



SQL-проєкт: Аналіз бази даних "Employees"

Побудова запитів у DBeaver для вивчення структури, зарплат та працівників з використанням бази даних employees.

Мета:

Відпрацювати основні SQL-запити та зчитувати актуальні зарплати та базову інформацію про працівників з бази даних Employees за допомогою SQL-запитів.

База даних:

Інформація про базу - [за посиланням](#)

1

SQL Аналіз: Поточні зарплати працівників

Виведіть список з 10 співробітників, їх `emp_no`, `first_name`, `last_name` та їхню **поточну зарплату**.

```

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

	123 ↪ emp_no	A-Z first_name	A-Z last_name	123 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	Patricia	Bridgland	54,423

Примітка: Значення '9999-01-01' позначає поточну дату зарплати в базі

2

SQL Аналіз: Поточні зарплати працівників

Виведіть список з 10 співробітників, їх `emp_no`, `first_name`, `last_name` та їхню **поточну посаду**.

```

SELECT
    e.emp_no,
    e.first_name,
    e.last_name,
    t.title
FROM
    employees e
JOIN titles t ON
    e.emp_no = t.emp_no
WHERE
    t.to_date = '9999-01-01'
LIMIT 10;

```

	123 ↘ emp_no	A-Z first_name	A-Z last_name	A-Z 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	Patricia	Bridgland	Senior Engineer

Примітка:

Значення '**9999-01-01**' позначає поточну дату на посаді

3

SQL Аналіз: Відділи в яких працють робітники

Виведіть список з 10 співробітників, їх `emp_no`, `first_name`, `last_name` та `відділ`, де вони зараз працюють.

```

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

```

	123 ↕ emp_no ↑ ↓	A-Z first_name ↓	A-Z last_name ↓	A-Z ↕ dept_name ↓
1	10,001	Georgi	Facello	Development
2	10,002	Bezalel	Simmel	Sales
3	10,003	Parto	Bamford	Production
4	10,004	Chirstian	Koblick	Production
5	10,005	Kyoichi	Maliniak	Human Resources
6	10,006	Anneke	Preusig	Development
7	10,007	Tzvetan	Zielinski	Research
8	10,009	Sumant	Peac	Quality Management
9	10,010	Duangkaew	Piveteau	Quality Management
10	10,012	Patricia	Bridgland	Development

Примітка:

Значення '**9999-01-01**' позначає поточну дату
в департаменті

SQL Аналіз: Розподілу працівників між різними відділами

- Визначити кількість унікальних працівників у кожному відділі.
- Представити цю інформацію у впорядкованому вигляді, починаючи з відділу з найбільшою кількістю працівників.

```

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

```

	A-Z ↗ dept_name	123 employee_count
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

Примітка:

DISTINCT – шукаємо унікальні значення

COUNT – рахуємо скільки унікальних

GROUP BY – групуємо по департаментам

ORDER BY – сортуємо згруповані дані

DESC – сортування за спаданням

SQL Аналіз: Високооплачуваний працівник у відділі

Знайдіть, хто є найбільш високооплачуваним працівником у відділі Development

```

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

	123 ↗ emp_no	A-Z first_name	A-Z last_name	123 salary	A-Z ↗ dept_name
1	13,386	Khosrow	Sgarro	144,434	Development

Примітка:

ORDER BY - сортуємо заробітну плату

DESC - сортування за спаданням

LIMIT 1 - залишаємо одного

SQL Аналіз: Високооплачуваний працівник у компанії

Знайдіть, хто є найбільш високооплачуваним працівником у компанії

```

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

	123 ↗ emp_no ↘	A-Z first_name ↘	A-Z last_name ↘	123 salary ↘	A-Z ↗ dept_name ↘
1	43,624	Tokuyasu	Pesch	158,220	Sales

Примітка:

ORDER BY - сортуємо заробітну плату

DESC - сортування за спаданням

LIMIT 1 - залишаємо одного

SQL Аналіз: Третій високооплачуваний працівник у компанії

Виведіть **номер, ім'я, прізвище і розмір зарплати** співробітника з третьою найвищою зарплатою. Треба вивести лише один рядок даних.

