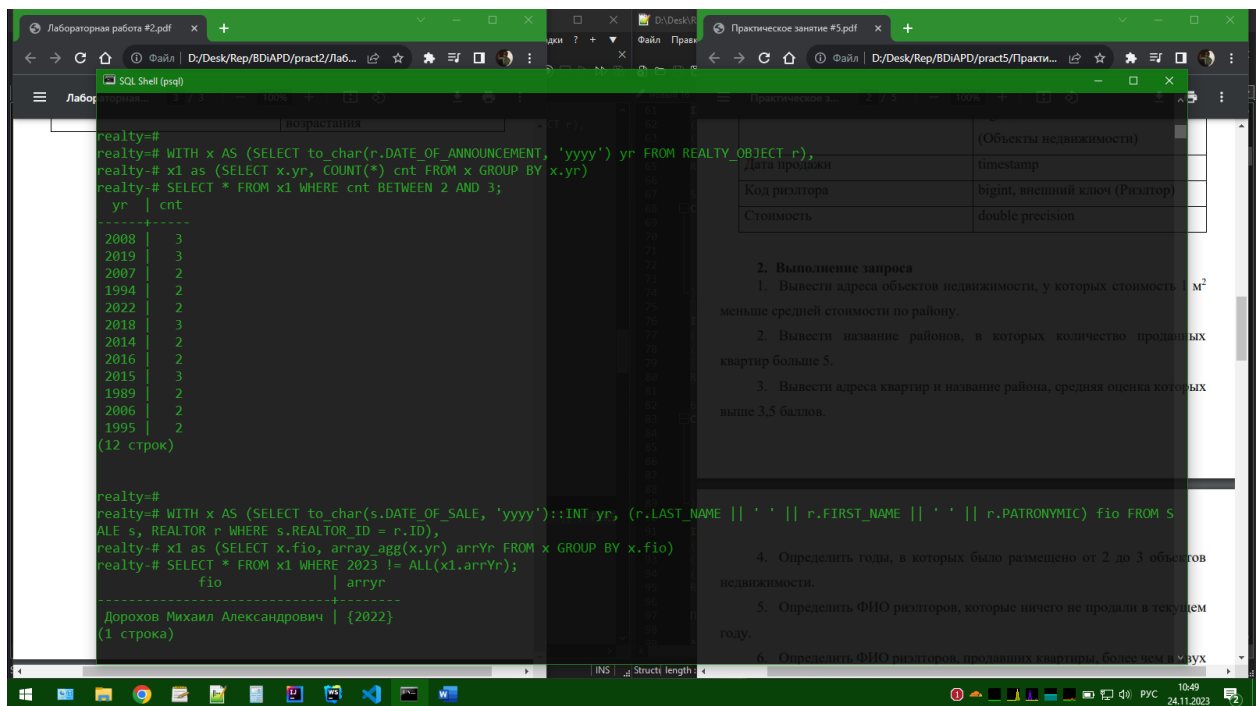


Рисунок 3 – Результат к 4 и 5 запросу

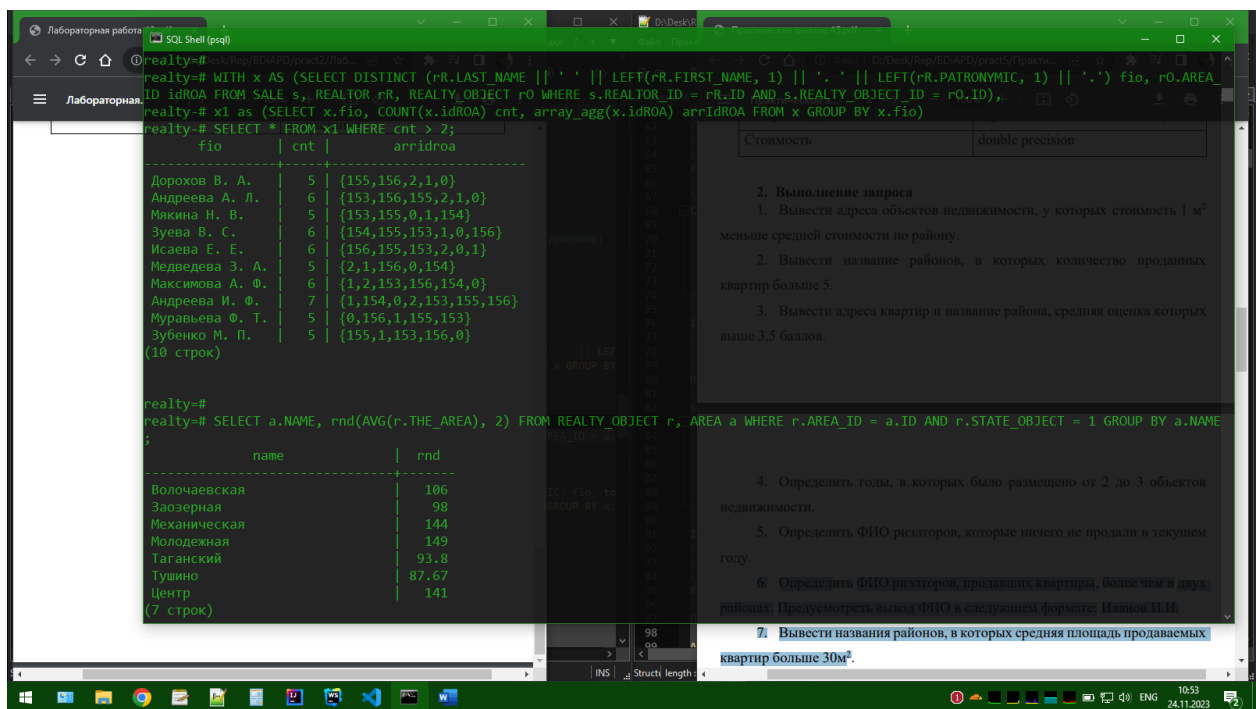


