

sem12_Nemkov

Task 1

Найдите сотрудника, получающего вторую максимальную зарплату.

Дополнения:

- написать запрос без оконных функций (или с ними, если изначально запрос был без них);
- найти сотрудника, получающего третью максимальную зарплату (неважно с применением оконных функций или без них).

1. В подзапросе выбираем всех сотрудников, у которых зп меньше максимальной. Из этого списка берем топ-1

```
SELECT e.*
FROM employees e
WHERE e.salary = (
    SELECT MAX(salary)
    FROM employees
    WHERE salary < (
        SELECT MAX(salary)
        FROM employees
    )
);
```

	id [PK] integer	name character varying (50)	department_id integer	grade character varying (50)	salary integer	email character varying (50)	boss_id integer	salary_rank bigint
1	18	Антон	1	senior	125	onmer@gmail.com	[null]	2

image-310.png

2) То же самое с оконной функцией: указываем, что нужно ранжировать по фрейму salary и берем топ-2

```
WITH ranked_employees AS (
    SELECT
        e.*,
        DENSE_RANK() OVER (ORDER BY salary DESC) as salary_rank
    FROM employees e
)
SELECT *
```

```
FROM ranked_employees
WHERE salary_rank = 2;
```

	id [PK] integer	name character varying (50)	department_id integer	grade character varying (50)	salary integer	email character varying (50)	boss_id integer	salary_rank bigint
1	18	Антон	1	senior	125	onmer@gmail.com	[null]	2

image-310.png

3) Ищем топ-3

```
WITH ranked_salaries AS (
    SELECT
        e.*,
        DENSE_RANK() OVER (ORDER BY salary DESC) as salary_rank
    FROM employees e
)
SELECT *
FROM ranked_salaries
WHERE salary_rank = 3;
```

	id [PK] integer	name character varying (50)	department_id integer	grade character varying (50)	salary integer	email character varying (50)	boss_id integer	salary_rank bigint
1	31	Антон	1	senior	120	onmer@gmail.com	[null]	3
2	4	Иван	1	senior	120	sdfeq@gmail.com	[null]	3

image-311.png

Проверка

Увидев весь список:

```
SELECT * FROM employees ORDER BY salary DESC;
```

	id [PK] integer	name character varying (50)	department_id integer	grade character varying (50)	salary integer	email character varying (50)	boss_id integer
1	21	Ирина	7	senior	150	luqjs@gmail.com	[null]
2	18	Антон	1	senior	125	onmer@gmail.com	[null]
3	31	Антон	1	senior	120	onmer@gmail.com	[null]
4	4	Иван	1	senior	120	sdfeq@gmail.com	[null]
5	1	Марина	1	middle	104	dgshj@gmail.com	4
6	5	Леонид	1	middle	104	iubuk@yandex.ru	4
7	9	Анна	9	senior	100	dldl@yandex.ru	[null]
8	25	Денис	1	senior	100	pertse@gmail.com	[null]

image-312.png

Можно убедиться, что результаты моих запросов верны

Task 2

Вывести email-адреса, встречающиеся больше одного раза.

Дополнения:

- найти сотрудников с одинаковыми email-адресами и именами;
 - вывести минимальный id из каждой группы дубликатов.
1. Выводим email-адреса, встречающиеся больше одного раза (группируем по email и считаем размер каждой группы)

```
SELECT email, COUNT(*) as email_count
FROM employees
GROUP BY email
HAVING COUNT(*) > 1;
```

	email character varying (50) 🔒	email_count bigint 🔒
1	dllldl@yandex.ru	2
2	leusn@yandex.ru	2
3	pertse@gmail.com	2
4	prbwh@mail.ru	2
5	ybdnk@yandex.ru	2
6	ososos@mail.ru	2
7	keytri@gmail.com	2
8	luqjs@gmail.com	4
9	oeeous@gmail.com	2
10	Indy@gmail.com	2
11	onmer@gmail.com	2
12	pebhgd@gmail.com	2
Total rows: 12 of 12 Query complete 00:00:00.085		

image-313.png

2. Ищем сотрудников с одинаковыми email-адресами и именами (делаем то же самое, но добавляем группировку на имена)

```
SELECT name, email, COUNT(*) as duplicate_count
FROM employees
GROUP BY name, email
HAVING COUNT(*) > 1;
```

	name character varying (50) 🔒	email character varying (50) 🔒	duplicate_count bigint 🔒
1	Михаил	pebhgd@gmail.com	2
2	Денис	pertse@gmail.com	2
3	Максим	ybdnk@yandex.ru	2
4	Антон	onmer@gmail.com	2
5	Сергей	oeeous@gmail.com	2
6	Алина	prbwh@mail.ru	2
7	Константин	Indy@gmail.com	2
8	Ирина	luqjs@gmail.com	4
9	Ангелина	keytri@gmail.com	2
10	Евгений	leusn@yandex.ru	2
11	Анна	dllldl@yandex.ru	2
Total rows: 11 of 11 Query complete 00:00:00.074 Ln 35, Col 1			

image-314.png

3. Выводим минимальный id у дубликатов

```

WITH duplicates AS (
    SELECT
        name,
        email,
        COUNT(*) as duplicate_count,
        MIN(id) as min_id
    FROM employees
    GROUP BY name, email
    HAVING COUNT(*) > 1
)
SELECT *
FROM duplicates
ORDER BY min_id;

