

Лабораторна робота 9

«Групування даних»

Для завдань/розділів відмічених « » – скріншот – обов'язково. Для інших – за домовленістю із викладачем.

Завдання лабораторної роботи виконую всі

Основне завдання

1. Скільки у фірмі відділів?

```
SELECT COUNT(*) FROM dept;
```

2. Скільки у фірмі співробітників?

```
SELECT COUNT(*) FROM emp;
```

3. У скільки відділів є співробітники?

```
SELECT COUNT(deptno) FROM (SELECT deptno FROM emp GROUP BY deptno) e1;
```

4. Скільки фірми аналітиків?

```
SELECT COUNT(empno) FROM emp GROUP BY JOB HAVING JOB = 'ANALYST';
```

5. Скільки людей працює у Гонконгу?

```
SELECT count(*) FROM emp JOIN dept USING (deptno) GROUP BY loc HAVING loc = 'HONKONG';
```

6. Скільки людей у фірмі немає підлеглих (не є керівниками)?

```
SELECT COUNT(*) FROM emp WHERE empno NOT IN (SELECT mgr FROM emp GROUP BY mgr HAVING NOT mgr is NULL);
```

7. У скількох містах є відділи фірми, де працює хоча б один співробітник?

```
SELECT COUNT (*) FROM (SELECT loc FROM dept JOIN emp USING (deptno) GROUP BY loc) l1;
```

8. Напишіть запит, що виводить мінімальну, середню, максимальну та сумарну зарплату всіх працівників

```
SELECT MIN(SAL), AVG(SAL), MAX(SAL) FROM emp;
```

9. Напишіть запит, що виводить номери та ім'я всіх співробітників, дохід яких вищий за середній.

```
SELECT empno, ename FROM emp WHERE sal > (SELECT AVG(sal) FROM emp);
```

10. У кого із співробітників найбільша премія?

```
SELECT empno, ename FROM emp WHERE comm = (SELECT MAX(comm) FROM emp);
```

11. Сформуйте запит, що виводить імена керівників та середню зарплату його підлеглих за винятком підлеглих — SALESMAN-ів

```
SELECT (SELECT ename FROM emp emp2 WHERE emp2.empno = emp1.mgr),  
AVG(sal) FROM emp emp1  
GROUP BY emp1.mgr  
HAVING (NOT emp1.mgr is NULL) AND NOT ((SELECT job FROM emp emp2 WHERE  
emp2.empno = emp1.mgr) = 'SALESMAN');
```

12. Сформуйте запит, що виводить імена керівників, у яких дохід більший ніж середній дохід у його підлеглих, за винятком підлеглих — SALESMAN-ів

```
SELECT (SELECT ename FROM emp emp2 WHERE emp2.empno = emp1.mgr),  
AVG(emp1.sal) FROM emp emp1  
GROUP BY emp1.mgr  
HAVING  
    (NOT emp1.mgr is NULL) AND  
    NOT ((SELECT job FROM emp emp2 WHERE emp2.empno = emp1.mgr) =  
'SALESMAN') AND  
    (AVG(emp1.sal) < (SELECT sal FROM emp emp2 WHERE emp2.empno =  
emp1.mgr));
```

13. Назва відділу, у якому максимальна середня зарплата.

```
SELECT dname, AVG(sal) as avg_sal FROM dept JOIN emp USING (deptno)
GROUP BY dname
ORDER BY avg_sal DESC
LIMIT 1;
```

```
1 SELECT dname, AVG(sal) as avg_sal FROM dept JOIN emp USING (deptno)
2 GROUP BY dname
3 ORDER BY avg_sal DESC
4 LIMIT 1;
```

dname	avg_sal
VILLAIN	5016.6666666666666667

14. Виведете назви відділів у порядку зростання сум зарплат працівників, які працюють у них.

```
SELECT dname, SUM(sal) as sum_sal FROM dept JOIN emp USING (deptno)
GROUP BY dname
ORDER BY sum_sal ASC;
```

```
1 SELECT dname, SUM(sal) as sum_sal FROM dept JOIN emp USING (deptno)
2 GROUP BY dname
3 ORDER BY sum_sal ASC;
```

dname	sum_sal
SALES2	7350
ACCOUNTING	7700
SALES	8450
RESEARCH	11125
VILLAIN	60200