Рисунок 4 – Результат к 6 и 7 запросу

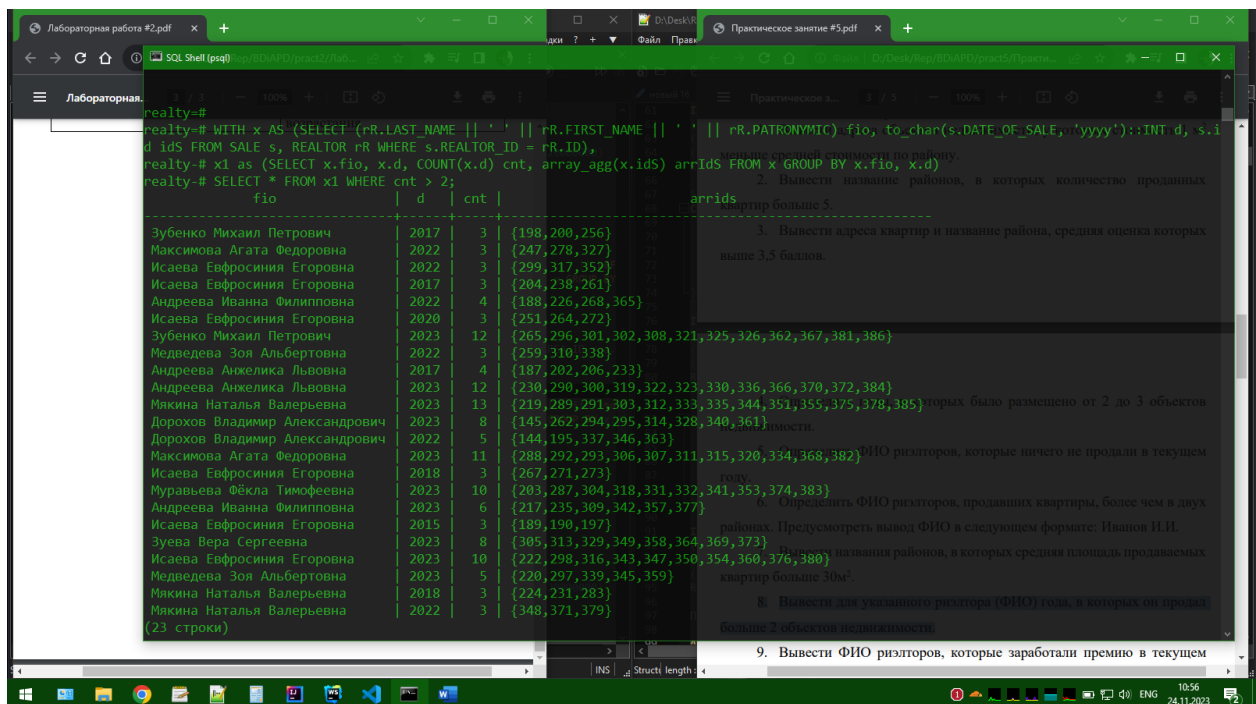


