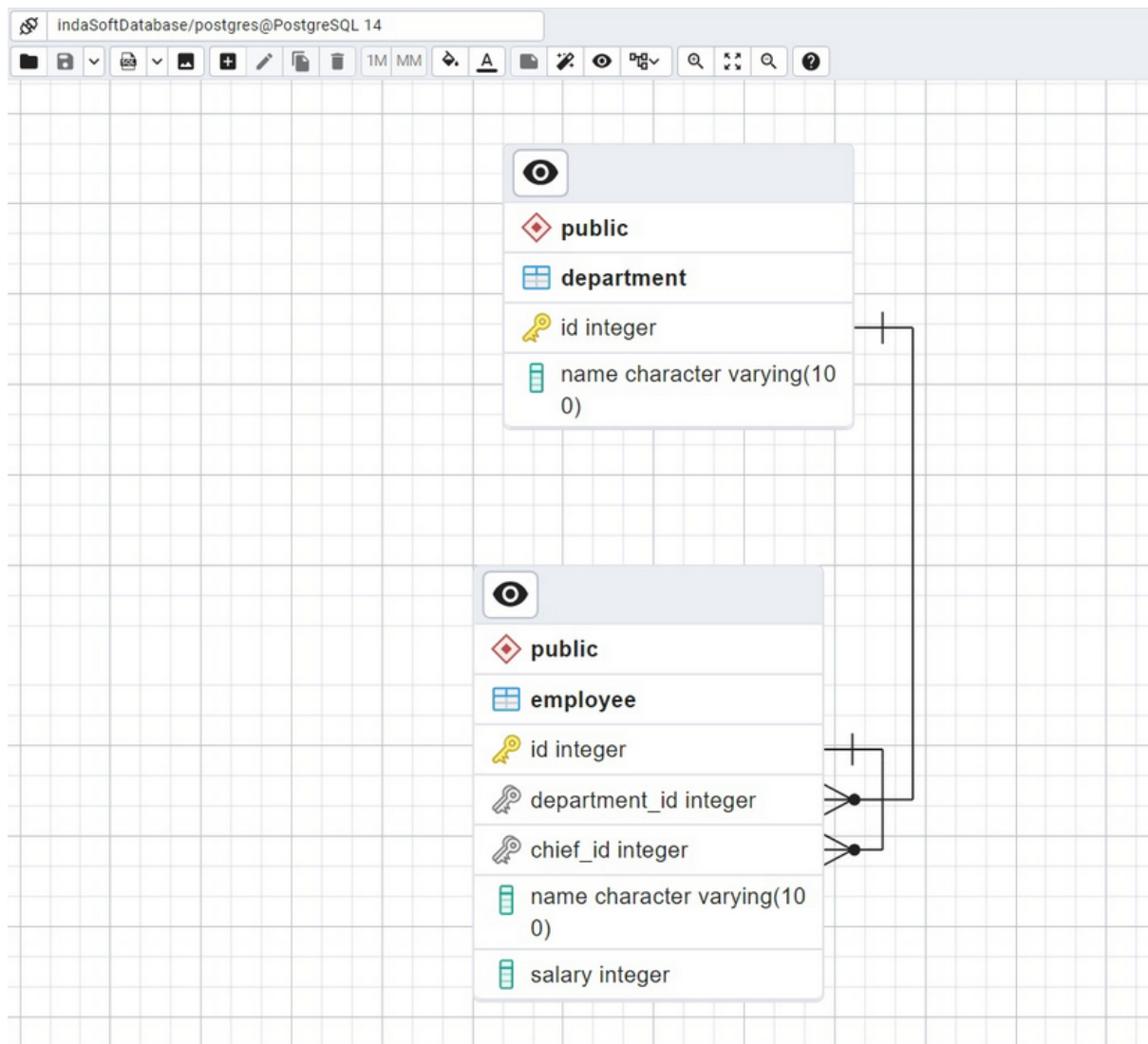


## Отчет по ТЗ 1

1. Развернуть на своем АРМ PostgreSQL версии 13/14. К данному SQL серверу установить дополнительно pgAdmin 4.
2. Создать средствами pgAdmin БД с двумя таблицами по схеме из задания 1 (SQL):  
Прислать скриншот со схемой таблиц в pgAdmin.



Далее тип значения salary был изменен на “numeric”, из-за удобства работы с этим типом данных и более корректной обработки.