15. * Назви відділу в якому найбільше співробітників.

```
SELECT dname, COUNT(empno) as emp_count FROM dept JOIN emp USING  
(deptno)  
GROUP BY dname  
ORDER BY emp_count DESC  
LIMIT 1;
```

16. * Міста, у яких немає співробітників.

```
SELECT loc FROM dept WHERE loc NOT IN (SELECT loc as emp_count FROM  
dept JOIN emp USING (deptno) GROUP BY loc);
```

17. * У якому місті премії йде найбільше грошей? (Результат – 1 рядок)

```
SELECT loc, COALESCE(SUM(comm), 0) as comm_sum FROM dept JOIN emp  
USING (deptno)  
GROUP BY loc  
ORDER BY comm_sum DESC  
LIMIT 1;
```

18. * Начальники, у яких BCI співробітники отримують більше 2500

```
SELECT mgr FROM emp  
GROUP BY mgr  
HAVING  
    NOT mgr is NULL AND  
    MIN(sal) > 2500;
```

Ранжування (хто на 1-му місці, ...)

19. Проранжуйте відділи за кількістю співробітників.

```
SELECT dname, COUNT(empno) as emp_count FROM dept JOIN emp USING  
(deptno)  
GROUP BY dname  
ORDER BY emp_count DESC;
```

20. Створіть звіт, у якому працівники будуть пронумеровані, впорядковані за зростанням зарплати всередині відділів.

```
SELECT deptno, ename, RANK() OVER (PARTITION BY deptno ORDER BY sal  
DESC) AS n_rank  
FROM emp  
ORDER BY deptno, n_rank;
```

21. Проранжуйте співробітників з Нью-Йорку за доходом.

```
SELECT ename, sal, DENSE_RANK() OVER(ORDER BY sal DESC) AS n_rank  
FROM emp JOIN dept ON (emp.deptno = dept.deptno AND dept.loc =  
'NEW_YORK')  
ORDER BY sal DESC, n_rank;
```

22. * Виведете ім'я, відділ та зарплату для співробітників, які отримують найменшу зарплату у відділі.

```
SELECT * FROM (SELECT ename, dname, sal, (DENSE_RANK() OVER (PARTITION  
BY deptno ORDER BY sal ASC)) as sal_rank  
FROM emp JOIN dept USING (deptno)) sal_ranks WHERE sal_rank = 1;
```

23. * Для кожного міста вивідіть ім'я та посаду людини із найбільшим доходом.

```
SELECT loc, ename, money FROM (SELECT loc, ename, (sal+COALESCE(comm,  
0)) as "money", (DENSE_RANK() OVER (PARTITION BY loc ORDER BY  
sal+COALESCE(comm, 0) DESC)) as "sal_rank"  
FROM emp JOIN dept USING (deptno)) r WHERE sal_rank = 1;
```

Зведені таблиці

24. Складіть звіт, який відображає середню зарплату за кожною з посад кожного відділу.

```
SELECT dname, job, AVG(sal) FROM emp JOIN dept USING (deptno)  
GROUP BY dname, job  
ORDER BY dname;
```

```
1 SELECT dname, job, AVG(sal) FROM emp JOIN dept USING (deptno)  
2 GROUP BY dname, job  
3 ORDER BY dname;
```

dname	job	avg
ACCOUNTING	CLERK	1200.0000000000000000
ACCOUNTING	MANAGER	1500.0000000000000000
ACCOUNTING	PRESIDENT	5000.0000000000000000
ACCOUNTING	TRAINEE	0.00000000000000000000
RESEARCH	ANALYST	3000.0000000000000000
RESEARCH	CLERK	716.6666666666666667
RESEARCH	MANAGER	2975.0000000000000000
SALES	MANAGER	2850.0000000000000000
SALES	SALESMAN	1400.0000000000000000
SALES2	SALESMAN	2450.0000000000000000
VILLAIN	BA	5016.6666666666666667

25. Складіть звіт, який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат групувану за «відділом та посадою» та відділом. (стовпці звіту: відділ, посада, сумарна зарплата)

```
SELECT dname, job, SUM(sal) FROM emp JOIN dept USING (deptno)
GROUP BY dname, job
ORDER BY dname;
```

