Лабораторна робота 5

Основне завдання

1. Сформулюйте запит, що виводить імена всіх підлеглих King.

SELECT ename

FROM Emp

WHERE mgr = (

SELECT empno

FROM Emp

WHERE ename = 'KING'

);

Изображение выглядит как текст

Автоматически созданное описание

2. Сформулюйте запит, що виводить імена співробітників, які працюють в одному відділі зі Smith.

SELECT ename

FROM Emp

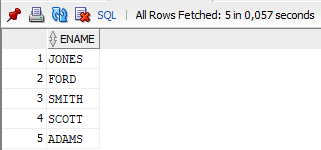
WHERE deptno = (

SELECT deptno

FROM Emp

WHERE ename = 'SMITH'

);



3. Сформулюйте запит, що виводить імена співробітників, найнятих після Smith до його відділу.

SELECT ename

FROM Emp

WHERE deptno = (

SELECT deptno

FROM Emp

WHERE ename = 'SMITH'

)

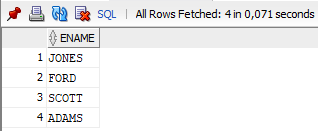
AND hiredate > (

SELECT hiredate

FROM Emp

WHERE ename = 'SMITH'

);



4. Сформуйте запит, що виводить прізвища керівників, у яких ВСІ підлеглі отримують більше $2500 **(зверніть увагу на цей запит!)**.

SELECT ename as manager\_name

FROM Emp e

WHERE job = 'MANAGER' AND NOT EXISTS (

SELECT \*

FROM Emp e2

WHERE e2.mgr = e.empno AND e2.sal <= 2500

);

Изображение выглядит как стол

Автоматически созданное описание

5. Сформулюйте запит, що виводить імена співробітників, найнятих після службовців відділу №30.

SELECT ename

FROM Emp

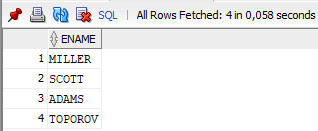
WHERE hiredate > ALL (

SELECT hiredate

FROM Emp

WHERE deptno = 30

);



6. Сформуйте запит, що виводить інформацію про працівників, зарплата яких більша за зарплату будь-якого (хоча б одного) з клерків.

SELECT ename, sal

FROM Emp

WHERE sal > ANY (

SELECT sal

FROM Emp

WHERE job = 'CLERK'

);

Изображение выглядит как стол

Автоматически созданное описание

7. Відділу кадрів потрібен звіт, що містить ім'я та зарплату для всіх співробітників з Dallas.

SELECT ename, sal

FROM Emp

WHERE deptno IN (

SELECT deptno

FROM Dept

WHERE loc = 'DALLAS'

)

Изображение выглядит как стол

Автоматически созданное описание

8. Сформувати запит, що виводить номер та прізвище працівника, номер та прізвище його менеджера.

SELECT e.empno, e.ename,

(SELECT m.empno || '-' || m.ename

FROM emp m

WHERE e.mgr = m.empno) as manager

FROM emp e;

Изображение выглядит как стол

Автоматически созданное описание

9. Сформувати запит, що виводить прізвище працівника та місто в якому він працює.

SELECT e.ename,

(SELECT d.loc

FROM dept d

WHERE e.deptno = d.deptno) as city

FROM emp e;

Изображение выглядит как стол

Автоматически созданное описание

10. Завдання:

1. Створіть запит, який виведе імена, номер відділу та зарплату співробітників з відділу №10.

SELECT e.ename, e.deptno, e.sal

FROM (SELECT \* FROM emp WHERE deptno = 10) e;

Изображение выглядит как стол

Автоматически созданное описание

2. Створіть запит, який вибере всіх службовців із зарплатою понад $1500 з результатів запиту 1.

