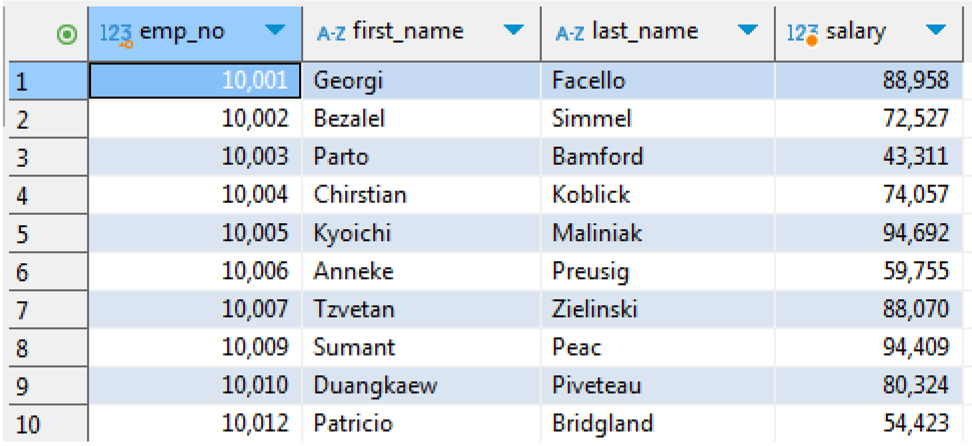
Employee Data Analysis with SQL

1. Join of the "employees" and "salaries" tables.

Retrieve a list of 10 employees, including their `emp_no`, `first_name`, `last_name`, and their current salary.

```
SELECT emp.emp_no,
       first_name,
       last_name,
       salary
FROM employees emp
JOIN salaries s ON emp.emp_no = s.emp_no
WHERE s.to_date = '9999-01-01'
LIMIT 10;
```

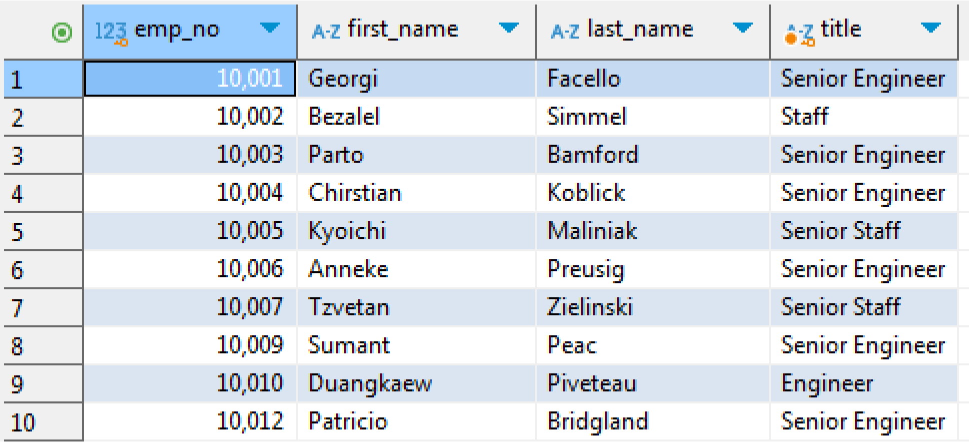


	emp_no	first_name	last_name	salary
1	10,001	Georgi	Facello	88,958
2	10,002	Bezalel	Simmel	72,527
3	10,003	Parto	Bamford	43,311
4	10,004	Chirstian	Koblick	74,057
5	10,005	Kyoichi	Maliniak	94,692
6	10,006	Anneke	Preusig	59,755
7	10,007	Tzvetan	Zielinski	88,070
8	10,009	Sumant	Peac	94,409
9	10,010	Duangkaew	Piveteau	80,324
10	10,012	Patricio	Bridgland	54,423

2. Join of the "employees" and "titles" tables.

Retrieve a list of 10 employees, including their `emp_no`, `first_name`, `last_name`, and their current job title.

```
SELECT emp.emp_no,

       emp.first_name,

       emp.last_name,

       t.title

FROM employees emp

JOIN titles t ON emp.emp_no = t.emp_no

WHERE t.to_date = '9999-01-01'

LIMIT 10;
```