### 3. Прислать выгрузку данных из двух таблиц.

Таблица department:



















Data Output   Сообщения   Notifications		
        		
	id [PK] integer	name character varying (100)
1	1	Отдел продаж
2	2	Отдел разработки
3	3	Отдел маркетинга

Таблица employee:

Data Output   Сообщения   Notifications					
        					
	id [PK] integer	department_id integer	chief_id integer	name character varying (100)	salary numeric
1	1	3	10	Пирогов Александр Иванович	27000
2	2	3	10	Рыбин Алексей Алексеевич	34000
3	3	3	10	Лесничий Василий Федорович	130000
4	4	3	10	Ивина Мария Васильевна	76000
5	5	3	10	Плетнев Александр Алексеевич	56000
6	6	3	10	Михайленко Олег Михайлович	110000
7	7	2	7	Симонов Виктор Павлович	130000
8	8	2	7	Ерофеева Алина Васильевна	40000
9	9	2	7	Погорянский Ростислав Александрович	100000
10	10	2	7	Петренко Сергей Иванович	80000
11	11	2	7	Соболев Михаил Олегович	20000
12	12	1	12	Петров Валентин Владимирович	80000
13	13	2	7	Сидоров Владислав Васильевич	40000
14	14	1	12	Сильченко Алексей Игоревич	70000
15	15	1	12	Абрамов Антон Сергеевич	40000

Некоторые параметры могли быть подвержены изменению, для выполнения выборок в 4ом задании.

**4. Для реализованных таблиц реализовать следующие SQL-запросы (прислать текст запросов и результат выборки):**

4.1 Вывести список сотрудников, которые получают заработную плату ниже, чем у непосредственного руководителя.

```
SELECT e.id, e.name  
FROM public.employee e  
JOIN public.employee chief ON e.chief_id = chief.id  
WHERE e.salary < chief.salary;
```

Вывод:

	id [PK] integer	name character varying (100)
1	14	Сильченко Алексей Игоревич
2	15	Абрамов Антон Сергеевич
3	1	Пирогов Александр Иванович
4	2	Рыбин Алексей Алексеевич
5	8	Ерофеева Алина Васильевна
6	11	Соболев Михаил Олегович
7	13	Сидоров Владислав Васильевич

4.2 Вывести список сотрудников, которые получают в отделе минимальную заработную плату в своем отделе.

```
SELECT e.id, e.name
FROM public.employee e
JOIN (
    SELECT department_id, MIN(salary) AS min_salary
    FROM public.employee
    GROUP BY department_id
) min_salaries ON e.department_id = min_salaries.department_id
WHERE e.salary = min_salaries.min_salary;
```

Вывод:

Data Output			Сообщения	Notifications
	id [PK] integer		name character varying (100)	
1	15		Абрамов Антон Сергеевич	
2	1		Пирогов Александр Иванович	
3	11		Соболев Михаил Олегович	

4.3 Вывести список ID отделов, количество сотрудников в которых не превышает трех человек.

```
SELECT department_id
FROM public.employee
GROUP BY department_id
HAVING COUNT(*) <= 3;
```

Вывод:

Data Output    Сообще

<div> <div>☰+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> </div>		
	department_id integer	🔒
1		1

4.4 Вывести список сотрудников, не имеющих назначенного руководителя, который работал бы в том же отделе.

```

SELECT e.id, e.name
FROM public.employee e
LEFT JOIN public.employee chief ON e.chief_id = chief.id
WHERE chief.id IS NULL OR chief.department_id !=
e.department_id;

```

Вывод:

Data Output    Сообщения    Notifications

<div> <div>☰+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	id [PK] integer	✎	name character varying (100)
1	3		Лесничий Василий Федорович
2	4		Ивина Мария Васильевна
3	5		Плетнев Александр Алексеевич
4	6		Михайленко Олег Михайлович
5	1		Пирогов Александр Иванович
6	2		Рыбин Алексей Алексеевич

4.5 Найти список ID отделов с максимальной суммарной заработной платой сотрудников.

```
SELECT department_id  
FROM public.employee  
GROUP BY department_id  
HAVING SUM(salary) = (SELECT MAX(total_salary) FROM (SELECT  
SUM(salary) AS total_salary FROM public.employee GROUP BY  
department_id) AS subquery);
```








Вывод:

Data Output		Сообщения	
<div><div></div><div></div><div></div><div></div><div></div><div></div></div>			
	department_id integer		
1		3	

4.6 Составить SQL-запрос, вычисляющий сумму всех значений всех ЗП в конкретном столбце таблицы.

```
SELECT SUM(salary) AS total_salary  
FROM public.employee  
WHERE department_id = 1
```

Вывод:

Data Output		Сообщения	
<div><div></div></div>			
	total_salary numeric		
1	140000		

**5. Реализовать хранимую процедуру (UPDATESALARYFORDEPARTMENT) со**

**следующими условиями:**

**5.1 Входные данные: ID отдела, PERCENT процент повышения ЗП**

**5.2 Логика: данная процедура должна у всех сотрудников в рамках отдела с**

**заданным ID (кроме начальника отдела) повышать ЗП на заданный**

**процент (PERCENT). В случае, если после повышения ЗП у кого-либо из**

**сотрудников ЗП оказалась выше, чем у начальника отдела, то повысить**

**ЗП для начальника до аналогичной ЗП.**

**5.3 На выходе вернуть перечень сотрудников (все данные из таблицы**

**employee) с обновленной и старой ЗП.**

Текст ХП

DECLARE

chief\_salary NUMERIC;

newchief\_salary NUMERIC;

salaries NUMERIC[];

max\_salary NUMERIC;

