



**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 №6**

Выполнил студент группы ЭФМО-02-23

Мурадов Н.Н.

Москва 2023

## СОДЕРЖАНИЕ

Задача.....	3
Решение.....	4
Результаты .....	9

## Задача

1. Конвертирует стоимость объекта недвижимости в евро и долларах.

Вх. параметр	Вых. параметр
курс, код объекта недвижимости	стоимость в евро

2. Рассчитывает заработную плату риэлтора по формуле:  $N \cdot S + R$ , где  $N$  – общая сумма проданных объектов недвижимости в месяц (подсчет осуществляется автоматически по данным таблицы «Продажи» с использованием агрегатной функции),  $S$  – коэффициент,  $R$  – премия.

Вх. параметр	Вых. параметр
$S$ , $R$ , начальная и конечная дата, фамилия риэлтора	размер заработной платы

3. Добавить таблицу «Заработная плата риэлтора», содержащая сведения: год, месяц, размер выплаты. Изменить тело функции из пункта 2 таким образом, чтобы рассчитанная заработная плата сохранялась в этой таблице.

4. Рассчитывает процент изменения продажной стоимости объекта недвижимости от первоначально заявленной и срок продажи (в месяцах)

5. Изменить функцию, созданную в пункте 5 таким образом, чтобы в зависимости от срока продажи выводилось сообщение

Срок	Сообщение
менее 3 месяцев	Очень быстро
от 3 месяцев до 6 месяцев	Быстро
от 6 месяцев до 1 года	Долго
от 1 года	Очень долго

6. Формирует список средних оценок по каждому критерию для объектов недвижимости.

7. Написать функцию, которая рассчитывает сумму ежемесячного платежа по ипотеке. **Входные параметры:** код объекта недвижимости, процентная ставка, срок, первоначальный взнос.

8. Написать функцию, которая рассчитывает сумму налога на недвижимость.

9. Добавить таблицу «Динамика цен», где будет храниться изменения стоимости. Таблица будет содержать следующие колонки: код объекта недвижимости, новая стоимость, дата изменения.

Написать функцию, которая будет возвращать следующий результат:

Дата	Новая стоимость	Изменение	% Изменения
25.03.2022	1000000	0	0
28.03.2022	1500000	500000	+50
01.04.2022	1300000	-300000	-13,33

% изменения рассчитывается по формуле:

$((V2 - V1) / V1) \times 100$ . Где  $V1$  – старое значение, а  $V2$  – новое

## Решение

Листинг кода:

```
1:

CREATE OR REPLACE FUNCTION getExchangesRate(bigint, float, float)
RETURNS TABLE("В рублях" BIGINT, "В долларах" FLOAT, "В евро" FLOAT) AS
$$
    SELECT PRICE, rnd(price/$2, 2), rnd(price/$3, 2) FROM REALTY_OBJECT r
WHERE ID = $1;
$$ language SQL;

DROP FUNCTION getexchangesrate(bigint,double precision,double precision);

SELECT * FROM getExchangesRate(124, 88.81, 96.92);

2:

CREATE OR REPLACE FUNCTION getSalary(S float, R bigint, f text, beg
TIMESTAMP, ends TIMESTAMP)
RETURNS TABLE(idrr BIGINT, last_name character(15), y integer, m integer, cnt
BIGINT
, arrids BIGINT[], arridr BIGINT[], sumobj text, "Сумма заработной платы" text) AS
$$
BEGIN
    DROP TABLE IF EXISTS x, x1;
    create temporary table x as SELECT rR.ID idRR, rR.LAST_NAME,
to_char(DATE_OF_SALE, 'yyyy')::INT y, to_char(DATE_OF_SALE, 'mm')::INT m, s.id idS,
s.REALTY_OBJECT_ID idR FROM SALE s, REALTOR rR WHERE REALTOR_ID = rR.ID
AND rR.LAST_NAME = f AND DATE_OF_SALE BETWEEN beg AND ends;
    create temporary table x1 as SELECT x.idRR, x.LAST_NAME, x.y, x.m,
COUNT(x.m), array_agg(idS) arrids, array_agg(idR), to_char(SUM(PRICE), '999 999 999')
sumObj FROM x, REALTY_OBJECT r WHERE idR = r.ID GROUP BY x.idRR,
x.LAST_NAME, x.y, x.m;
    return query(SELECT x1.*, to_char(replace(x1.sumObj, ' ', ' '::INT*S+R, '999
999.99') FROM x1);
END $$ language plpgsql;

SELECT * FROM getSalary(0.002, 40000, 'Мякина', '2017-05-07', '2023-11-24');

DROP FUNCTION getSalary(S float, R bigint, f text, beg TIMESTAMP, ends
TIMESTAMP);

3:

CREATE TABLE REALTORS_SALARY (
    ID BIGINT PRIMARY KEY,
    REALTOR_ID BIGINT REFERENCES REALTOR (ID),
    YEAR BIGINT,
    MONTH BIGINT,
    VALUE FLOAT
);
```

```

CREATE OR REPLACE FUNCTION getSalary3(S float, R bigint, f text, beg
TIMESTAMP, ends TIMESTAMP)
RETURNS TABLE(ID BIGINT, REALTOR_ID BIGINT, YEAR BIGINT, MONTH
BIGINT, VALUE FLOAT) AS $$
BEGIN
    DROP TABLE IF EXISTS x, x1;
    create temporary table x as SELECT rR.ID idRR, rR.LAST_NAME,
to_char DATE_OF_SALE, 'yyyy')::INT y, to_char DATE_OF_SALE, 'mm')::INT m, s.id idS,
s.REALTY_OBJECT_ID idR FROM SALE s, REALTOR rR WHERE s.REALTOR_ID =
rR.ID AND rR.LAST_NAME = f AND DATE_OF_SALE BETWEEN beg AND ends;
    create temporary table x1 as SELECT x.idRR, x.LAST_NAME, x.y, x.m,
COUNT(x.m), array_agg(idS) arrids, array_agg(idR), to_char(SUM(PRICE), '999 999 999')
sumObj FROM x, REALTY_OBJECT r WHERE idR = r.ID GROUP BY x.idRR,
x.LAST_NAME, x.y, x.m;
    INSERT INTO REALTORS_SALARY SELECT
        ROW_NUMBER() over() + 508, idRR, y, m, rnd(replace(sumObj, ' ',
)::INT*S+R, 2)
        from x1;
    RETURN query(SELECT * FROM REALTORS_SALARY);
END $$ language plpgsql;

SELECT * FROM getSalary3(0.002, 40000, 'Мякина', '2017-05-07', '2023-11-24');

DROP FUNCTION getSalary3(S float, R bigint, f text, beg TIMESTAMP, ends
TIMESTAMP);

DROP TABLE REALTORS_SALARY;

delete FROM REALTORS_SALARY WHERE ID>508;

4:

CREATE OR REPLACE FUNCTION getInfo6_4()
RETURNS TABLE(sP double precision, rP double precision, dR TIMESTAMP, dS
TIMESTAMP, "%" FLOAT, getDiffMonth NUMERIC) AS $$
BEGIN
    DROP TABLE IF EXISTS x;
    create temporary table x as SELECT s.PRICE sP, r.PRICE rP,
DATE_OF_ANNOUNCEMENT dR, DATE_OF_SALE dS FROM SALE s,
REALTY_OBJECT r WHERE REALTY_OBJECT_ID = r.ID;
    RETURN query(SELECT x.*, rnd((x.sP-x.rP)/x.rP * 100, 2), getDiffMonth(x.dS,
x.dR) FROM x);
END $$ language plpgsql;

SELECT * FROM getInfo6_4();

DROP FUNCTION getInfo6_4();

5:

CREATE OR REPLACE FUNCTION getInfo6_5()