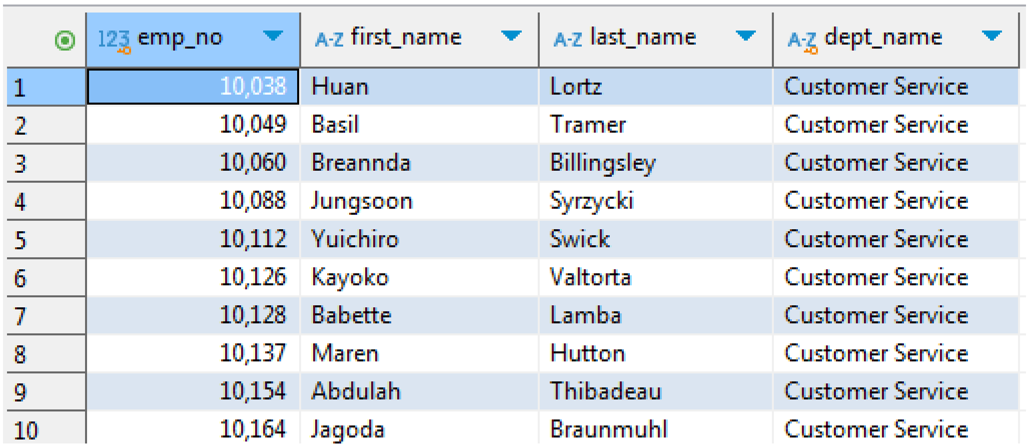


	emp_no	first_name	last_name	title
1	10,001	Georgi	Facello	Senior Engineer
2	10,002	Bezalel	Simmel	Staff
3	10,003	Parto	Bamford	Senior Engineer
4	10,004	Chirstian	Koblick	Senior Engineer
5	10,005	Kyoichi	Maliniak	Senior Staff
6	10,006	Anneke	Preusig	Senior Engineer
7	10,007	Tzvetan	Zielinski	Senior Staff
8	10,009	Sumant	Peac	Senior Engineer
9	10,010	Duangkaew	Piveteau	Engineer
10	10,012	Patricio	Bridgland	Senior Engineer

3. Join the "employees", "dept_emp", and "departments" tables.

Retrieve a list of 10 employees, including their `emp_no`, `first_name`, `last_name`, and the department they are currently working in.

```
SELECT emp.emp_no,
    emp.first_name,
    emp.last_name,
    d.dept_name
FROM employees emp
JOIN dept_emp de ON emp.emp_no = de.emp_no
JOIN departments d ON de.dept_no = d.dept_no
WHERE de.to_date = '9999-01-01'
LIMIT 10;
```



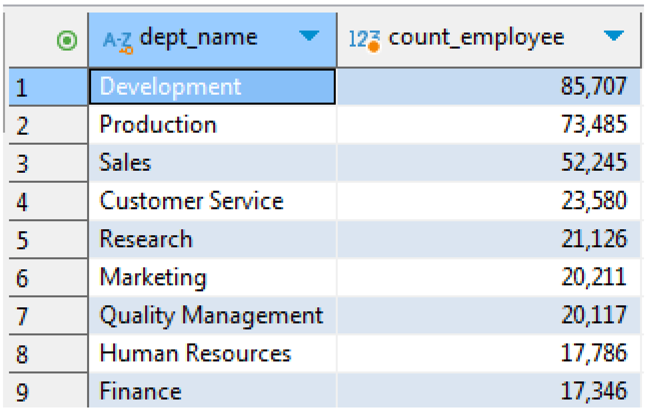
	emp_no	first_name	last_name	dept_name
1	10,038	Huan	Lortz	Customer Service
2	10,049	Basil	Tramer	Customer Service
3	10,060	Breannda	Billingsley	Customer Service
4	10,088	Jungsoon	Syrzycki	Customer Service
5	10,112	Yuichiro	Swick	Customer Service
6	10,126	Kayoko	Valtorta	Customer Service
7	10,128	Babette	Lamba	Customer Service
8	10,137	Maren	Hutton	Customer Service
9	10,154	Abdulah	Thibadeau	Customer Service
10	10,164	Jagoda	Braunmuhl	Customer Service

