

# sem4\_Nemkov\_KMBO\_05\_23

## Task 1

Посчитать количество сотрудников и их среднюю зарплату, сгруппированные по:

- отделу, полу и грейду;
- отделу и полу;
- отделу.

Также вывести общий итог (все отделы, оба пола, все грейды).

Вывод должен быть в формате:

- отдел (в случае группировки всех отделы выводить 'All Departments');
- пол (в случае группировки полов выводить 'All Genders');
- грейд (в случае группировки грейдов выводить 'All Grades');
- количество сотрудников;
- средняя зарплата.

*Указание:* для отдела, пола и грейда используйте функцию COALESCE.

```
-- группируем по department, gender, grade
SELECT
    department,
    gender,
    grade,
    COUNT(*) AS employee_count,
    AVG(salary) AS average_salary
FROM employees
GROUP BY department, gender, grade
UNION ALL
-- группируем по department, gender
SELECT
    department,
    gender,
    'All Grades' AS grade,
    COUNT(*) AS employee_count,
    AVG(salary) AS average_salary
FROM employees
GROUP BY department, gender
UNION ALL
-- группируем по department
SELECT
    department,
    'All Genders' AS gender,
    'All Grades' AS grade,
    COUNT(*) AS employee_count,
    AVG(salary) AS average_salary
FROM employees
GROUP BY department
UNION ALL
-- итог без группировки
```

```

SELECT
    'All Departments' AS department,
    'All Genders' AS gender,
    'All Grades' AS grade,
    COUNT(*) AS employee_count,
    AVG(salary) AS average_salary
FROM employees
ORDER BY department, gender, grade;

```

result:

	department character varying	gender character varying	grade character varying	employee_count bigint	average_salary numeric
1	All Departments	All Genders	All Grades	20	89.0000000000000000
2	hr	All Genders	All Grades	6	79.3333333333333333
3	hr	жен	All Grades	3	81.6666666666666667
4	hr	жен	middle	1	70.0000000000000000
5	hr	жен	senior	2	87.5000000000000000
6	hr	муж	All Grades	3	77.0000000000000000
7	hr	муж	junior	1	65.0000000000000000
8	hr	муж	middle	1	78.0000000000000000
9	hr	муж	senior	1	88.0000000000000000
10	it	All Genders	All Grades	8	102.8750000000000000
11	it	жен	All Grades	4	93.5000000000000000
12	it	жен	junior	2	87.0000000000000000
13	it	жен	middle	2	100.0000000000000000
14	it	муж	All Grades	4	112.2500000000000000
15	it	муж	middle	2	102.0000000000000000
16	it	муж	senior	2	122.5000000000000000
17	sales	All Genders	All Grades	6	80.1666666666666667
18	sales	жен	All Grades	3	87.3333333333333333
19	sales	жен	junior	1	66.0000000000000000
20	sales	жен	senior	2	98.0000000000000000
Total rows: 24 of 24    Query complete 00:00:00.134    Ln 17, Col 31					

*image-69.png*

## Task 2

Для каждого сотрудника вывести сколько процентов составляет его зарплата от общей зарплате, сгруппированной по отделу и грейду, а также количество сотрудников в отделе, имеющих такой же грейд, но получающих БОльшую зарплату.

Вывод должен быть в формате:

- имя;
- отдел;
- грейд;
- общая зарплата, сгруппированная по отделу и грейду;
- процент зарплате сотрудника от общей зарплате;
- количество сотрудников того же отдела и грейда, получающих БОльшую зарплату.

```

-- cte для расчета общей зарплате по каждому отделу и грейду
WITH department_grade_salary AS (
    SELECT
        department,
        grade,

```

```

        SUM(salary) AS total_salary
    FROM employees
    GROUP BY department, grade
),
-- cte для расчета информации по каждому сотруднику
employee_stats AS (
    SELECT
        e.id,
        e.name,
        e.department,
        e.grade,
        e.salary,
        dgs.total_salary, -- общая зарплата по отделу и грейду из первого CTE
        -- процент зарплаты сотрудника от общей зарплаты по отделу и грейду
        ROUND((e.salary * 100.0 / dgs.total_salary), 2) AS salary_percentage,
        -- подзапрос для подсчета количества сотрудников в том же отделе и с тем же
        грейдом,
        -- но у которых больше зп
        (
            SELECT COUNT(*)
            FROM employees e2
            WHERE e2.department = e.department
                AND e2.grade = e.grade
                AND e2.salary > e.salary
        ) AS higher_salary_count
    FROM employees e
    -- джойним CTE, чтобы получить общую зарплату по отделу и грейду
    JOIN department_grade_salary dgs
        ON e.department = dgs.department
        AND e.grade = dgs.grade
)
-- берем все что нужно по условиям:
SELECT
    name,
    department,
    grade,
    total_salary,
    salary_percentage,
    higher_salary_count
FROM employee_stats
ORDER BY department, grade, name;