Рисунок 5 – Результат к 8 запросу

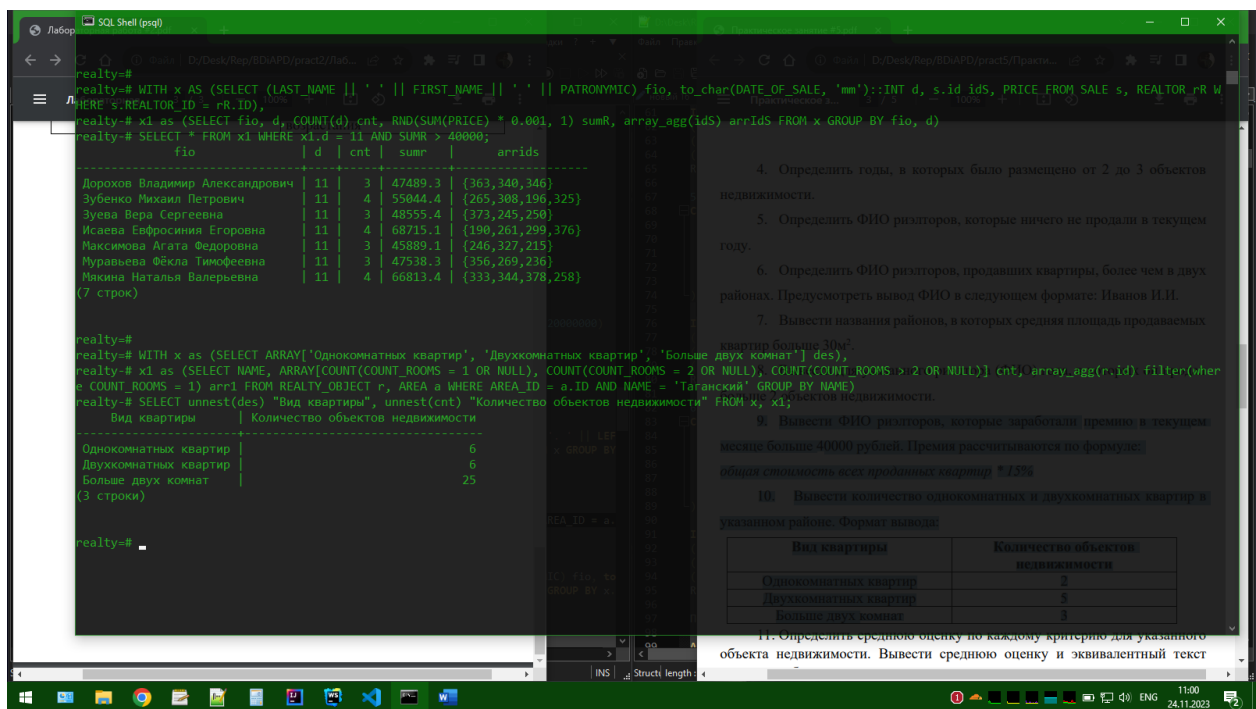


Рисунок 6 – Результат к 9 и 10 запросу