26. * Складіть звіт, який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат, групувану за «місто, відділ, посада», «відділ і посаду» та по відділу.

```
SELECT loc, dname, job, SUM(sal) FROM emp JOIN dept USING (deptno)
GROUP BY loc, dname, job
ORDER BY loc;
```

27. * Складіть звіт який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат групувану за «місто, відділ, посада», «відділ та посаду», «місто, посада», «місто, відділ», «місто», «відділ» , "Посада".

```
SELECT loc, dname, job, SUM(sal) FROM emp JOIN dept USING (deptno)
GROUP BY loc, dname, job
ORDER BY loc;
```

```
1 SELECT loc, dname, job, SUM(sal) FROM emp JOIN dept USING (deptno)
2 GROUP BY loc, dname, job
3 ORDER BY loc;
```

loc	dname	job	sum
CHICAGO	SALES	MANAGER	2850
CHICAGO	SALES	SALESMAN	5600
DALLAS	RESEARCH	ANALYST	6000
DALLAS	RESEARCH	CLERK	2150
DALLAS	RESEARCH	MANAGER	2975
HONKONG	SALES2	SALESMAN	7350
NEW_YORK	ACCOUNTING	CLERK	1200
NEW_YORK	ACCOUNTING	MANAGER	1500
NEW_YORK	ACCOUNTING	PRESIDENT	5000
NEW_YORK	ACCOUNTING	TRAINEE	0
NEW_YORK	VILLAIN	BA	60200

Статистика

28. * Виведіть інформацію у 3 колонки: назва міста, кількість керівників, кількість осіб що НЕ мають підлеглих (не керівників). Для «**» у списку обов'язково мають бути присутні і міста без співробітників.

```
SELECT loc, COUNT(DISTINCT mgr) as mgr_count, (COUNT(*) -  
COUNT(DISTINCT mgr)) as non_mgr_count  
FROM emp JOIN dept USING (deptno)  
GROUP BY loc;
```

```
1 |SELECT  
2 |   loc,  
3 |   COUNT(DISTINCT mgr) as mgr_count,  
4 |   (COUNT(*) - COUNT(DISTINCT COALESCE(mgr, 0))) as non_mgr_count  
5 | FROM emp RIGHT JOIN dept USING (deptno)  
6 | GROUP BY loc;
```

loc	mgr_count	non_mgr_count
BOSTON	0	0
CHICAGO	2	3
DALLAS	5	1
HONKONG	2	1
NEW_YORK	2	14

Деканат

Візьміть таблиці зі свого ДЗ для 1 модуля.

30. * Скільки балів виділено на дисципліну <назва дисципліни>? (сума балів за завдання дисципліни)

```
SELECT discipline, SUM(max_grade) FROM disciplines JOIN tasks ON  
(disciplines.id = tasks.discipline)  
GROUP BY discipline;
```

```
1 SELECT discipline, SUM(max_grade) FROM disciplines JOIN tasks ON (disciplines.id = tasks.discipline)
2 GROUP BY discipline;
3
4
5
6
7
8
9
```

Save Run

discipline	sum
1	300

31. * Скільки балів, зароблено студентом <ім'я студента> з дисципліни <назва дисципліни>. Зверніть увагу на свою схему: чи може студент кілька разів здавати роботу (1 спроба, 2 спроба)? Чи може бути оцінена робота кілька разів? ДЗ на «5» вимагало обидві ці можливості.

```
SELECT people_d.name as stud_name, disciplines.name as disc_name,  
sum(grade) FROM disciplines  
JOIN tasks ON (disciplines.id = tasks.discipline)  
JOIN reports ON (reports.task = tasks.id)  
JOIN grades ON (reports.id = grades.report AND grades.fin = True)  
JOIN people_d ON (people_d.id = grades.student)  
GROUP BY people_d.name, disciplines.name;
```

```
1 SELECT people_d.name as stud_name, disciplines.name as disc_name, sum(grade) FROM disciplines
2 JOIN tasks ON (disciplines.id = tasks.discipline)
3 JOIN reports ON (reports.task = tasks.id)
4 JOIN grades ON (reports.id = grades.report AND grades.fin = True)
5 JOIN people_d ON (people_d.id = grades.student)
6 GROUP BY people_d.name, disciplines.name;
7
8
9
```

Save Run

stud_name	disc_name	sum
Petro	Data Bases	55
Danil	Data Bases	150
Dmytro	Data Bases	172

Складні вибірки.