```

```

    RETURNS TABLE(sP double precision, rP double precision, dR TIMESTAMP, dS
TIMESTAMP, "%" FLOAT, getDiffMonth NUMERIC, "Срок" text) AS $$
    BEGIN
        DROP TABLE IF EXISTS x, x1;
        create temporary table x as SELECT s.PRICE sP, r.PRICE rP,
DATE_OF_ANNOUNCEMENT dR, DATE_OF_SALE dS FROM SALE s,
REALTY_OBJECT r WHERE REALTY_OBJECT_ID = r.ID;
        create temporary table x1 as SELECT x.*, rnd((x.sP-x.rP)/x.rP * 100, 2),
getDiffMonth(x.dS, x.dR) dif FROM x;
        RETURN query(SELECT *, (CASE
            WHEN x1.dif < 3 THEN 'Очень быстро'
            WHEN x1.dif >= 3 AND x1.dif < 6 THEN 'Быстро'
            WHEN x1.dif >= 6 AND x1.dif < 12 THEN 'Долго'
            WHEN x1.dif >= 12 THEN 'Очень долго'
            ELSE "
        END) FROM x1 limit 30);
    END $$ language plpgsql;

SELECT * FROM getInfo6_5();

DROP FUNCTION getInfo6_5();

6:

CREATE OR REPLACE FUNCTION getInfo6_6(idRe BIGINT)
RETURNS TABLE("Критерий" character(30), "Средняя оценка" text) AS $$
BEGIN
    DROP TABLE IF EXISTS x;
    create temporary table x as SELECT NAME, RND(AVG(VALUE), 1) avg
FROM Score s, REALTY_OBJECT r, ASSESSMENT_CRITERION asCr WHERE
REALTY_OBJECT_ID = idRe AND REALTY_OBJECT_ID = r.ID AND
ASSESSMENT_CRITERION_ID = asCr.ID GROUP BY NAME;
    RETURN query(SELECT x.NAME, (x.avg || ' из 5') FROM x);
    END $$ language plpgsql;

SELECT * FROM getInfo6_6(20);

DROP FUNCTION getInfo6_6(idRe BIGINT);

7:

CREATE OR REPLACE FUNCTION getInfo6_7(idRe BIGINT, per FLOAT, y
BIGINT, fPay DOUBLE PRECISION)
RETURNS DOUBLE PRECISION AS $$
DECLARE sP DOUBLE PRECISION;
        rez DOUBLE PRECISION;
        i FLOAT;
        n BIGINT;
    BEGIN
        sP := (SELECT s.PRICE sP FROM SALE s, REALTY_OBJECT r WHERE r.ID
= idRe AND REALTY_OBJECT_ID = r.ID limit 1);
        sP := sP - fPay;

```

```

        i := per / 100 / 12;
        n := y * 12;
        rez := sP * (i + i/(POWER(1+i, n) - 1));
        RETURN rnd(rez, 2);
END $$ language plpgsql;

SELECT * FROM getInfo6_7(20, 10.0, 20, 1000000);

DROP FUNCTION getInfo6_7(idRe BIGINT, per FLOAT, y BIGINT, fPay DOUBLE
PRECISION);

8:

CREATE OR REPLACE FUNCTION getInfo6_8(idRe BIGINT, lastTax DOUBLE
PRECISION)
RETURNS DOUBLE PRECISION AS $$
DECLARE sP DOUBLE PRECISION;
        rez DOUBLE PRECISION;
        rate FLOAT;

BEGIN
        sP := (SELECT s.PRICE sP FROM SALE s, REALTY_OBJECT r WHERE r.ID
= idRe AND REALTY_OBJECT_ID = r.ID limit 1);
        rate := (SELECT CASE
                WHEN sP < 10000000 THEN 0.1
                WHEN sP >= 10000000 AND sP < 20000000 THEN 0.15
                WHEN sP >= 20000000 AND sP < 50000000 THEN 0.2
                WHEN sP >= 50000000 AND sP < 300000000 THEN 0.3
        END);
        sP := sP / rand(3, 10);
        rez := (sP * rate - lastTax)* 0.05 + lastTax;
        RETURN rnd(rez, 2);
END $$ language plpgsql;

SELECT * FROM getInfo6_8(20, 15000);

DROP FUNCTION getInfo6_8(idRe BIGINT, lastTax DOUBLE PRECISION);

9:

CREATE TABLE PRICE_MOVEMENT (
        ID BIGINT PRIMARY KEY,
        REALTY_OBJECT_ID BIGINT REFERENCES REALTY_OBJECT (ID),
        NEW_PRICE DOUBLE PRECISION,
        LAGG DOUBLE PRECISION,
        DATE_OF_CHANGE TIMESTAMP
);

CREATE OR REPLACE FUNCTION getInfo6_9(idRe BIGINT)
RETURNS TABLE("Дата" TIMESTAMP, "Новая стоимость" DOUBLE PRECISION
, "Изменение" DOUBLE PRECISION, "% Изменения" FLOAT) AS $$
BEGIN
        DROP TABLE x;