```

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

```

	123 ↗ emp_no	A-Z first_name	A-Z last_name	123 salary
1	254,466	Honesty	Mukaidono	156,286

Примітка:

ORDER BY - сортуємо заробітну плату

DESC - сортування за спаданням

offset 2 - пропускаємо перші 2 рядки

LIMIT 1 - залишаємо одного

також є 2ий варіант написати **LIMIT 2,1;**

SQL Аналіз: Співробітники, які змінювали посаду

Визначте всіх співробітників, які мали більше однієї посади за час роботи в компанії. Відобразіть їх `emp_no`, `first_name`, `last_name` та список всіх посад, на яких вони працювали.

```

SELECT
    e.emp_no,
    e.first_name,
    e.last_name,
    group_concat(t.title) AS titles_held
FROM
    employees e
JOIN titles t ON
    e.emp_no = t.emp_no
GROUP BY
    e.emp_no,
    e.first_name,
    e.last_name
HAVING
    count(DISTINCT t.title) > 1;
  
```

123 ↗ emp_no	A-Z first_name	A-Z last_name	A-Z titles_held
10,004	Chirstian	Koblick	Engineer,Senior Engineer
10,005	Kyoichi	Maliniak	Senior Staff,Staff
10,007	Tzvetan	Zielinski	Senior Staff,Staff
10,009	Sumant	Peac	Assistant Engineer,Engineer,Senior Engineer
10,012	Patricio	Bridgland	Engineer,Senior Engineer
10,017	Cristinel	Bouloucos	Senior Staff,Staff
10,018	Kazuhide	Peha	Engineer,Senior Engineer
10,026	Yongqiao	Berztiss	Engineer,Senior Engineer
10,027	Divier	Reistad	Engineer,Senior Engineer
10,029	Otmar	Herbst	Engineer,Senior Engineer
10,030	Elvis	Demeyer	Engineer,Senior Engineer
10,031	Karsten	Joslin	Engineer,Senior Engineer
10,032	Jeong	Reistad	Engineer,Senior Engineer
10,035	Alain	Chappelet	Engineer,Senior Engineer
10,037	Pradeep	Makrucki	Engineer,Senior Engineer
10,038	Huan	Lortz	Senior Staff,Staff

Примітка:

`count(DISTINCT)` – рахує унікальні посади

`group_concat` – сконкатенувати декілька посад

SQL Аналіз: Переводи між відділами

Визначте всіх співробітників, які працювали в більш ніж одному відділі.
 Відобразіть їх **emp_no**, **first_name**, **last_name** та **назви всіх відділів**, в яких вони працювали.

```

SELECT
    e.emp_no,
    e.first_name,
    e.last_name,
    group_concat(t.title) AS titles_held
FROM
    employees e
JOIN
    titles t ON
        e.emp_no = t.emp_no
GROUP BY
    e.emp_no,
    e.first_name,
    e.last_name
HAVING
    count(DISTINCT t.title) > 1;
    
```

123 ↗ emp_no	A-Z first_name	A-Z last_name	A-Z departament_held
10,010	Duangkaew	Piveteau	Quality Management,Production
10,018	Kazuhide	Peha	Production,Development
10,029	Otmar	Herbst	Production,Quality Management
10,040	Weiyi	Meriste	Research,Development
10,050	Yinghua	Dredge	Finance,Sales
10,060	Breannda	Billingsley	Customer Service,Sales
10,070	Reuven	Gariglano	Development,Research
10,080	Premal	Baek	Finance,Human Resources
10,088	Jungsoon	Syrzycki	Sales,Customer Service
10,098	Sreekrishna	Servieres	Production,Customer Service
10,108	Lunjin	Giveon	Marketing,Human Resources

Примітка:

count(DISTINCT) - рахує унікальні відділи

group_concat - сконкатенувати декілька відділів

SQL Аналіз: Поточна зарплата керівника

Для кожного відділу вкажіть назву відділу, **emp_no** поточного керівника, його **first_name**, **last_name** та поточну зарплату.