4. The company management wants a clear understanding of the distribution of employees across different departments.

Specifically, the task is to:

- **Determine the number of unique employees in each department.**
- **Present this information in a sorted order, starting with the department having the highest number of employees.**

```
SELECT d.dept_name,
       COUNT(DISTINCT de.emp_no) AS count_employee
FROM dept_emp de
JOIN departments d ON de.dept_no = d.dept_no
GROUP BY d.dept_name
ORDER BY count_employee DESC;
```

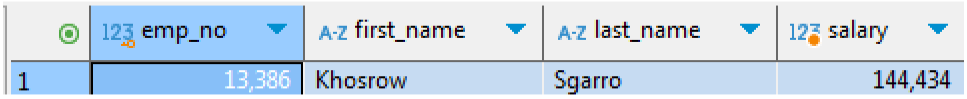


	dept_name	count_employee
1	Development	85,707
2	Production	73,485
3	Sales	52,245
4	Customer Service	23,580
5	Research	21,126
6	Marketing	20,211
7	Quality Management	20,117
8	Human Resources	17,786
9	Finance	17,346

5. Find the highest-paid employee in the Development department.

You need to determine their employee ID, first name, last name, and salary.

```
SELECT emp.emp_no,
       emp.first_name,
       emp.last_name,
       s.salary
FROM employees emp
JOIN salaries s ON emp.emp_no = s.emp_no
JOIN dept_emp de ON emp.emp_no = de.emp_no
JOIN departments d ON de.dept_no = d.dept_no
WHERE d.dept_name = 'Development'
ORDER BY s.salary DESC
LIMIT 1;
```

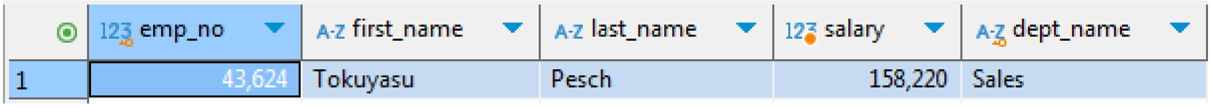


	emp_no	first_name	last_name	salary
1	13,386	Khosrow	Sgarro	144,434

6. Determine the department where the highest-paid employee in the company works.

Retrieve their employee ID, first name, last name, salary, and department.

```
SELECT emp.emp_no,
       emp.first_name,
       emp.last_name,
       s.salary,
       d.dept_name
FROM employees emp
JOIN salaries s ON emp.emp_no = s.emp_no
JOIN dept_emp de ON emp.emp_no = de.emp_no
JOIN departments d ON de.dept_no = d.dept_no
ORDER BY s.salary DESC
LIMIT 1;
```

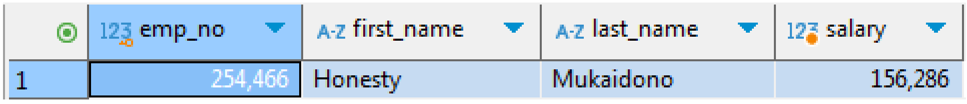


	emp_no	first_name	last_name	salary	dept_name
1	43,624	Tokuyasu	Pesch	158,220	Sales

7. Retrieve the employee ID, first name, last name, and salary of the employee with the third highest salary (i.e., if we ranked employees by descending salary, this employee would be third in the list).