```

	name character varying (50) 🔒	email character varying (50) 🔒	duplicate_count bigint 🔒	min_id integer 🔒
1	Анна	dllldl@yandex.ru	2	9
2	Ангелина	keytri@gmail.com	2	10
3	Денис	pertse@gmail.com	2	12
4	Максим	ybdnk@yandex.ru	2	13
5	Сергей	oeeous@gmail.com	2	14
6	Михаил	pebhgd@gmail.com	2	15
7	Константин	Indy@gmail.com	2	16
8	Алина	prbwh@mail.ru	2	17
9	Антон	onmer@gmail.com	2	18
10	Евгений	leusn@yandex.ru	2	19
11	Ирина	luqjs@gmail.com	4	20
Total rows: 11 of 11 Query complete 00:00:00.079 Ln 41, Col 12				

image-315.png

Task 3

Найти топ-3 отдела по средней зарплате сотрудников.

Дополнения:

- необходимо учесть только отделы с количеством сотрудников больше 3;
- найти департамент с максимальной медианной зарплатой.

1. Топ-3 отдела по средней зарплате (с учетом отделов, где больше 3 сотрудников)

```
SELECT
    d.id,
    d.name as department_name,
    AVG(e.salary) as average_salary,
    COUNT(e.id) as employee_count
FROM departments d
JOIN employees e ON d.id = e.department_id
GROUP BY d.id, d.name
HAVING COUNT(e.id) > 3
ORDER BY average_salary DESC
LIMIT 3;
```

	id [PK] integer	department_name character varying (50)	average_salary numeric	employee_count bigint
1	1	it	102.6363636363636364	11
2	9	sales	79.6250000000000000	8
3	2	hr	77.2857142857142857	7
Total rows: 3 of 3 Query complete 00:00:00.083 Ln 54, Col 1				

image-316.png

2) Департамент с максимальной медианной зарплатой

```
WITH department_medians AS (
    SELECT
        d.id,
        d.name as department_name,
        PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY e.salary) as
median_salary,
        COUNT(e.id) as employee_count
    FROM departments d
    JOIN employees e ON d.id = e.department_id
    GROUP BY d.id, d.name
)
SELECT *
FROM department_medians
ORDER BY median_salary DESC
LIMIT 1;
```

	id [PK] integer	department_name character varying (50)	median_salary double precision	employee_count bigint
1	7	business	150	1

image-317.png

Task 4

Найти все департаменты, в которых нет сотрудников.

Дополнения:

- что если у сотрудника department_id = NULL? Как это повлияет на запрос?

1. Основной запрос для поиска департаментов без сотрудников

```
SELECT d.*
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
WHERE e.id IS NULL;
```

	id [PK] integer	name character varying (50)
1	5	design
2	6	pr

image-318.png

2) Если у некоторых сотрудников `department_id` может быть NULL, это не повлияет на основной запрос, так как:

- LEFT JOIN соединяет все записи из таблицы `departments` с соответствующими записями из `employees`
- Сотрудники с `department_id = NULL` не будут соответствовать ни одному департаменту и не появятся в результате JOIN
- Условие `WHERE e.id IS NULL` отфильтрует только департаменты, у которых нет ни одного сотрудника (включая случаи, когда сотрудники есть, но с NULL `department_id`)

Task 5

Вывести иерархию подчинения для сотрудника с `id = 5`.

Дополнения:

- добавьте условие: вывести только сотрудников с зарплатой `> 90`.
1. По аналогии с заданием про компьютерные игры в одной из предыдущих работ сделаю с помощью рекурсии:

```
WITH RECURSIVE employee_hierarchy AS (
    SELECT id, name, salary, boss_id, 1 AS level
    FROM employees
    WHERE id = 5 AND salary > 90

    UNION ALL

    SELECT e.id, e.name, e.salary, e.boss_id, eh.level + 1
    FROM employees e
```

```
JOIN employee_hierarchy eh ON e.boss_id = eh.id
WHERE e.salary > 90
)
SELECT id, name, boss_id, level
FROM employee_hierarchy
ORDER BY level, id;
```





	id integer 	name character varying (50) 	boss_id integer 	level integer 
1	5	Леонид	4	1
2	10	Ангелина	5	2
3	23	Ангелина	5	2
Total rows: 3 of 3 Query complete 00:00:00.115 Ln 93, Col 1				

image-320.png