```

SELECT
    d.dept_name,
    e.emp_no,
    e.first_name,
    e.last_name,
    s.salary
FROM
    employees e
JOIN salaries s ON
    s.emp_no = e.emp_no
JOIN dept_manager de ON
    de.emp_no = e.emp_no
JOIN departments d ON
    d.dept_no = de.dept_no
WHERE
    s.to_date = '9999-01-01'
    AND de.to_date = '9999-01-01';
  
```

	A-Z ▾ dept_name ▾	123 ▾ emp_no ▾	A-Z first_name ▾	A-Z last_name ▾	123 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

Примітка:

s.to_date - вибираємо актуальну зарплату

de.to_date - вибираємо поточного керівника

'9999-01-01' - умовне позначення для актуальної дати

Використання ROW_NUMBER

Для перших 10 співробітників виведіть їх `emp_no`, `first_name`, `last_name`, а також **порядковий номер** співробітника в таблиці.

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

	emp_no	first_name	last_name	row_num
1	10,001	Georgi	Facello	1
2	10,002	Bezalel	Simmel	2
3	10,003	Parto	Bamford	3
4	10,004	Chirstian	Koblick	4
5	10,005	Kyoichi	Maliniak	5
6	10,006	Anneke	Preusig	6
7	10,007	Tzvetan	Zielinski	7
8	10,008	Saniya	Kalloufi	8
9	10,009	Sumant	Peac	9
10	10,010	Duangkaew	Piveteau	10

12

Вивести зарплатну динаміку співробітника

Виведіть **emp_no**, **дату початку** (**from_date**), **зарплату** та **попереднє значення зарплати** для співробітника з **emp_no = 10001**.

```

SELECT
    emp_no,
    from_date,
    salary,
    LAG(salary) OVER(
        ORDER BY
        from_date) AS previous_salary
FROM
    salaries
WHERE
    emp_no = 10001;
  
```

	123 🔍 emp_no	⌚ 🔍 from_date	123 salary	123 previous_salary
1	10,001	1986-06-26	60,117	[NULL]
2	10,001	1987-06-26	62,102	60,117
3	10,001	1988-06-25	66,074	62,102
4	10,001	1989-06-25	66,596	66,074
5	10,001	1990-06-25	66,961	66,596
6	10,001	1991-06-25	71,046	66,961
7	10,001	1992-06-24	74,333	71,046
8	10,001	1993-06-24	75,286	74,333
9	10,001	1994-06-24	75,994	75,286

Визначення рангу зарплати в межах відділу

Виведіть **emp_no**, **dept_name**, поточну зарплату та ранг співробітника в межах його відділу за розміром зарплати.

```

SELECT
    e.emp_no,
    d.dept_name,
    s.salary,
    RANK() OVER (PARTITION BY de.dept_no
ORDER BY
    salary DESC) AS salary_rank
FROM
    employees e
JOIN salaries s ON
    s.emp_no = e.emp_no
JOIN dept_emp de ON
    e.emp_no = de.emp_no
JOIN departments d ON
    d.dept_no = de.dept_no
WHERE
    s.to_date = '9999-01-01'
LIMIT 10;

```

	123 ↗ emp_no	A-Z ↗ dept_name	123 salary	123 salary_rank
1	466,852	Marketing	145,128	1
2	89,825	Marketing	143,644	2
3	107,140	Marketing	142,506	3
4	80,937	Marketing	142,485	4
5	51,531	Marketing	141,842	5
6	102,078	Marketing	140,974	6
7	477,358	Marketing	140,077	7
8	278,498	Marketing	138,755	8
9	235,645	Marketing	138,637	9
10	475,041	Marketing	137,413	10

Примітка:

PARTITION BY - ділить по департаментам
 Rank() OVER() - розбиває на ранги

14

Зміна зарплат протягом 2 років

Виведіть різницю між поточною зарплатою співробітника з `emp_no = 10001` та його зарплатою 2 роки тому. Формат відповіді: **номер співробітника, salary_2_years_ago, поточна ЗП, різниця у зарплатах 2 роки тому і зараз.**

```

SELECT
    s.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_difference
FROM
    salaries s
WHERE
    s.emp_no = 10001
ORDER BY
    from_date DESC
LIMIT 1;

```

	emp_no	salary_2_years_ago	salary	salary_difference
1	10,001	85,112	88,958	3,846

Примітка:

`LAG()` - Подивитися зарплату, яка була 2 рядки назад, відносно поточного рядка

Визначення найвищого зростання зарплати