```

```

        create temporary table x as SELECT r.ID rID, s.PRICE sP, s.DATE_OF_SALE
dS, r.PRICE rP FROM SALE s, REALTY_OBJECT r WHERE r.ID = idRe AND
REALTY_OBJECT_ID = r.ID;
        INSERT INTO PRICE_MOVEMENT SELECT
            ROW_NUMBER() over() + 524, x.rID, x.sP, LAG(x.sP) OVER(ORDER
BY x.dS), x.dS
            from x;
        RETURN query(SELECT DATE_OF_CHANGE, NEW_PRICE,
coalesce(NEW_PRICE - LAGG, 0), coalesce(rnd((NEW_PRICE-LAGG)/LAGG * 100, 2), 0)
FROM PRICE_MOVEMENT ORDER BY DATE_OF_CHANGE);
        END $$ language plpgsql;

SELECT * FROM getInfo6_9(20);
--SELECT * FROM PRICE_MOVEMENT;

delete FROM PRICE_MOVEMENT WHERE ID>524;

DROP FUNCTION getInfo6_9(idRe BIGINT);

delete FROM PRICE_MOVEMENT WHERE ID>524;

DROP TABLE PRICE_MOVEMENT;

```



## Результаты

On Disk\Rep\BDIAP\pract\_06\_bd6\_dannyy.sql - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксис Опции Инструменты Макросы Запуск Плагины Вкладки ? + ▾

BD6.sql BD6\_6es\_dannyy.sql

```

25
26 DROP FUNCTION getSalary(S float, R bigint, f text, beg TIMESTAM
27

SQL Shell (psql)

psql (15.4):
Наведите "help", чтобы получить справку.
Наведите "?column" для описания типов (для "?" используйте "?" для
realty=#
realty=# SELECT * FROM getexchangesRate(124, 88.81, 96.92);
 0 руб/лв | 0 долларов | 0 евро
19993181 | 225123.08 | 206285.4
(1 строка)

CREATE OR REPLACE FUNCTION getSalary(S float, R bigint, f text, beg TIMESTAM
RETURNING TABLE OF bigint AS
BEGIN
RETURN QUERY SELECT * FROM getSalary(S, R, f, beg, beg);
END;

realty=# SELECT * FROM getSalary(0.002, 40000, 'Микшина', '2017-05-07', '2023-11-24');
БАЛАНС: таблица "K" не существует, пропускается
БАЛАНС: таблица "K" не существует, пропускается
idrr | last_name | l | n | cnt | aridrr | sumobj | Сумма заработной платы
-----+-----+-----+-----+-----+-----+-----
14 | Микшина | K | 1 | 1 | 166 | 62 333.58
14 | Микшина | K | 1 | 1 | 183 | 70 279.63
14 | Микшина | K | 1 | 1 | 172 | 63 667.92
14 | Микшина | K | 1 | 1 | 175 | 71 440.15
14 | Микшина | K | 1 | 1 | 3 | 64 000.00
14 | Микшина | K | 1 | 1 | 19 | 64 087.96
14 | Микшина | K | 1 | 1 | 168 | 74 329.68
14 | Микшина | K | 1 | 1 | 163,168,129 | 134 915.81
14 | Микшина | K | 1 | 1 | 135 | 69 425.28
14 | Микшина | K | 1 | 1 | 142 | 73 348.01
14 | Микшина | K | 1 | 1 | 127 | 66 993.81
14 | Микшина | K | 1 | 1 | 143,129,101 | 130 376.89
14 | Микшина | K | 1 | 1 | 161 | 67 899.71
14 | Микшина | K | 1 | 1 | 143 | 79 865.76
14 | Микшина | K | 1 | 1 | 3,162,135,125 | 164 636.26
14 | Микшина | K | 1 | 1 | 135 | 69 425.28
(16 строка)

CREATE OR REPLACE FUNCTION getSalary(S float, R bigint, f text, beg TIMESTAM
RETURNING TABLE OF bigint AS
BEGIN
RETURN QUERY SELECT * FROM getSalary(S, R, f, beg, beg);
END;

realty=#