```

**Result:**

	name character varying (50)	department character varying (50)	grade character varying (50)	total_salary bigint	salary_percentage numeric	higher_salary_count bigint
1	Михаил	hr	junior	65	100.00	0
2	Борис	hr	middle	148	52.70	0
3	Дарья	hr	middle	148	47.30	1
4	Алина	hr	senior	263	34.22	0
5	Ирина	hr	senior	263	32.32	2
6	Сергей	hr	senior	263	33.46	1
7	Елена	it	junior	174	48.28	1
8	Ксения	it	junior	174	51.72	0
9	Ангелина	it	middle	404	23.76	3
10	Денис	it	middle	404	24.75	2
11	Леонид	it	middle	404	25.74	0
12	Марина	it	middle	404	25.74	0
13	Антон	it	senior	245	51.02	0
14	Иван	it	senior	245	48.98	1
15	Евгений	sales	junior	130	49.23	1
16	Оксана	sales	junior	130	50.77	0
17	Максим	sales	middle	70	100.00	0
18	Анна	sales	senior	281	35.59	0
19	Вероника	sales	senior	281	34.16	1
20	Константин	sales	senior	281	30.25	2

Total rows: 20 of 20    Query complete 00:00:00.163    Ln 1, Col 1

image-70.png

## Task 3

Вывести самого низкооплачиваемого и самого высокооплачиваемого сотрудника из каждого города.

Вывод должен быть в формате:

- имя;
- город
- зарплата;
- статус («Высокооплачиваемый» или «Низкооплачиваемый»).

```
WITH salary_ranks AS (
    SELECT
        name,
        city,
        salary,
        -- ранжируем сотрудников по зарплате в каждом городе по возрастанию
        ROW_NUMBER() OVER (PARTITION BY city ORDER BY salary ASC) AS min_salary_rank,
        -- ранжируем сотрудников по зарплате в каждом городе по убыванию
        ROW_NUMBER() OVER (PARTITION BY city ORDER BY salary DESC) AS max_salary_rank
    FROM employees
)
SELECT
    name,
    city,
    salary,
    CASE
        WHEN min_salary_rank = 1 THEN 'Низкооплачиваемый'
        WHEN max_salary_rank = 1 THEN 'Высокооплачиваемый'
    END AS status
FROM salary_ranks
WHERE
    min_salary_rank = 1
```

```
OR max_salary_rank = 1
ORDER BY city, status;
```

## Result:

	name character varying (50) 🔒	city character varying (50) 🔒	salary integer 🔒	status text 🔒
1	Ангелина	Воронеж	96	Высокооплачиваемый
2	Сергей	Воронеж	88	Низкооплачиваемый
3	Алина	Казань	90	Низкооплачиваемый
4	Антон	Москва	125	Высокооплачиваемый
5	Михаил	Москва	65	Низкооплачиваемый
6	Леонид	Самара	104	Высокооплачиваемый
7	Евгений	Самара	64	Низкооплачиваемый
8	Денис	Тула	100	Высокооплачиваемый
9	Ирина	Тула	85	Низкооплачиваемый

image-71.png

## Task 4

Для каждого сотрудника вывести его опыт работы (разница между now() и его hiring date), опыт работы сотрудника, получающего максимальную зарплату по отделу, и максимальную зарплату по отделу.

Вывод должен быть в формате:

- имя;
- город;
- отдел;
- зарплата;
- опыт работы (количество лет, месяцев и дней, отработанных сотрудником);
- опыт работы человека, получающего максимальную зарплату по отделу;
- максимальная зарплата по отделу.