SELECT \*

FROM (

SELECT e.ename, e.deptno, e.sal

FROM (SELECT \* FROM emp WHERE deptno = 10) e

) t

WHERE t.sal > 1500;

Изображение выглядит как стол

Автоматически созданное описание

3. Створіть подання(view) на основі запиту 1.

CREATE VIEW emp\_dept10 AS

SELECT e.ename, e.deptno, e.sal

FROM (SELECT \* FROM emp WHERE deptno = 10) e;



4. Створіть запит, який вибере всіх службовців із зарплатою понад $1500 з подання 3.

SELECT \*

FROM emp\_dept10

WHERE sal > 1500;

Изображение выглядит как стол

Автоматически созданное описание

5. Видаліть створене подання.

DROP VIEW emp\_dept10;

Изображение выглядит как текст

Автоматически созданное описание

11. Створіть звіт, який містить номери відділів, у яких немає клерків.

SELECT deptno

FROM emp

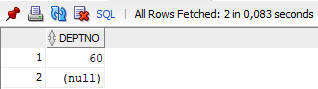
WHERE job != 'CLERK'

MINUS

SELECT deptno

FROM emp

WHERE job = 'CLERK';



12. Створіть звіт, який містить номери відділів, де немає співробітників.

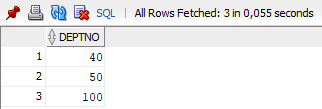
SELECT deptno

FROM dept

MINUS

SELECT deptno

FROM emp;



13. Створіть звіт, який містить номер відділу та посади, які є у відділах № 10, 30, та 20. Номери відділів мають бути у вказаному порядку (10, 30, 20).

SELECT \*

FROM (SELECT deptno, job

FROM Emp

WHERE deptno = 10

UNION

SELECT deptno, job

FROM Emp

WHERE deptno = 30

UNION

SELECT deptno, job

FROM Emp

WHERE deptno = 20)

ORDER BY

CASE deptno

WHEN 10 THEN 1

WHEN 30 THEN 2

WHEN 20 THEN 3

END;

Изображение выглядит как стол

Автоматически созданное описание

14. Створіть звіт, який містить імена та номери всіх співробітників відділу №10 (запит 1) ТА всіх співробітників у яких зарплата вища за $1500 (запит 2). Відсортуйте дані зі спадання імен співробітників.

SELECT empno, ename

FROM Emp

WHERE deptno = 10

UNION ALL

SELECT empno, ename

FROM Emp

WHERE sal > 1500

ORDER BY ename DESC;

Изображение выглядит как стол

Автоматически созданное описание

**Некоректний запит!** Отриманий результат містить дублікати.

15.Завдання:

Створіть представлення Emp\_USA яке містить усіх співробітників, що працюють у США (Нью Йорку, Далласі, Бостоні, Чикаго) (можна без Set-операцій)

CREATE VIEW Emp\_USA AS

SELECT \*

FROM Emp

WHERE deptno IN (

SELECT deptno

FROM Dept

WHERE loc IN ('NEW\_YORK', 'DALLAS', 'CHICAGO', 'BOSTON')

);

Изображение выглядит как текст

Автоматически созданное описание

та представлення Emp\_SALES, що містить всіх співробітників, що працюють на посаді SALES.

CREATE VIEW Emp\_SALES AS

SELECT \*

FROM Emp

WHERE deptno IN (

SELECT deptno

FROM Dept

WHERE dname LIKE '%SALES%'

);

Изображение выглядит как текст

Автоматически созданное описание

Складіть звіти:

1. Неповторний список співробітників, що працюють в Emp\_USA або Emp\_SALES.

SELECT \* FROM Emp\_USA

UNION

SELECT \* FROM Emp\_SALES;

Изображение выглядит как стол

Автоматически созданное описание

2. Список усіх співробітників, що працюють в Emp\_USA або Emp\_SALES.

SELECT \* FROM Emp\_USA

UNION ALL

SELECT \* FROM Emp\_SALES;

Изображение выглядит как стол

Автоматически созданное описание

3. Список співробітників які працюють і в Emp\_USA, і в Emp\_SALES.