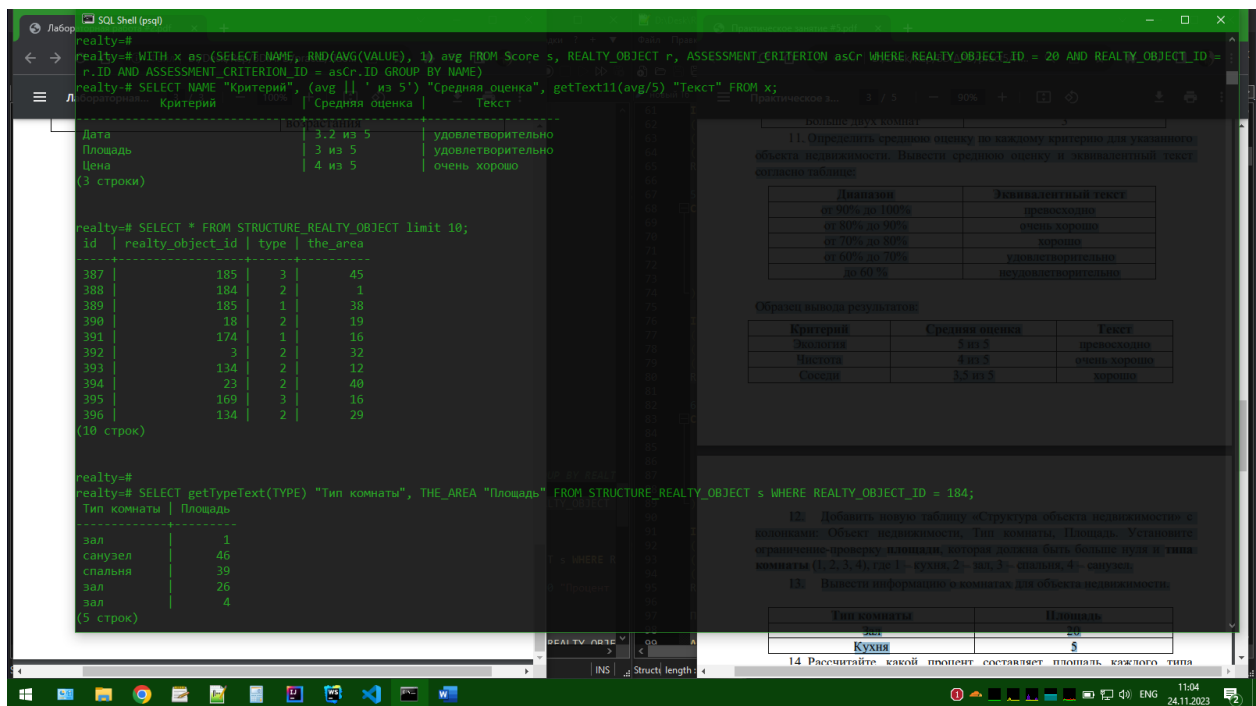


Рисунок 7 – Результаты к 11, 12 и 13 запросам

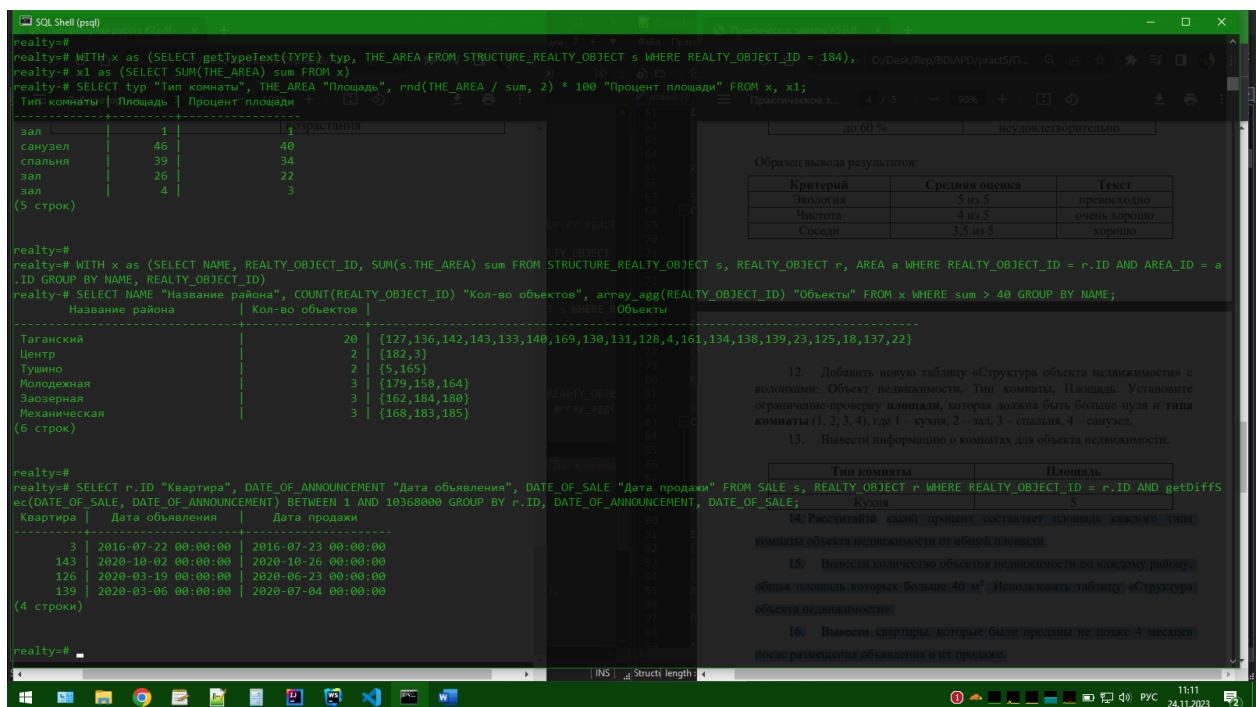


