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));

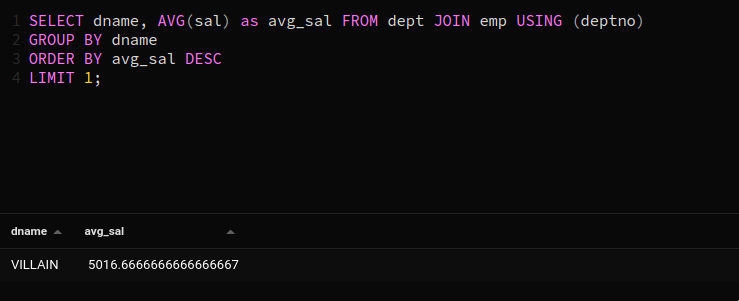
13. 📷 Назва відділу, у якому максимальна середня зарплата.

SELECT dname, AVG(sal) as avg\_sal FROM dept JOIN emp USING (deptno)

GROUP BY dname

ORDER BY avg\_sal DESC

LIMIT 1;

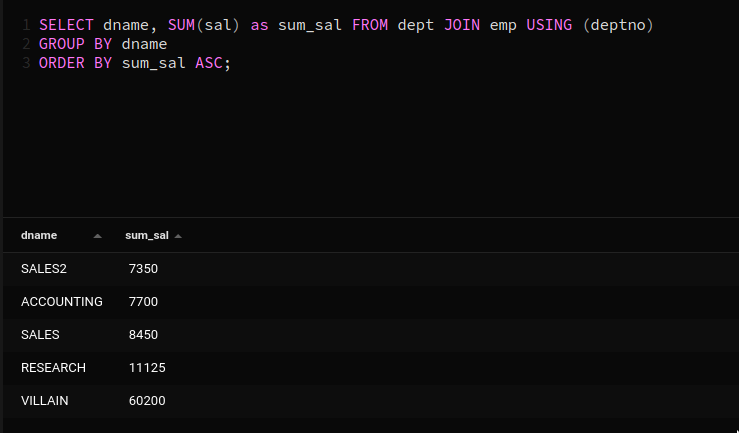


14. 📷 Виведете назви відділів у порядку зростання сум зарплат працівників, які працюють у них.

SELECT dname, SUM(sal) as sum\_sal FROM dept JOIN emp USING (deptno)

GROUP BY dname

ORDER BY sum\_sal ASC;



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 ВСІ 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 найбльшим 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"

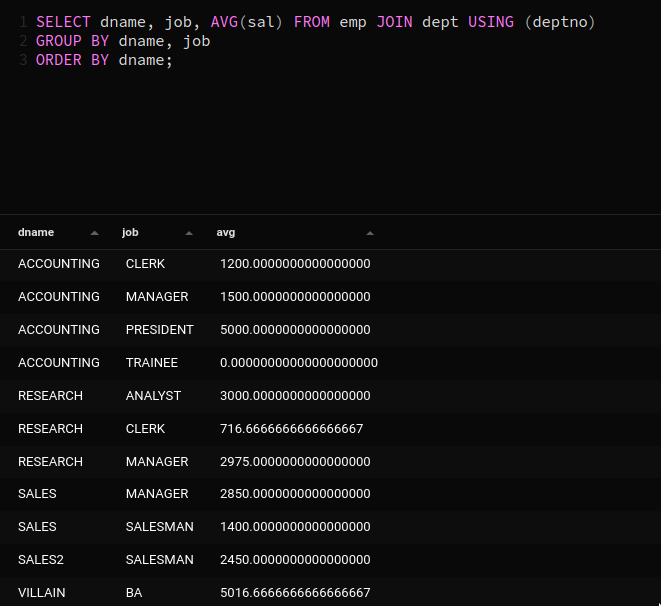
FROM emp JOIN dept USING (deptno)) r WHERE sal\_rank = 1;

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;



25. Складіть звіт, який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат груповану 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, груповану 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