SELECT \* FROM Emp\_USA

INTERSECT

SELECT \* FROM Emp\_SALES;

Изображение выглядит как стол

Автоматически созданное описание

4. Список співробітників, які працюють і в Emp\_SALES але не в Emp\_USA.

SELECT \* FROM Emp\_SALES

MINUS

SELECT \* FROM Emp\_USA;

Изображение выглядит как стол

Автоматически созданное описание

5. Видалити обидва представлення.

DROP VIEW Emp\_USA;

DROP VIEW Emp\_SALES;

Изображение выглядит как текст

Автоматически созданное описание

16. Складіть звіт, який відповідає всім наступним вимогам:

Звіт 1 містить ім'я співробітника та номер відділу для всіх співробітників, незалежно чи вони належать якомусь відділу (можна без Set-операцій).

CREATE VIEW report1 AS

SELECT ename, deptno

FROM Emp

UNION ALL

SELECT NULL AS ename, deptno

FROM Dept

WHERE deptno NOT IN (SELECT deptno FROM Emp);



Звіт 2 містить назву та номер відділу для всіх відділів, незалежно чи є у відділі хоча б один співробітник (можна без Set-операцій).

CREATE VIEW report2 AS

SELECT dname, deptno

FROM Dept

UNION ALL

SELECT NULL AS dname, deptno

FROM Emp

WHERE deptno NOT IN (SELECT deptno FROM Dept);



Об'єднуємо Звіти 1 та Звіт 2: Звіт 3 повинен містити 2 стовпці - ім'я співробітника та номер відділу або назву відділу та номер відділу.

SELECT ename, deptno

FROM report1

WHERE ename IS NOT NULL

UNION

SELECT dname, deptno

FROM report2

WHERE dname IS NOT NULL;

Изображение выглядит как стол

Автоматически созданное описание

17. Створіть запит, який виведе імена, номер відділу та зарплату співробітників з відділу №20, які йдуть на 2-му та 3-му місці за рівнем зарплати в цьому відділі.

Зверніть увагу, що найбільшу зарплату в цьому відділі – 3000 отримують двоє. Отже вивести треба Adams і Joins.

Підказки:

Для того, щоб обмежити кількість рядків, що виводяться в підзапитах, можна скористатися умовою: where rownum<=3, конструкцією fetch … rows.

SELECT e1.ename, e1.deptno, e1.sal

FROM Emp e1

WHERE e1.deptno = 20

AND e1.sal IN (

SELECT DISTINCT e2.sal

FROM Emp e2

WHERE e2.deptno = 20

ORDER BY e2.sal DESC

OFFSET 1 ROW FETCH NEXT 2 ROWS ONLY

)

ORDER BY e1.sal DESC;

Изображение выглядит как стол

Автоматически созданное описание

18. Завдання:

Створіть запит, який виведе імена, номер відділу та зарплату 2-х співробітників з відділу №30 та 2-х з відділу №20 із найбільшими зарплатами.

Підказки: Запити до службовців відділів 30 і 20 можна об'єднати використовуючи Union.

Для того, щоб мати можливість використовувати сортування в запитах усередині команди Union, їх потрібно обернути в Select.

SELECT e1.ename, e1.deptno, e1.sal

FROM Emp e1

WHERE e1.deptno = 30 AND e1.sal IN (

SELECT e2.sal

FROM Emp e2

WHERE e2.deptno = 30

ORDER BY e2.sal DESC

FETCH FIRST 2 ROWS ONLY

)

UNION

SELECT e3.ename, e3.deptno, e3.sal

FROM Emp e3

WHERE e3.deptno = 20 AND e3.sal IN (

SELECT e4.sal

FROM Emp e4

WHERE e4.deptno = 20

ORDER BY e4.sal DESC

FETCH FIRST 2 ROWS ONLY

)

ORDER BY sal DESC;

Изображение выглядит как стол

Автоматически созданное описание

19. У компанії вирішили влаштувати обмін досвідом між співробітниками. Для цього щодня один із співробітників робить доповідь. Співробітники виступають за абеткою. Після того, як всі співробітники зробили доповідь, черга знов переходить до першого співробітника. Створіть запит, який повертає ім'я доповідача, якщо відомо ім'я попереднього доповідача.