Рисунок 8 – Результаты к 14, 15 и 16 запросам

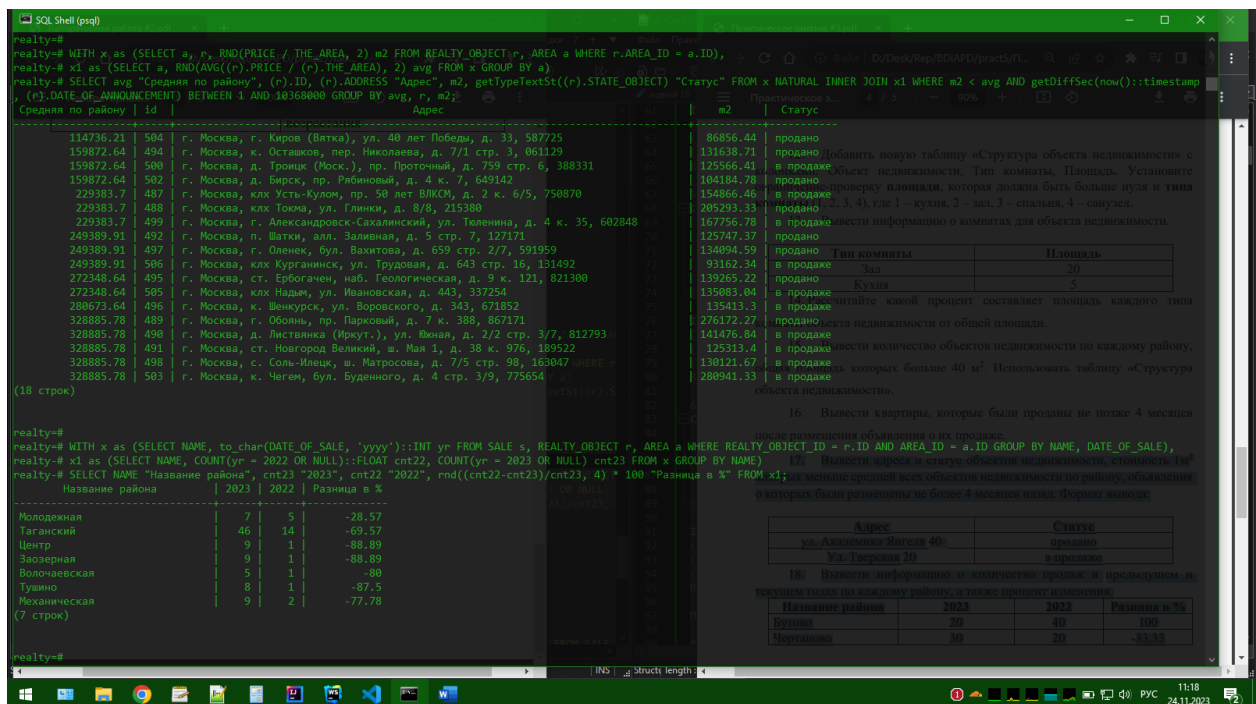