```

Структ length: 7449 lines: 104

On Disk\Rep\BDIAP\pract\_06\_bd6\_dannyy.sql - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксис Опции Инструменты Макросы Запуск Плагины Вкладки ? + ▾

BD6.sql BD6\_6es\_dannyy.sql

Практическое зан...

1 / 2 100%

Вид учебного материала Практика

Преподаватель Евдошенко Олег Игоревич

Эстр 1 семестр, 2023-2024

1. Конвертирует стоимость объекта недвижимости в евро и долларх.

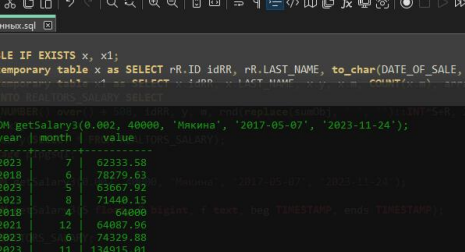
Вх. параметр	Вых. параметр
курс, код объекта недвижимости	стоимость в евро
2. Рассчитывает заработную плату ризлтора по формуле:  $N * S * R$ , где  $N$  – общая сумма проданных объектов недвижимости в месяц (получет автоматически по данным таблицы «Продажи» с использованием агрегатной функции),  $S$  – коэффициент,  $R$  – премия.

Вх. параметр	Вых. параметр
N, R, начальная и конечная дата, фамилия ризлтора	размер заработной платы
3. Добавить таблицу «Заработная плата ризлтора», содержащая значения: год, месяц, размер выплаты. Изменить тело функции из пункта 2 таким образом, чтобы рассчитанная заработная плата сохранялась в этой таблице.
4. Рассчитывает процент изменения продажной стоимости объекта недвижимости от первоначальной заявленной и срок продажи (в месяцах)
5. Изменить функцию, созданную в пункте 5 таким образом, чтобы в зависимости от срока продажи выводилось сообщение

Срок	Сообщение
менее 3 месяцев	Очень быстро
от 3 месяцев до 6 месяцев	Быстро
от 6 месяцев до 1 года	Долго
от 1 года	Очень долго
6. Формирует список средних оценок по каждому критерию для объекта недвижимости.
7. Написать функцию, которая рассчитывает сумму ежемесячного платежа по ипотеке. **Входные параметры:** код объекта недвижимости, процентная ставка, срок, первоначальный взнос.

Структ length: 97413

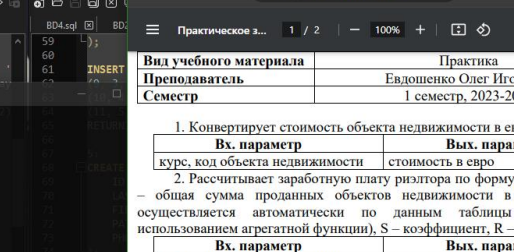
Рисунок 1 – Результат к 1 и 2 задаче



```

40 BEGIN
41 DROP TABLE IF EXISTS x;
42 create temporary table x as SELECT r.R_ID rRR, r.R_LAST_NAME, to_char(DATE_OF_SALE,
43 'DD-MM-YY') as DATE_OF_SALE FROM REALTY3 r WHERE r.R_LAST_NAME = 'Иванов';
44 SQL Shell (psql)
45 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
46 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
47 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
48 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
49 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
50 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
51 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
52 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
53 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
54 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
55 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
56 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
57 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
58 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
59 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
60 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
61 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
62 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
63 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
64 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
65 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
66 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
67 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
68 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
69 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
70 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
71 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
72 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
73 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
74 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
75 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
76 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
77 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
78 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
79 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
80 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
81 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
82 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
83 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
84 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
85 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
86 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
87 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
88 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
89 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
90 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
91 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
92 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
93 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
94 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
95 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
96 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
97 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
98 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
99 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');
100 realty=# SELECT * FROM getSalary3(0.002, 40000, 'Иванов', '2017-05-07', '2023-11-24');