Изображение выглядит как текст

Автоматически созданное описание

20. Виведіть звіт: ім'я співробітника та його місце роботи.

Для людей з посадою MANAGER місце роботи – місто, в якому він працює. Для решти місце роботи – назва відділу, де вони працюють. У запиті назву міста та назву відділу знаходимо підзапитом.

21. «Злиття та приєднання»:

Вітаємо! Ми купили та приєднали до своєї фірми крутий start-up. Дані щодо його співробітників є у файлі union.sql Виведіть список співробітників (наших та зі стартапу): номер, ПІБ, зарплату, премію.

Изображение выглядит как текст

Автоматически созданное описание

22. Знайти людину, яка є керівником (має підлеглих), але не менеджер (має іншу посаду аніж «MANAGER»).

SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno

FROM emp

WHERE empno IN (

SELECT DISTINCT mgr

FROM emp

WHERE job != 'MANAGER'

);

Изображение выглядит как стол

Автоматически созданное описание

**Некоректний запит!** Виведено керівників з посадою 'MANAGER'.

23. Начальник, всі підлеглі якого працюють у тому ж відділі.

SELECT DISTINCT e1.\*

FROM Emp e1

WHERE NOT EXISTS (

SELECT e2.empno

FROM Emp e2

WHERE e2.mgr = e1.empno AND e2.deptno <> e1.deptno

);

Изображение выглядит как стол

Автоматически созданное описание

**Некоректний запит!** У результаті виведено також співробітників, які не є керівниками.

24. Вивести імена співробітників, у яких керівник працює у іншому МІСТІ. Зверніть увагу, що у місті може бути кілька відділів.

SELECT e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm, e.deptno

FROM Emp e

WHERE e.mgr IN (

SELECT empno

FROM Emp

WHERE deptno IN (

SELECT deptno

FROM Dept

WHERE loc <> (

SELECT loc

FROM Dept

WHERE deptno = e.deptno

)

)

);

Изображение выглядит как стол

Автоматически созданное описание

25. Для всіх відділів вивести ім'я співробітника із найбільшою зарплатою у цьому відділі (без функції МАХ). Іншими словами: із підзапиту завжди повертати лише один рядок із визначеним пріоритетом.

SELECT e1.ename, d.dname

FROM Emp e1, Dept d

WHERE e1.deptno = d.deptno

AND NOT EXISTS (

SELECT \*

FROM Emp e2

WHERE e2.deptno = d.deptno

AND e2.sal > e1.sal

);

Изображение выглядит как стол

Автоматически созданное описание

Так як співробітники FORD і SCOTT мають однакову зарплату і працюють в одному і тому ж відділі, тому запит вивів такий результат.

Завдання 2

**Варіант 1**

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

2. Уважно подивіться на Ваші таблиці «клієнт» та «співробітник». За умовою клієнт може бути співробітником. Реалізуйте запит, що знайде людей, про яких в різних таблицях міститься різна інформація (наприклад: людина одна, а ім’я чи адреса не збігаються). Поясніть, якщо це неможливо у Вашій БД.

3. «Скрудж». Знайдіть рахунки, на які гроші виключно нараховуються, але не витрачаються.

4. Виведіть клієнтів, за рахунками яких не проводились операції останні півроку. (Перевірка: у клієнта є активний ( багато операції) та неактивний (нема операцій) рахунок – клієнта не має виводити. Перевірка 2: клієнти без рахунків).

5. Співробітник може бути клієнтом. Операції по рахунку обов’язково перевіряються співробітником. Знайдіть співробітників, які перевіряли операції самі собі.