Указание\_: для расчета опыта работы используйте функцию AGE.

```
WITH max_salary_per_department AS (
    SELECT
        department,
        MAX(salary) AS max_salary
    FROM employees
    GROUP BY department
),
max_salary_employee AS (
    SELECT
        e.id,
        e.department,
        AGE(e.hiring_date) AS max_salary_experience
    FROM employees e
    JOIN max_salary_per_department msd
        ON e.department = msd.department
        AND e.salary = msd.max_salary
)
```

```

SELECT
    e.name,
    e.city,
    e.department,
    e.salary,
    AGE(e.hiring_date) AS experience,
    mse.max_salary_experience,
    msd.max_salary
FROM employees e
-- джойним cte для максимальной зарплаты по отделу
JOIN max_salary_per_department msd
    ON e.department = msd.department
-- джойним cte для опыта работы сотрудника с максимальной зарплатой
LEFT JOIN max_salary_employee mse
    ON e.department = mse.department
ORDER BY e.department, e.name;

```

## Result:

	name character varying (50)	city character varying (50)	department character varying (50)	salary integer	experience interval	max_salary_experience interval	max_salary integer
1	Алина	Казань	hr	90	3 years 7 mons 24 days	3 years 7 mons 24 days	90
2	Борис	Самара	hr	78	1 year 6 mons 7 days	3 years 7 mons 24 days	90
3	Дарья	Самара	hr	70	2 years 4 mons 9 days	3 years 7 mons 24 days	90
4	Ирина	Тула	hr	85	4 years 3 mons 4 days	3 years 7 mons 24 days	90
5	Михаил	Москва	hr	65	3 mons 2 days	3 years 7 mons 24 days	90
6	Сергей	Воронеж	hr	88	4 years 1 mon 15 days	3 years 7 mons 24 days	90
7	Ангелина	Воронеж	it	96	2 years 3 mons 28 days	4 years 9 mons 16 days	125
8	Антон	Москва	it	125	4 years 9 mons 16 days	4 years 9 mons 16 days	125
9	Денис	Тула	it	100	1 year 11 mons 21 days	4 years 9 mons 16 days	125
10	Елена	Самара	it	84	1 year 1 mon 1 day	4 years 9 mons 16 days	125
11	Иван	Москва	it	120	4 years 6 mons 4 days	4 years 9 mons 16 days	125
12	Ксения	Москва	it	90	10 mons 9 days	4 years 9 mons 16 days	125
13	Леонид	Самара	it	104	1 year 11 mons 23 days	4 years 9 mons 16 days	125
14	Марина	Москва	it	104	2 years 8 mons 4 days	4 years 9 mons 16 days	125
15	Анна	Москва	sales	100	3 years 8 mons 7 days	3 years 8 mons 7 days	100
16	Вероника	Москва	sales	96	3 years 3 mons 26 days	3 years 8 mons 7 days	100
17	Евгений	Самара	sales	64	7 mons 25 days	3 years 8 mons 7 days	100
18	Константин	Тула	sales	85	4 years 4 mons 25 days	3 years 8 mons 7 days	100
19	Максим	Самара	sales	70	2 years 3 mons 6 days	3 years 8 mons 7 days	100
20	Оксана	Москва	sales	66	1 year 1 mon 27 days	3 years 8 mons 7 days	100
Total rows: 20 of 20    Query complete 00:00:00.145    Ln 1, Col 1							

image-72.png

## Task 5

Добавить дополнительный номер телефона (89997778834) сотруднику с именем «Денис», а также поменять его ник в telegram на «@already~~senior~~».

*Вывести сотрудников, у которых номер телефона заканчивается на «34».*

*Вывод должен быть в формате:*

— имя;

— все номера телефонов сотрудника.

Указание\_: для разворачивания массива JSON в набор значений text используйте функцию jsonb\_array\_elements\_text.



```

-- добавляем номер Денису
UPDATE employees
SET contacts = jsonb_set(
    jsonb_set(contacts, '{phone}',
        COALESCE(contacts->'phone', '[]'::jsonb) || '["89997778834"]'::jsonb
    ),
    '{tg}', '"@already_senior"'
)
WHERE name = 'Денис';

--проверяем, смотрим номера
SELECT
    e.name,
    jsonb_agg(phone) AS phone_numbers
FROM employees e,
    LATERAL jsonb_array_elements_text(e.contacts->'phone') AS phone
WHERE phone LIKE '%34'
GROUP BY e.name;

```

## Result:

	name character varying (50) 	phone_numbers jsonb 
1	Борис	["89010973434"]
2	Ангелина	["89608971234"]
3	Антон	["89407625434"]
4	Марина	["89650957834"]
5	Ксения	["89994512334"]
6	Денис	["89997778834"]
7	Елена	["89340972334"]

*image-73.png*