```



Вид учебного материала	Практика
Преподаватель	Евдошенко Олег Игоревич
Семестр	1 семестр, 2023-2024

1. Конвертирует стоимость объекта недвижимости в евро и доллары.
2. Рассчитывает заработную плату ризитора по формуле:  $N * S + R$ , где  $N$  – общая сумма проданных объектов недвижимости в месяц (подсчит осуществляется автоматически по данным таблицы «Продажи» с использованием агрегатной функции),  $S$  – коэффициент,  $R$  – премия.
3. Добавить таблицу «Зароботная плата ризитора», содержащая сведения, год, месяц, размер выплаты. Изменить тело функции из пункта 2 таким образом, чтобы рассчитанная заработная плата сохранялась в этой таблице.
4. Рассчитывает процент изменения продажной стоимости объекта недвижимости от первоначально заявленной и срок продажи (в месяцах)
5. Изменить функцию, созданную в пункте 5 таким образом, чтобы в зависимости от срока продажи выводилось сообщение

Срок	Сообщение
менее 3 месяцев	Очень быстро
от 3 месяцев до 6 месяцев	Быстро
от 6 месяцев до 1 года	Долго
от 1 года	Очень долго

6. Формирует список средних оценок по каждому критерию для объекта недвижимости.
7. Написать функцию, которая рассчитывает сумму ежемесячного платежа по ипотеке. **Входные параметры:** код объекта недвижимости, процентная ставка, срок, первоначальный взнос.