```

WITH MaxDate AS (
    SELECT MAX(from_date) AS latest_date
    FROM salaries
),
SalaryDifference AS (
    SELECT
        s.emp_no,
        LAG(s.salary) OVER (PARTITION BY s.emp_no ORDER BY s.from_date) AS salary_1_years_ago,
        s.salary,
        s.salary - LAG(s.salary) OVER (PARTITION BY s.emp_no ORDER BY s.from_date) AS diff,
        s.from_date
    FROM salaries s
    JOIN MaxDate md
    ON s.from_date BETWEEN DATE_SUB(md.latest_date, INTERVAL 2 YEAR)
                        AND DATE_SUB(md.latest_date, INTERVAL 1 YEAR)
)
SELECT
    emp_no,
    salary_1_years_ago,
    salary,
    MAX(diff) AS max_difference
FROM SalaryDifference
GROUP BY emp_no, salary_1_years_ago, salary
ORDER BY max_difference DESC
LIMIT 1;

```

	emp_no	salary_1_years_ago	salary	max_difference
1	278,208	68,917	73,416	4,499

Знайдіть співробітника, який отримав найбільше збільшення зарплати протягом року, який закінчився 1 рік до найновішої дати в базі даних (максимальне значення from_date в таблиці salaries). Формат відповіді: **номер співробітника, ЗП з попереднього року, поточна ЗП, на яку змінилась ЗП.**

Примітка:

- 1) Обираємо зарплати за останній рік (WITH MaxDate)
- 2) Додаємо зарплату "рік тому" для кожного працівника (LAG() + PARTITION BY emp_no)
- 3) Рахуємо різницю і сортуємо по найбільшому зростанню
- 4) Показуємо лише 1 працівника з найбільшим ростом

Середня зарплата та перший праціник на кожній посаді

```

WITH avg_salary_per_title AS (
SELECT t.title,
       AVG(s.salary) AS avg_salary
FROM titles t
JOIN salaries s ON t.emp_no = s.emp_no
WHERE t.to_date = '9999-01-01' AND s.to_date = '9999-01-01'
GROUP BY t.title
),
first_hires AS (
  SELECT
    t.title,
    e.first_name,
    e.hire_date
  FROM titles t
  JOIN employees e ON t.emp_no = e.emp_no
  WHERE t.to_date = '9999-01-01'
),
min_hire_per_title AS (
SELECT
    title,
    min(hire_date) AS min_hire_date
  FROM first_hires
  GROUP BY t.title
)
SELECT
    fh.title,
    avg.avg_salary,
    fh.hire_date
  FROM first_hires fh
  JOIN min_hire_per_title mh
  ON fh.title = mh.title AND fh.hire_date = mh.min_hire_date
  JOIN avg_salary_per_title avg
  ON fh.title = avg.title
ORDER BY
    fh.title,
    fh.first_name;

```

Виведіть **назву посади, середню зарплату** співробітників на ній, **ім'я першого співробітника** (або співробітників, якщо в один і той самий день було прийнято на цю посаду кілька людей), який був прийнятий на цю посаду (згідно дати прийому на роботу - **hire_date**), а також **hire_date** цього співробітника чи співробітників. У відповіді **відсортуйте** результати за **назвою посади і ім'ям співробітника**.

	A-Z title	123 avg_salary	hire_date
1	Assistant Engineer	57,317.5736	1985-02-02
2	Assistant Engineer	57,317.5736	1985-02-02
3	Engineer	59,602.7378	1985-02-01
4	Manager	77,723.6667	1985-01-14
5	Senior Engineer	70,823.4376	1985-02-01
6	Senior Engineer	70,823.4376	1985-02-01
7	Senior Engineer	70,823.4376	1985-02-01
8	Senior Engineer	70,823.4376	1985-02-01
9	Senior Engineer	70,823.4376	1985-02-01
10	Senior Engineer	70,823.4376	1985-02-01
11	Senior Engineer	70,823.4376	1985-02-01
12	Senior Staff	80,706.4959	1985-01-01
13	Senior Staff	80,706.4959	1985-01-01
14	Staff	67,330.6652	1985-01-01
15	Staff	67,330.6652	1985-01-01
16	Staff	67,330.6652	1985-01-01
17	Staff	67,330.6652	1985-01-01
18	Technique Leader	67,506.5903	1985-01-01
19	Technique Leader	67,506.5903	1985-01-01
20	Technique Leader	67,506.5903	1985-01-01

Використання NTILE

Розподіліть зазначених співробітників на 3 групи за розміром їх поточної зарплати. Виведіть **emp_no**, **first_name**, **last_name**, **зарплату** та **номер групи** для кожного з них.

емп_но співробітників, зарплати яких будемо досліджувати: 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 s.to_date = '9999-01-01'
    AND e.emp_no IN (
        419748, 496734, 264134, 209076, 86631,
        456171, 16887, 230255, 246465, 420097,
        44362, 280370, 479975, 433320, 473949
);