BEGIN

-- Получим зарплату начальника отдела

SELECT e.salary INTO chief\_salary

FROM public.employee e

WHERE e.department\_id = newdepartment\_id AND e.id =

e.chief\_id;

-- Получим увеличенную зарплату сотрудников отдела

SELECT ARRAY\_AGG(ROUND(e.salary + e.salary \* (percent / 100)))

```
    INTO salaries
    FROM public.employee e
    WHERE e.id != e.chief_id and e.department_id =
newdepartment_id;
```

```
    -- Найдем максимальное значение в массиве salaries
    SELECT MAX(salary)
    INTO max_salary
    FROM unnest(salaries) AS salary;
```

```
    -- Обновим зарплаты сотрудников
    RETURN QUERY
    SELECT
    e.id,
    e.name,
    CASE
    WHEN e.id != e.chief_id THEN
        CASE
        WHEN (e.salary + e.salary * (percent / 100)) < chief_salary
THEN
            ROUND(e.salary + e.salary * (percent / 100))

        ELSE
        ROUND(e.salary + e.salary * (percent / 100))
        END
        WHEN e.id = e.chief_id THEN
            CASE
            WHEN chief_salary < max_salary THEN
                max_salary
            ELSE
                chief_salary
            END
        END

    END AS new_salary,
    e.salary AS old_salary
```



```
FROM public.employee e
WHERE e.department_id = newdepartment_id;
END;
```

**P.S** до этого была введена команда для добавления ХП, с двумя входными аргументами newdepartment\_id и percent.

Вызов процедуры

```
SELECT * FROM updatesalaryfordepartment2(1,27);
```

Вывод:

Data Output   Сообщения   Notifications					
	id integer		name character varying		
1	12		Петров Валентин Владимирович	new_salary numeric	old_salary numeric
2	15		Абрамов Антон Сергеевич	88900	80000
3	14		Сильченко Алексей Игоревич	50800	40000
				88900	70000

## Скриншот ХП из pgAdmin 4

(=) updatesalaryfordepartment2(newdepartment\_id integer, percent numeric)

General Определение Код Параметры Security SQL

```
1 DECLARE
2     chief_salary NUMERIC;
3     newchief_salary NUMERIC;
4     salaries NUMERIC[];
5     max_salary NUMERIC;
6 BEGIN
7     -- Получим зарплату начальника отдела
8     SELECT e.salary INTO chief_salary
9     FROM public.employee e
10    WHERE e.department_id = newdepartment_id AND e.id = e.chief_id;
11
12
13    -- Получим увеличенную зарплату сотрудников отдела
14    SELECT ARRAY_AGG(ROUND(e.salary + e.salary * (percent / 100)))
15    INTO salaries
16    FROM public.employee e
17    WHERE e.id != e.chief_id and e.department_id = newdepartment_id;
18
19    -- Найдем максимальное значение в массиве salaries
20    SELECT MAX(salary)
21    INTO max_salary
22    FROM unnest(salaries) AS salary;
23
24    -- Обновим зарплаты сотрудников
25    RETURN QUERY
26    SELECT
27        e.id,
28        e.name,
29        CASE
30            WHEN e.id != e.chief_id THEN
31                CASE
32                    WHEN (e.salary + e.salary * (percent / 100)) < chief_salary THEN
33                        ROUND(e.salary + e.salary * (percent / 100))
34
35
36                    ELSE
37                        ROUND(e.salary + e.salary * (percent / 100))
38                END
39            WHEN e.id = e.chief_id THEN
40                CASE
41                    WHEN chief_salary < max_salary THEN
42                        max_salary
43                    ELSE
44                        chief_salary
```

## Настройки

updatesalaryfordepartment2(newdepartment\_id integer, percent numeric)

General Определение Код Параметры Security SQL

Тип результата

TABLE(id integer, name character varying, new\_salary numeric, old\_salary nur

Язык

plpgsql

Аргументы

	Тип данных	Режим	Argument name	По умолчанию
	integer   v	IN   v	newdepartment_id	
	numeric   v	IN   v	percent	

Закрыть

Сбросить

Сохранить

updatesalaryfordepartment2(newdepartment\_id integer, percent numeric)

General Определение Код Параметры Security SQL

Изменчивость

VOLATILE | v

Возвращает множество?

☒

Строгая?

☐

Определяет контекст безопасности?

☐

Оконная?

☐

Параллельная

UNSAFE | v

Ожидаемая стоимость

100

Ожидаемое число строк

1000

Герметичная?

☐

Support function

Закрыть

Сбросить

Сохранить

Остальное по-умолчанию.