Only one row of data should be returned.

```
SELECT emp.emp_no,
       emp.first_name,
       emp.last_name,
       s.salary
FROM employees emp
JOIN salaries s ON emp.emp_no = s.emp_no
ORDER BY s.salary DESC
LIMIT 2, 1
;
```

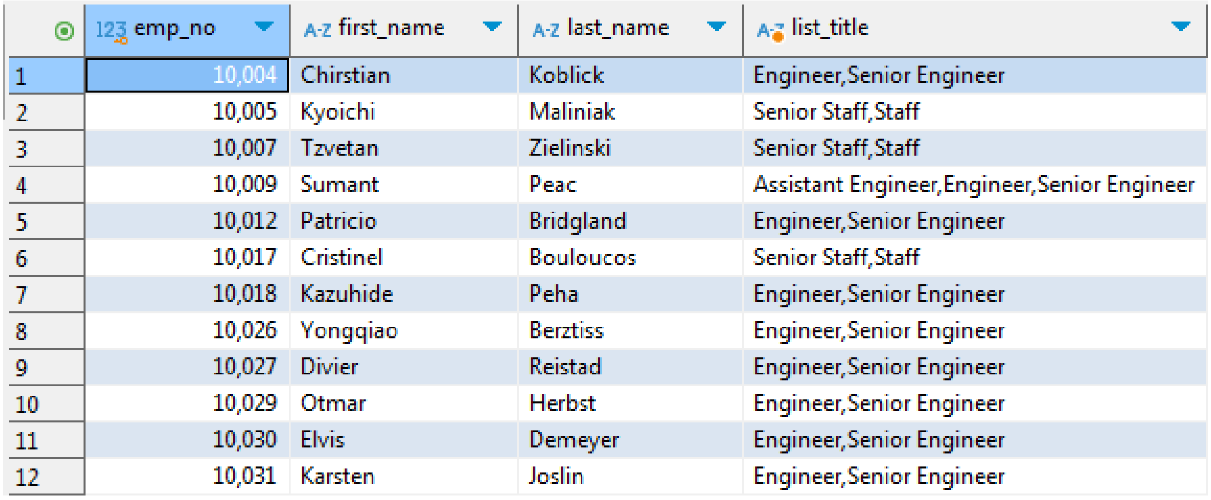


	emp_no	first_name	last_name	salary
1	254,466	Honesty	Mukaidono	156,286

8. Find employees with multiple job titles.

Some employees may change their job titles during their time at the company. Identify all employees who have had more than one job title during their employment. Display their `emp_no`, `first_name`, `last_name`, and a list of all the job titles they have held.

```
SELECT emp.emp_no,
       emp.first_name,
       emp.last_name,
       GROUP_CONCAT(t.title) AS list_title
FROM employees emp
JOIN titles t ON emp.emp_no = t.emp_no
GROUP BY emp.emp_no,
         emp.first_name,
         emp.last_name
HAVING COUNT(t.title) > 1;
```