Рисунок 9 – Результат к 17 и 18 запросу

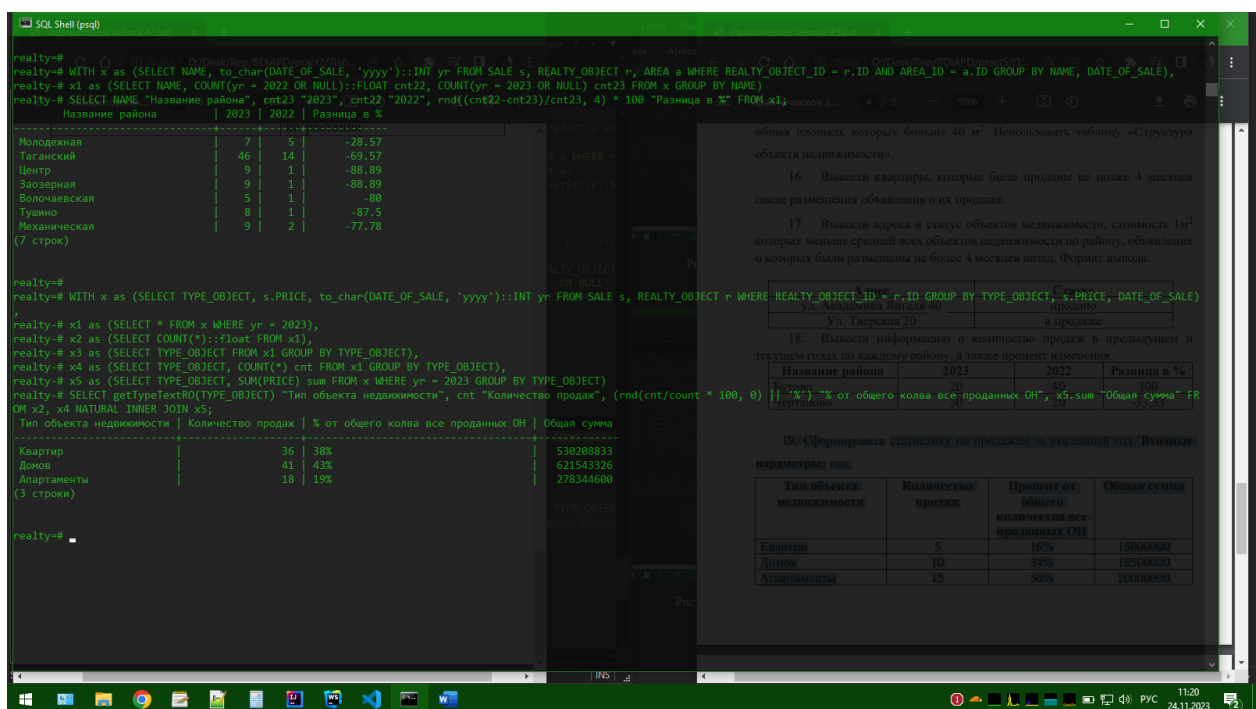


Рисунок 10 – Результат к 19 запросу