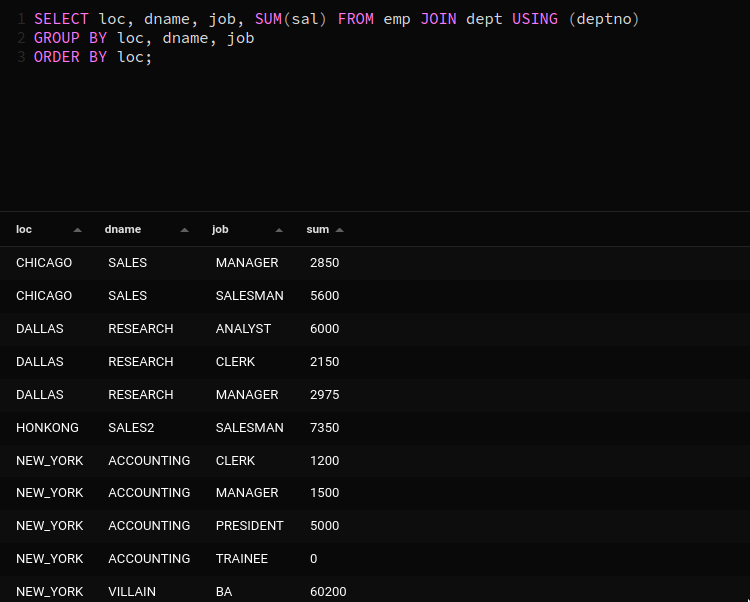
ORDER BY loc;

27. \* 📷 Складіть звіт який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат груповану 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;

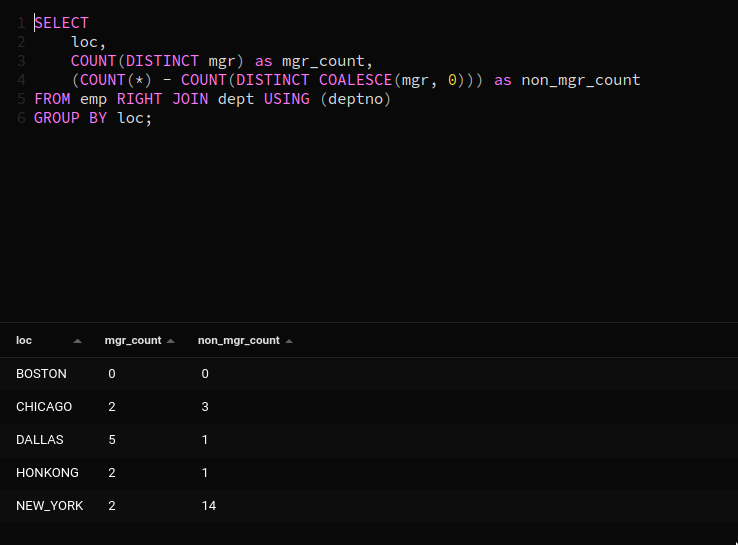


Статистика  
28. \* 📷 Виведіть інформацію у 3 колонки: назва міста, кількість керівників, кількість осіб що НЕ мають підлеглих ( не керівників). Для «\*\*» у списку обов’язково мають бути присутні і міста без співробітників.

SELECT loc, COUNT(DISTINCT mgr) as mgr\_count, (COUNT(\*) - COUNT(DISTINCT mgr)) as non\_mgr\_count

FROM emp JOIN dept USING (deptno)

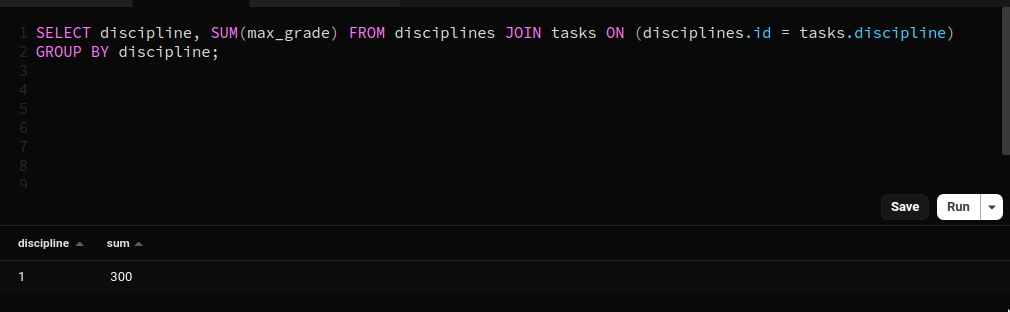
GROUP BY loc;



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;