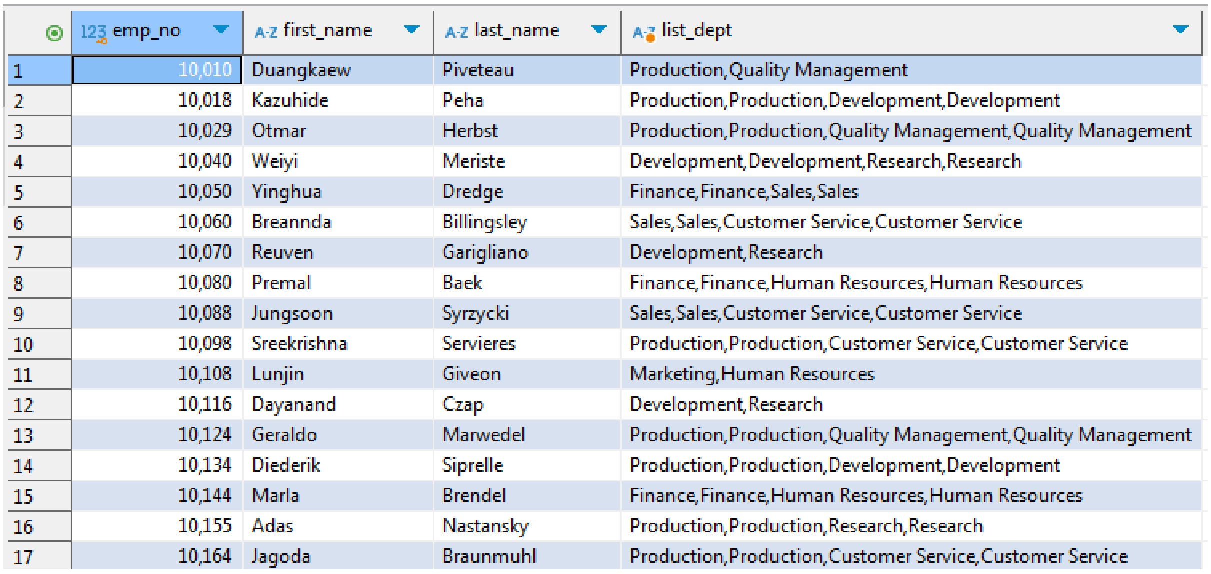


	emp_no	first_name	last_name	list_title
1	10,004	Chirstian	Koblick	Engineer,Senior Engineer
2	10,005	Kyoichi	Maliniak	Senior Staff,Staff
3	10,007	Tzvetan	Zielinski	Senior Staff,Staff
4	10,009	Sumant	Peac	Assistant Engineer,Engineer,Senior Engineer
5	10,012	Patricio	Bridgland	Engineer,Senior Engineer
6	10,017	Cristinel	Bouloucos	Senior Staff,Staff
7	10,018	Kazuhide	Peha	Engineer,Senior Engineer
8	10,026	Yongqiao	Berztiss	Engineer,Senior Engineer
9	10,027	Divier	Reistad	Engineer,Senior Engineer
10	10,029	Otmar	Herbst	Engineer,Senior Engineer
11	10,030	Elvis	Demeyer	Engineer,Senior Engineer
12	10,031	Karsten	Joslin	Engineer,Senior Engineer

9. Employee department transfers.

Employees sometimes move between departments. Identify all employees who have worked in more than one department. Display their `emp_no`, `first_name`, `last_name`, and the names of all the departments they have worked in.

```
SELECT emp.emp_no,
       emp.first_name,
       emp.last_name,
       GROUP_CONCAT(d.dept_name) AS list_dept
FROM employees emp
JOIN dept_emp de ON emp.emp_no = de.emp_no
JOIN departments d ON de.dept_no = d.dept_no
GROUP BY emp.emp_no,
         emp.first_name,
         emp.last_name
HAVING COUNT(DISTINCT d.dept_name) > 1;
```