```

	123 ↗ emp_no ▾	A-Z first_name ▾	A-Z last_name ▾	123 salary ▾	123 salary_group ▾
1	419,748	Jixiang	Rindone	140,784	1
2	496,734	Dayanand	Morrey	136,130	1
3	264,134	Reuven	Uludag	134,570	1
4	209,076	Moon	Uhrik	133,712	1
5	86,631	Tristan	Isaac	133,038	1
6	456,171	Xuejun	Papsdorf	130,010	2
7	16,887	Bernardo	Sinitsyn	129,036	2
8	230,255	Ennio	Lorcy	128,018	2
9	246,465	Apostol	Leuchs	127,924	2
10	420,097	Aimee	Stifter	126,885	2
11	44,362	Manton	Kitai	126,640	3
12	280,370	Mayuko	Vesel	126,629	3
13	479,975	Kazuhiro	Pews	126,163	3
14	433,320	Gunilla	Takkinen	125,600	3
15	473,949	Pantung	Picht	124,950	3

Загальні висновки

1. Оцінка ефективності витрат на персонал

Аналіз зміни заробітної плати в динаміці дозволяє виявити, наскільки виправдані витрати компанії на того чи іншого працівника. Це дає змогу оцінити співвідношення між рівнем оплати праці та кар'єрним зростанням, продуктивністю або внутрішніми переходами між департаментами.

Цінність: Допомагає HR і фінансовому відділу приймати рішення щодо підвищень, оптимізації зарплат, перегляду бюджетів.

2. Контроль витрат на робочу силу

Порівняння середніх зарплат по департаментах і виявлення розриву між найвищими та найнижчими окладами дає можливість бізнесу краще розуміти структуру витрат на персонал. Це сприяє стратегічному плануванню фонду оплати праці.

Цінність: Дозволяє обґрунтовано планувати бюджет на персонал.

3. Аналіз внутрішньої структури та корпоративної культури

Дані показують, що значна частина працівників має тривалий досвід у компанії та часто переходить між департаментами. Це свідчить про ефективне внутрішнє середовище, яке сприяє розвитку і лояльності співробітників.

Цінність: Індикатор здорової корпоративної культури. Дає підстави не лише зберігати внутрішні практики, а й презентувати компанію як надійного роботодавця.

4. Формування програми мотивації та винагород

Виявлення працівників з великим стажем дозволяє компанії створити систему заохочень для лояльних співробітників (наприклад, бонуси за роки роботи). Це може підвищити мотивацію і зменшити плинність кадрів.

Цінність: Дає змогу вибудовувати персоналізовану систему преміювання.

5. Оцінка та балансування структури департаментів

Порівняльний аналіз кількості працівників та рівнів зарплат у різних департаментах допомагає зрозуміти, як формується структура бізнесу. Це може свідчити про пріоритети компанії або виявляти дисбаланс у навантаженні чи фінансуванні відділів.

Цінність: Дозволяє переглянути ресурсне забезпечення відділів або пріоритети бізнесу.

Отримані аналітичні інсайти на основі SQL-даних можуть бути безпосередньо використані для прийняття стратегічних HR- та фінансових рішень. Це не лише підвищує ефективність бізнесу, але й сприяє розвитку внутрішньої культури компанії.