Рисунок 2 – Результат к 3 и 4 задаче

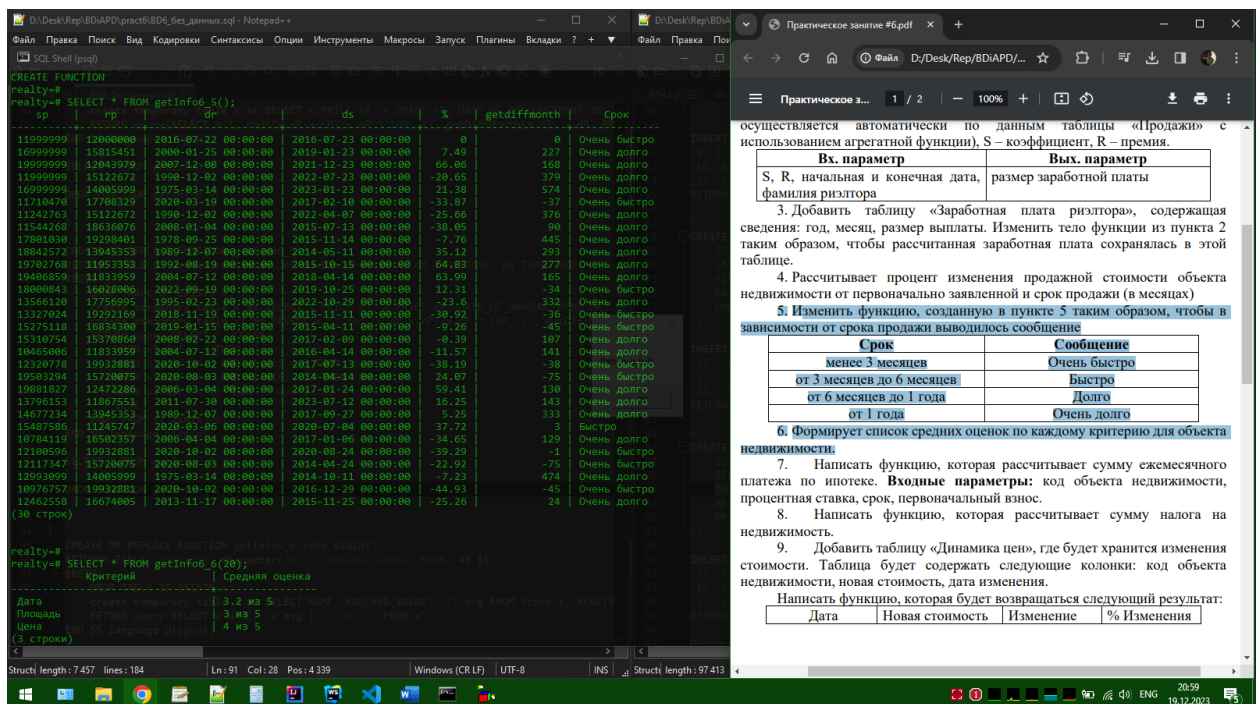


Рисунок 3 – Результат к 5 и 6 задаче

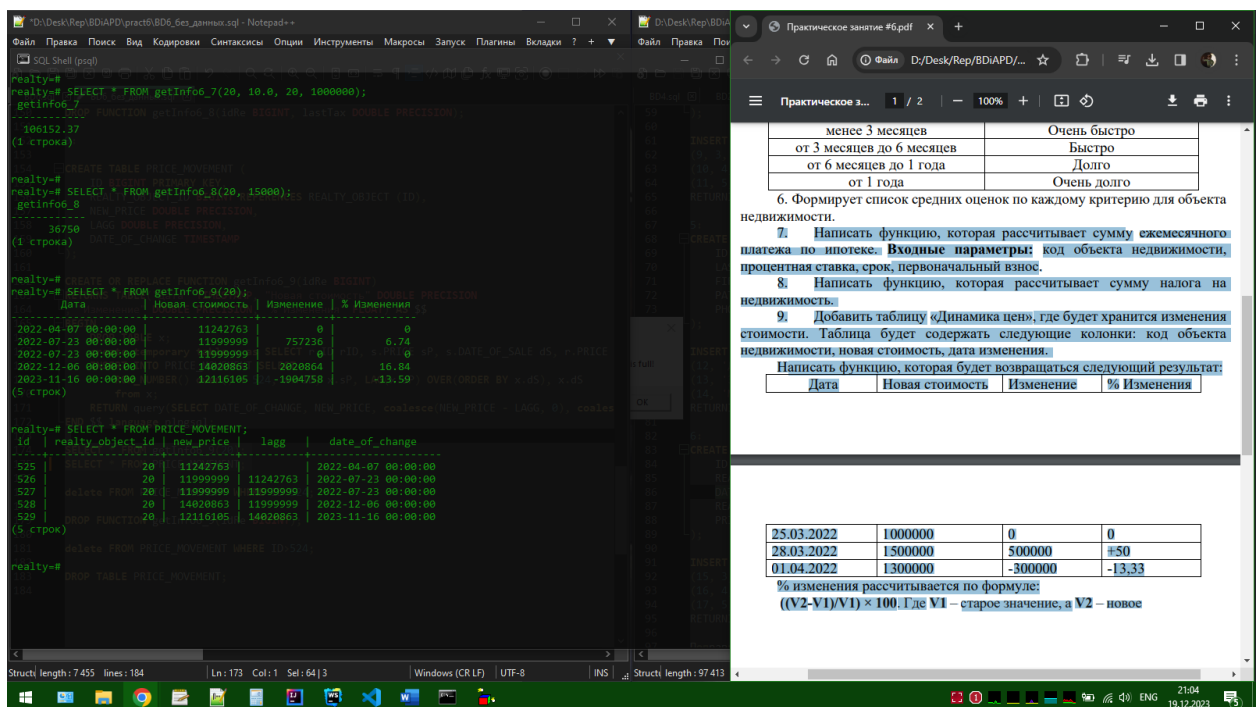


Рисунок 4 – Результат к 7, 8 и 9 задаче