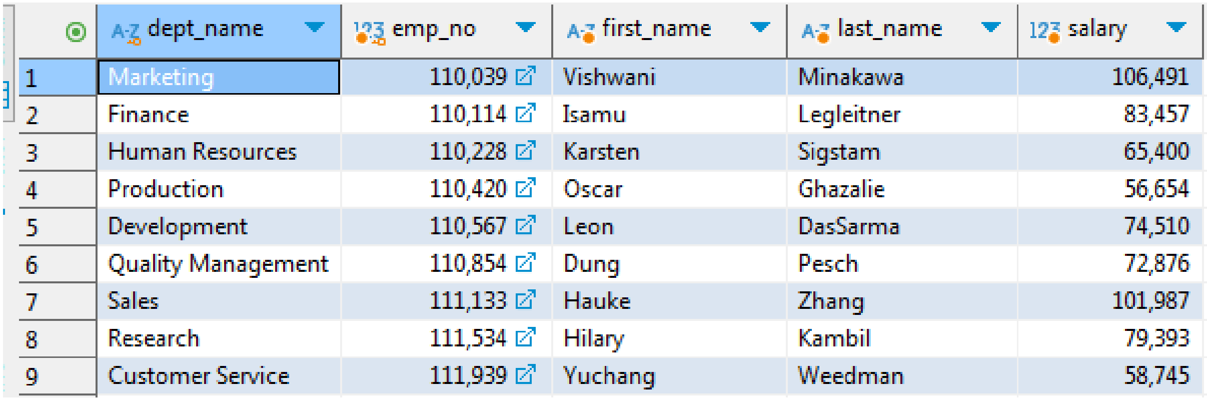


	emp_no	first_name	last_name	list_dept
1	10,010	Duangkaew	Piveteau	Production,Quality Management
2	10,018	Kazuhide	Peha	Production,Production,Development,Development
3	10,029	Otmar	Herbst	Production,Production,Quality Management,Quality Management
4	10,040	Weiyi	Meriste	Development,Development,Research,Research
5	10,050	Yinghua	Dredge	Finance,Finance,Sales,Sales
6	10,060	Breannda	Billingsley	Sales,Sales,Customer Service,Customer Service
7	10,070	Reuven	Garigliano	Development,Research
8	10,080	Premal	Baek	Finance,Finance,Human Resources,Human Resources
9	10,088	Jungsoon	Syrzycki	Sales,Sales,Customer Service,Customer Service
10	10,098	Sreekrishna	Servieres	Production,Production,Customer Service,Customer Service
11	10,108	Lunjin	Giveon	Marketing,Human Resources
12	10,116	Dayanand	Czap	Development,Research
13	10,124	Geraldo	Marwedel	Production,Production,Quality Management,Quality Management
14	10,134	Diederik	Siprelle	Production,Production,Development,Development
15	10,144	Marla	Brendel	Finance,Finance,Human Resources,Human Resources
16	10,155	Adas	Nastansky	Production,Production,Research,Research
17	10,164	Jagoda	Braunmuhl	Production,Production,Customer Service,Customer Service

10. Current salary of department managers.

For each department, display the department name, the `emp_no` of the current manager, their `first_name`, `last_name`, and current salary.

```
SELECT t.dept_name,
       t.emp_no,
       t.first_name,
       t.last_name,
       s.salary
FROM
  (SELECT d.dept_name,
          dm.emp_no,
          emp.first_name,
          emp.last_name,
          MAX(dm.from_date) AS from_date,
          MAX(dm.to_date) AS TO_DATE
   FROM departments d
   JOIN dept_manager dm ON d.dept_no = dm.dept_no
   JOIN employees emp ON dm.emp_no = emp.emp_no
   GROUP BY d.dept_name,
            dm.emp_no,
            emp.first_name,
            emp.last_name
   HAVING MAX(dm.to_date) = '9999-01-01') t
JOIN salaries s ON t.emp_no = s.emp_no
AND s.to_date = '9999-01-01'
WHERE s.from_date =
    (SELECT MAX(from_date)
     FROM salaries
     WHERE emp_no = t.emp_no
       AND TO_DATE = '9999-01-01');
```



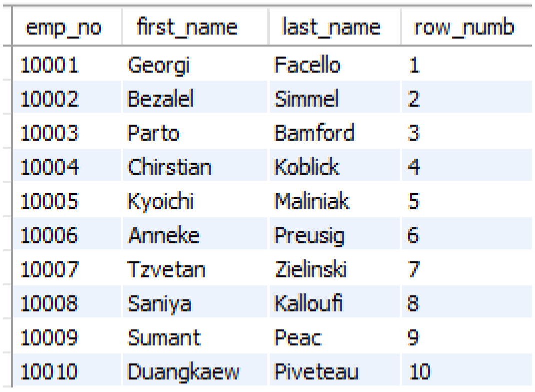
	dept_name	emp_no	first_name	last_name	salary
1	Marketing	110,039	Vishwani	Minakawa	106,491
2	Finance	110,114	Isamu	Legleitner	83,457
3	Human Resources	110,228	Karsten	Sigstam	65,400
4	Production	110,420	Oscar	Ghazalie	56,654
5	Development	110,567	Leon	DasSarma	74,510
6	Quality Management	110,854	Dung	Pesch	72,876
7	Sales	111,133	Hauke	Zhang	101,987
8	Research	111,534	Hilary	Kambil	79,393
9	Customer Service	111,939	Yuchang	Weedman	58,745

Task using Window Functions

1. Using ROW_NUMBER.

For the first 10 employees (sorted by `emp_no`), display their `emp_no`, `first_name`, `last_name`, and their row number in the table.

```
SELECT emp_no,
       first_name,
       last_name,
       ROW_NUMBER() OVER (
                   ORDER BY emp_no) AS row_numb
FROM employees
LIMIT 10;
```