33. ** В компанії вирішили влаштувати обмін досвідом між співробітниками. Для цього кожен день один із співробітників робить доповідь. Співробітники виступають за абеткою. Після того, як всі співробітники зробили доповідь, перший співробітник. Створіть запит, який повертає ім'я доповідача, якщо відомо ім'я попереднього доповідача. З використанням агрегатних функцій завдання значно спрощується. Його можна зробити навіть у 1 рядок.

```
WITH ft as  
      (SELECT ename as current_emp, LEAD(ename, 1) OVER (ORDER BY ename  
ASC) as next_emp FROM emp)  
SELECT current_emp, COALESCE(next_emp, (SELECT current_emp FROM ft  
LIMIT 1)) as next_emp FROM ft WHERE current_emp = 'WARD';
```

Завдання за варіантами

Варіант 1

Оберіть буд-які 2 запити на Ваш розсуд (4 для «*», 8 для «**»).
Для «*» різницю у курсах валют ігноруємо, для «**» - враховуємо.

2. Виведіть рахунки, на які не надходили гроші у цьому місяці. (Але могли бути витрати)

```
SELECT DISTINCT accounts.* FROM accounts  
LEFT JOIN transactions ON (accounts.accountid = transactions.toid)  
WHERE  
    transactions.transactionid is NULL OR  
    transactions.toid NOT IN (  
        SELECT t1.toid FROM transactions t1  
        WHERE (date_trunc('month', t1.transactiondate) =  
date_trunc('month', CURRENT_DATE)))
```

```
1 SELECT DISTINCT accounts.* FROM accounts  
2 LEFT JOIN transactions ON (accounts.accountid = transactions.toid)  
3 WHERE  
4     transactions.transactionid is NULL OR  
5     transactions.toid NOT IN (  
6         SELECT t1.toid FROM transactions t1  
7         WHERE (date_trunc('month', t1.transactiondate) = date_trunc('month', CURRENT_DATE))  
8 )
```

[Save](#)

accountid ▲	clientid ▲	balance ▲	isblocked ▲
5	1	500	true
2	5	1000	true
3	7	5000	true
4	8	111000	false

3. «Скрудж». Знайдіть клієнтів, на рахунки яких гроші надходять, але не витрачаються.

```
SELECT accounts.* FROM accounts  
LEFT JOIN transactions ON (accounts.accountid = transactions.fromid)  
WHERE transactions.transactionid is Null;
```

```
1 SELECT accounts.* FROM accounts  
2 LEFT JOIN transactions ON (accounts.accountid = transactions.fromid)  
3 WHERE transactions.transactionid is Null;
```

accountid ▲	clientid ▲	balance ▲	isblocked ▲
3	7	5000	true

5. Хто з клієнтів найбагатший.

```
SELECT people.*, SUM(balance) as "money" FROM people  
JOIN accounts ON (accounts.clientid = people.id)  
GROUP BY people.id  
ORDER BY money DESC  
LIMIT 1;
```

```
1 SELECT people.*, SUM(balance) as "money" FROM people  
2 JOIN accounts ON (accounts.clientid = people.id)  
3 GROUP BY people.id  
4 ORDER BY money DESC  
5 LIMIT 1;
```

id ▲	firstname ▲	secondname ▲	sanctions ▲	money ▲
8	DANIL	FOVSKIY	(NULL)	111000

7. Виберіть по одному співробітнику з кожної посади.

```
SELECT DISTINCT ON (1)
    positions.name,
    workers.workerid,
    people.firstname,
    people.secondname
FROM positions
JOIN workers ON (workers.position = positions.positionid)
JOIN people ON (people.id = workers.pid)
```

```
1 SELECT DISTINCT ON (1)
2     positions.name,
3     workers.workerid,
4     people.firstname,
5     people.secondname
6 FROM positions
7 JOIN workers ON (workers.position = positions.positionid)
8 JOIN people ON (people.id = workers.pid)
```

name ▲	workerid ▲	firstname ▲	secondname ▲
admin	1	PETRO	IVANOV
cashier	2	DEMID	VASILEVSKIY