**Розв’язання**

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

SELECT \*

FROM "BankAcc"

WHERE "BankAcc\_ID" NOT IN (

SELECT "BankAcc\_ID"

FROM "Operation"

WHERE "OperationKind" = 'INCOME'

AND EXTRACT(MONTH FROM "OperationDate") = EXTRACT(MONTH FROM SYSDATE)

);

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Так як остання операція ‘INCOME’ відбулася ще 30 листопада 2022 року, тому запит вивів усі банківські рахунки.

2. Уважно подивіться на Ваші таблиці «клієнт» та «співробітник». За умовою клієнт може бути співробітником. Реалізуйте запит, що знайде людей, про яких в різних таблицях міститься різна інформація (наприклад: людина одна, а ім’я чи адреса не збігаються). Поясніть, якщо це неможливо у Вашій БД.

Изображение выглядит как текст

Автоматически созданное описание

3. «Скрудж». Знайдіть рахунки, на які гроші виключно нараховуються, але не витрачаються.

SELECT ba."BankAcc\_ID", ba."Sum", ba."Code", ba."DateOpen", ba."DateClose", ba."PassportInfo"

FROM "BankAcc" ba

WHERE ba."BankAcc\_ID" NOT IN (

SELECT o."BankAcc\_ID"

FROM "Operation" o

WHERE o."OperationKind" = 'OUTCOME'

)

AND ba."BankAcc\_ID" IN (

SELECT o."BankAcc\_ID"

FROM "Operation" o

WHERE o."OperationKind" = 'INCOME'

);

Изображение выглядит как стол

Автоматически созданное описание

4. Виведіть клієнтів, за рахунками яких не проводились операції останні пів-року. (Перевірка: у клієнта є активний (багато операції) та неактивний (нема операцій) рахунок – клієнта не має виводити. Перевірка 2: клієнти без рахунків).

SELECT "PassportInfo"

FROM "Client"

WHERE NOT EXISTS (

SELECT "BankAcc\_ID"

FROM "BankAcc"

WHERE "PassportInfo" = "Client"."PassportInfo"

AND (

"DateClose" IS NULL

OR "DateClose" >= ADD\_MONTHS(SYSDATE, -6)

)

AND EXISTS (

SELECT "Operation\_ID"

FROM "Operation"

WHERE "BankAcc\_ID" = "BankAcc"."BankAcc\_ID"

AND "OperationDate" >= ADD\_MONTHS(SYSDATE, -6)

)

)

AND NOT EXISTS (

SELECT "BankAcc\_ID"

FROM "BankAcc"

WHERE "PassportInfo" = "Client"."PassportInfo"

AND (

"DateClose" IS NULL

OR "DateClose" >= ADD\_MONTHS(SYSDATE, -6)

)

AND NOT EXISTS (

SELECT "Operation\_ID"

FROM "Operation"

WHERE "BankAcc\_ID" = "BankAcc"."BankAcc\_ID"

AND "OperationDate" >= ADD\_MONTHS(SYSDATE, -6)

)

);

Изображение выглядит как стол

Автоматически созданное описание

**Некоректний запит!** Даний запит виводить лише клієнтів без банківських рахунків.

5. Співробітник може бути клієнтом. Операції по рахунку обов’язково перевіряються співробітником. Знайдіть співробітників, які перевіряли операції самі собі.

SELECT e."PassportInfo", e."EmpDate", e."Position"

FROM "Employee" e

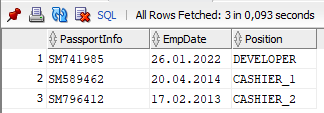
WHERE e."PassportInfo" IN (

SELECT o."PassportInfo"

FROM "Operation" o

WHERE o."PassportInfo" = e."PassportInfo"

);



**Некоректний запит!** Даний запит виводить лише співробітників, які перевіряли операції на банківських рахунках.