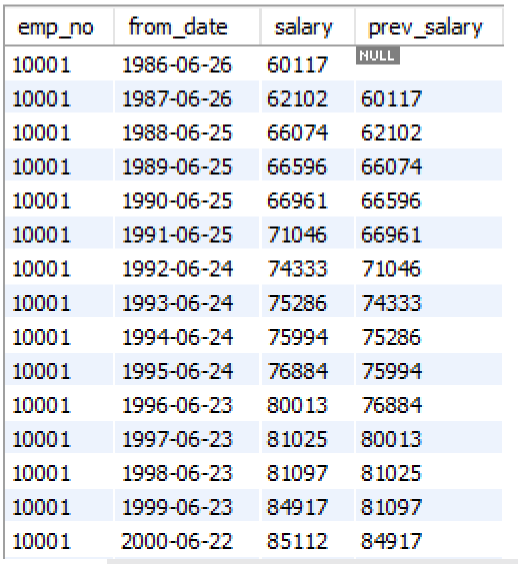


emp_no	first_name	last_name	row_numb
10001	Georgi	Facello	1
10002	Bezalel	Simmel	2
10003	Parto	Bamford	3
10004	Chirstian	Koblick	4
10005	Kyoichi	Maliniak	5
10006	Anneke	Preusig	6
10007	Tzvetan	Zielinski	7
10008	Saniya	Kalloufi	8
10009	Sumant	Peac	9
10010	Duangkaew	Piveteau	10

2. Display the salary history of an employee.

Retrieve `emp_no`, `from_date`, salary, and the previous salary value for the employee with `emp_no = 10001`.

```
SELECT emp_no,
       from_date,
       salary,
       LAG(salary) OVER (PARTITION BY emp_no
                  ORDER BY TO_DATE) AS prev_salary
FROM salaries
WHERE emp_no = '10001';
```

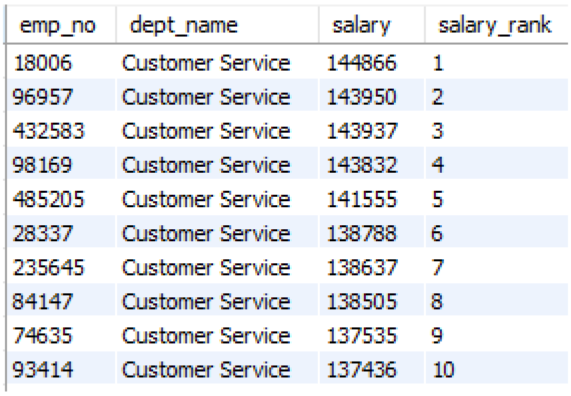


emp_no	from_date	salary	prev_salary
10001	1986-06-26	60117	NULL
10001	1987-06-26	62102	60117
10001	1988-06-25	66074	62102
10001	1989-06-25	66596	66074
10001	1990-06-25	66961	66596
10001	1991-06-25	71046	66961
10001	1992-06-24	74333	71046
10001	1993-06-24	75286	74333
10001	1994-06-24	75994	75286
10001	1995-06-24	76884	75994
10001	1996-06-23	80013	76884
10001	1997-06-23	81025	80013
10001	1998-06-23	81097	81025
10001	1999-06-23	84917	81097
10001	2000-06-22	85112	84917

3. Determine salary rank within the department.

Retrieve emp_no, dept_name, current salary, and the rank of the employee within their department based on salary.

```
SELECT s.emp_no,
       d.dept_name,
       s.salary,
       RANK() OVER (PARTITION BY d.dept_name
                    ORDER BY s.salary DESC) AS salary_rank
FROM salaries s
JOIN dept_emp de ON s.emp_no = de.emp_no
JOIN departments d ON de.dept_no = d.dept_no
WHERE s.to_date = '9999-01-01';
```

