# Задание

1. Определить, во сколько раз зарплата каждого сотрудника меньше максимальной зарплаты по компании.
2. Определить, во сколько раз зарплата сотрудника отличается от средней зарплаты по департаменту.
3. Вывести список всех сотрудников. Для каждого сотрудника вывести среднюю зарплату по департаменту и среднюю зарплату по должности. Определить, во сколько раз средняя зарплата по департаменту отличается от средней зарплаты по должности.
4. Вывести список сотрудников, получающих минимальную зарплату по департаменту. Если в каком-либо департаменте несколько сотрудников получают минимальную зарплату, вывести того, чья фамилия идет раньше по алфавиту.
5. На основе таблицы employees создать таблицу scores c результатами соревнований со следующим маппингом: employee\_id -> man\_id, department\_id -> division, salary -> score. Вывести список людей, занявших первые 3 места в каждом дивизионе (т.е. занявших три позиции с максимальным количеством очков).
6. Отсортировать список сотрудников по фамилиям и разбить на 5 по возможности равных групп. Для каждого сотрудника вывести разницу между его зарплатой и средней зарплатой по группе.
7. Для каждого сотрудника посчитать количество сотрудников, принятых на работу в период ± 1 год от даты его принятия на работу, а также количество сотрудников, принятых позже данного сотрудника, но в этом же году. Если два сотрудника приняты в один день, считать принятым позже сотрудника с б*о*льшим employee\_id.

# Решение

1. SELECT employee\_id, salary,  
   ((MAX(salary) OVER ())/salary) this\_times\_smaller\_than\_max  
   FROM employees;
2. SELECT employee\_id,department\_id, salary,

((AVG(salary) OVER (PARTITION BY department\_id))/salary) this\_times\_smaller\_than\_dept\_avg

FROM employees;

1. SELECT employee\_id,department\_id, job\_id,

AVG(salary) OVER (PARTITION BY department\_id) avg\_dept,

AVG(salary) OVER (PARTITION BY job\_id) avg\_job,

(AVG(salary) OVER (PARTITION BY department\_id))/(AVG(salary) OVER (PARTITION BY job\_id)) dept\_divide\_by\_job

FROM employees;

1. SELECT department\_id, first\_name, last\_name, salary

FROM (SELECT department\_id,

first\_name,

last\_name,

salary,

DENSE\_RANK() OVER (PARTITION BY department\_id ORDER BY salary, last\_name) dr

FROM employees)

WHERE dr = 1

ORDER BY department\_id;

1. CREATE TABLE scores  
   (  
    man\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
    division INTEGER NOT NULL,  
    score double NOT NULL  
   );  
     
   INSERT INTO scores (man\_id, division, score)  
   SELECT employee\_id, department\_id, salary FROM employees;  
     
   SELECT division, man\_id, score FROM  
   (SELECT division, man\_id, score,  
   DENSE\_RANK() OVER (PARTITION BY division ORDER BY score DESC) dr  
   FROM scores)  
   WHERE dr <= 3  
   ORDER BY division, score DESC;

Или через создание таблицы как view

WITH scores(man\_id, division, score) AS (

SELECT employee\_id, department\_id, salary

FROM employees)

SELECT division, man\_id, score

FROM (SELECT division,

man\_id,

score,

DENSE\_RANK() OVER (PARTITION BY division ORDER BY score DESC) dr

FROM scores)

WHERE dr <= 3

ORDER BY division, score DESC;

1. SELECT employee\_id, last\_name, salary, avg\_salary, salary - avg\_salary

FROM (

SELECT employee\_id, last\_name, NTILE(5) OVER (ORDER BY last\_name) as bucket1, salary FROM employees

)

JOIN (

SELECT bucket2, AVG(salary) as avg\_salary

FROM (

SELECT NTILE(5) OVER (ORDER BY last\_name) as bucket2, salary FROM employees

)

GROUP BY bucket2

) ON bucket1 = bucket2 ORDER BY last\_name;

1. SELECT employee\_id,  
    (COUNT(\*)  
    OVER (ORDER BY JULIANDAY(hire\_date) RANGE BETWEEN 365 PRECEDING AND 365 FOLLOWING)) as '+-year',  
    (COUNT(\*)  
    OVER (PARTITION BY STRFTIME('%Y', hire\_date) ORDER BY JULIANDAY(hire\_date),employee\_id RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING EXCLUDE CURRENT ROW )) as same\_year\_later  
     
   FROM employees  
   ORDER BY hire\_date, employee\_id;