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

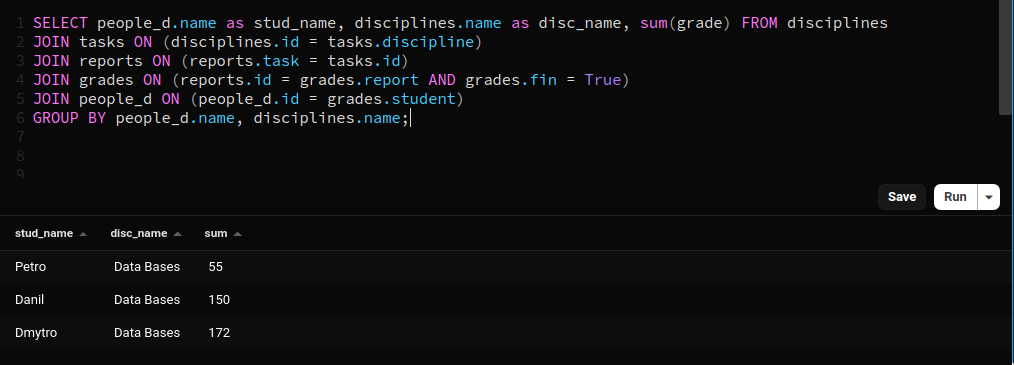
JOIN tasks ON (disciplines.id = tasks.discipline)

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;



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)

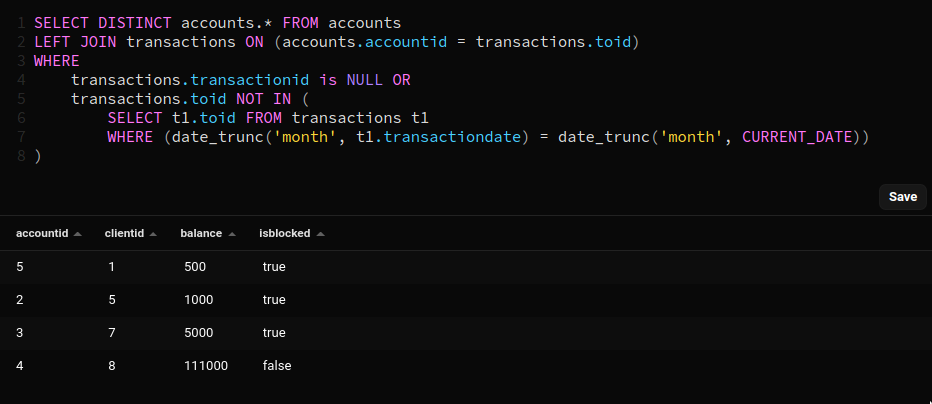
WHERE

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)))

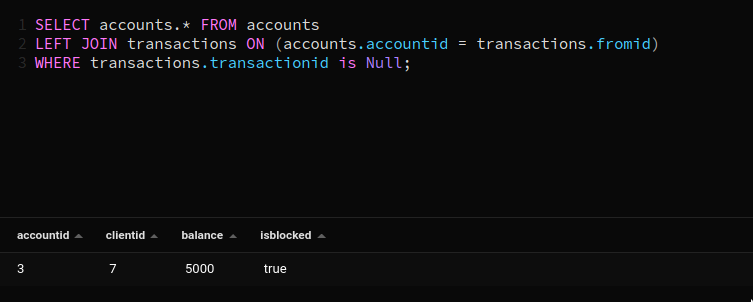


3. «Скрудж». Знайдіть клієнтів, на рахунки яких гроші надходять, але не витрачаються.

SELECT accounts.\* FROM accounts

LEFT JOIN transactions ON (accounts.accountid = transactions.fromid)

WHERE transactions.transactionid is Null;



5. Хто з клієнтів найбагатший.

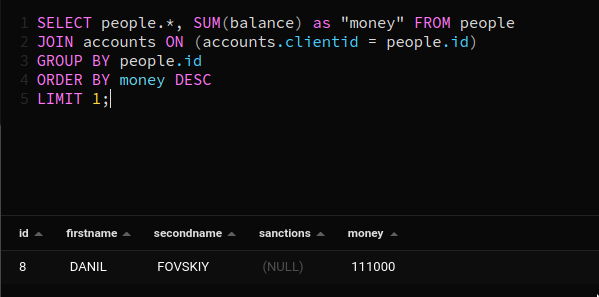
SELECT people.\*, SUM(balance) as "money" FROM people

JOIN accounts ON (accounts.clientid = people.id)

GROUP BY people.id

ORDER BY money DESC

LIMIT 1;



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)