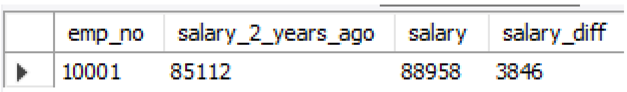


emp_no	dept_name	salary	salary_rank
18006	Customer Service	144866	1
96957	Customer Service	143950	2
432583	Customer Service	143937	3
98169	Customer Service	143832	4
485205	Customer Service	141555	5
28337	Customer Service	138788	6
235645	Customer Service	138637	7
84147	Customer Service	138505	8
74635	Customer Service	137535	9
93414	Customer Service	137436	10

4. Find the salary change of an employee over the last 2 years.

Retrieve the difference between the current salary of the employee with emp_no = 10001 and their salary from 2 years ago. The result should include: employee number, salary 2 years ago, current salary, and the difference in salary between 2 years ago and now.

```
SELECT emp_no,
       LAG(salary, 2) OVER (
                      ORDER BY from_date) AS salary_2_years_ago,
       salary,
       salary - LAG(salary, 2) OVER (
                         ORDER BY from_date) AS salary_diff
FROM salaries
WHERE emp_no = '10001'
ORDER BY from_date DESC
LIMIT 1;
```



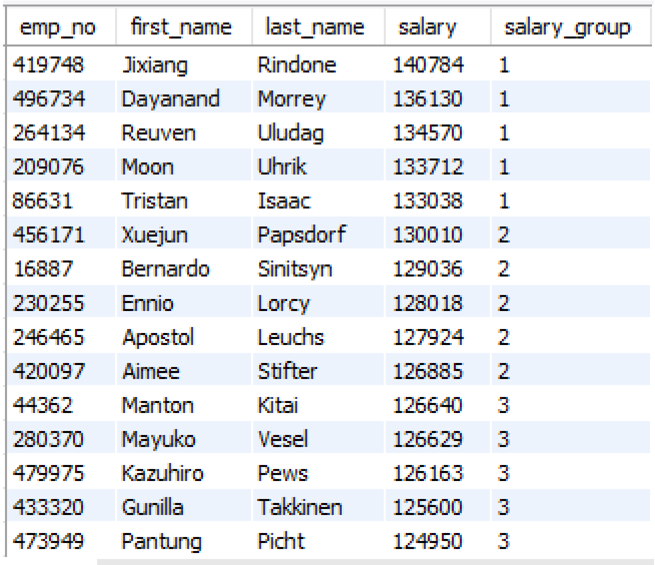
	emp_no	salary_2_years_ago	salary	salary_diff
▶	10001	85112	88958	3846

5. Use NTILE.

Distribute the specified employees into 3 groups based on their current salary. Display their `emp_no`, `first_name`, `last_name`, salary, and the group number for each.

The employee `emp_no` values to be analyzed are: `419748, 496734, 264134, 209076, 86631, 456171, 16887, 230255, 246465, 420097, 44362, 280370, 479975, 433320, 473949`.

```
SELECT e.emp_no,
       e.first_name,
       e.last_name,
       s.salary,
       NTILE(3) OVER (
               ORDER BY s.salary DESC) AS salary_group
FROM employees e
JOIN salaries s ON e.emp_no = s.emp_no
WHERE e.emp_no IN (419748,
               496734,
               264134,
               209076,
               86631,
               456171,
               16887,
               230255,
               246465,
               420097,
               44362,
               280370,
               479975,
               433320,
               473949)
  AND s.to_date = '9999-01-01'
ORDER BY salary_group,
        s.salary DESC;
```



emp_no	first_name	last_name	salary	salary_group
419748	Jixiang	Rindone	140784	1
496734	Dayanand	Morrey	136130	1
264134	Reuven	Uludag	134570	1
209076	Moon	Uhrik	133712	1
86631	Tristan	Isaac	133038	1
456171	Xuejun	Papsdorf	130010	2
16887	Bernardo	Sinitsyn	129036	2
230255	Ennio	Lorcy	128018	2
246465	Apostol	Leuchs	127924	2
420097	Aimee	Stifter	126885	2
44362	Manton	Kitai	126640	3
280370	Mayuko	Vesel	126629	3
479975	Kazuhiro	Pews	126163	3
433320	Gunilla	Takkinen	125600	3
473949	Pantung	Picht	124950	3