Лабораторна робота 10

Основне завдання

1. Напишіть запит, який повертав би з таблиці emp інформацію:

* про ім'я співробітника;
* про рівень підпорядкованості (найвищий рівень – головний начальник, який нікому не підпорядковується – рівень 1);
* про шлях підзвітності у форматі: /керівник 1/керівник 2/рядовий співробітник.

Звіт має бути відсортований за рівнем підпорядкованості.

SELECT

ename as "Name",

LEVEL as "Level",

SYS\_CONNECT\_BY\_PATH(ename, '/') as "Path"

FROM

Emp

START WITH

mgr IS NULL

CONNECT BY

PRIOR empno = mgr

ORDER BY

LEVEL, ename;

Изображение выглядит как стол

Автоматически созданное описание

2. Виведіть усі імена та зарплати начальників співробітника на ім'я «JET LI».

Наданий ChatGPT запит:

WITH manager\_chain (empno, ename, salary, manager\_empno, lvl) AS (

SELECT emp.empno, emp.ename, emp.sal, emp.mgr, 1 AS lvl

FROM emp

WHERE emp.ename = 'JET LI'

UNION ALL

SELECT emp.empno, emp.ename, emp.sal, emp.mgr, lvl + 1

FROM emp

JOIN manager\_chain ON emp.empno = manager\_chain.manager\_empno

)

SELECT m.ename, m.sal

FROM manager\_chain mc

JOIN emp m ON mc.manager\_empno = m.empno

WHERE mc.manager\_empno IS NOT NULL;

Изображение выглядит как текст

Автоматически созданное описание

Відкорегований запит:

WITH jet\_li (mgr, sal, ename) AS (

SELECT emp.mgr, emp.sal, emp.ename

FROM Emp emp

WHERE emp.ename = 'JET LI'

), manager (empno, ename, sal, mgr) AS (

SELECT emp.empno, emp.ename, emp.sal, emp.mgr

FROM Emp emp

JOIN jet\_li ON jet\_li.mgr = emp.empno

UNION ALL

SELECT emp.empno, emp.ename, emp.sal, emp.mgr

FROM Emp emp

JOIN manager ON manager.mgr = emp.empno

)

SELECT manager.ename, manager.sal

FROM manager

WHERE manager.sal = (

SELECT MAX(m.sal)

FROM manager m

);

Изображение выглядит как стол

Автоматически созданное описание

3. Хто з начальників співробітника «JET LI» отримує найвищу зарплату?

Наданий ChatGPT запит:

WITH jet\_li AS (

SELECT emp.mgr, emp.sal, emp.ename

FROM Emp emp

WHERE emp.ename = 'JET LI'

), manager AS (

SELECT emp.empno, emp.ename, emp.sal, emp.mgr

FROM Emp emp

JOIN jet\_li ON jet\_li.mgr = emp.empno

UNION ALL

SELECT emp.empno, emp.ename, emp.sal, emp.mgr

FROM Emp emp

JOIN manager ON manager.mgr = emp.empno

)

SELECT manager.ename, manager.sal

FROM manager

WHERE manager.sal = (

SELECT MAX(m.sal)

FROM manager m

);

Изображение выглядит как текст

Автоматически созданное описание

Відкорегований запит:

WITH manager\_chain (empno, ename, salary, manager\_empno, lvl) AS (

SELECT emp.empno, emp.ename, emp.sal, emp.mgr, 1 AS lvl

FROM emp

WHERE emp.ename = 'JET LI'

UNION ALL

SELECT emp.empno, emp.ename, emp.sal, emp.mgr, lvl + 1

FROM emp

JOIN manager\_chain ON emp.empno = manager\_chain.manager\_empno

)

SELECT m.ename, m.sal

FROM manager\_chain mc

JOIN emp m ON mc.manager\_empno = m.empno

WHERE mc.manager\_empno IS NOT NULL;

Изображение выглядит как стол

Автоматически созданное описание

4. Виведіть клерків, які є підлеглими (чи підлеглими підлеглих, …) Блейка.

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

FROM Emp

WHERE job = 'CLERK'

CONNECT BY PRIOR empno = mgr

START WITH mgr = (

SELECT empno

FROM Emp

WHERE ename = 'BLAKE'

)

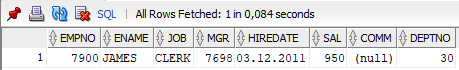
AND empno != (

SELECT empno

FROM Emp

WHERE ename = 'BLAKE'

);



5. Виведіть клерків, які НЕ є підлеглими (чи підлеглими підлеглих, …) Блейка.

WITH clerk\_tree (empno, ename, mgr) AS (

SELECT empno, ename, mgr

FROM Emp

WHERE job = 'CLERK'

UNION ALL

SELECT Emp.empno, Emp.ename, Emp.mgr

FROM Emp, clerk\_tree

WHERE Emp.empno = clerk\_tree.mgr

)

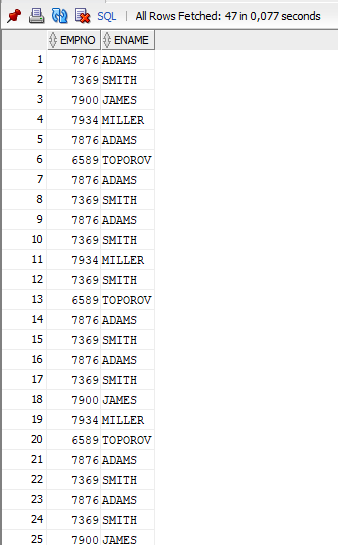
SELECT empno, ename

FROM clerk\_tree

WHERE CONNECT\_BY\_ISLEAF = 1

START WITH ename <> 'BLAKE'

CONNECT BY PRIOR empno = mgr;



**Некоректний запит!**

6. У кого зі співробітників 2-ї ланки (підлеглих Кінга) у підпорядкуванні найбільше клерків?

WITH Subordinates(empno, ename, lvl) AS (

SELECT empno, ename, 1 AS lvl

FROM Emp

WHERE empno = 7839 -- KING

UNION ALL

SELECT e.empno, e.ename, s.lvl + 1

FROM Emp e

JOIN Subordinates s ON s.empno = e.mgr

)

SELECT s1.ename, COUNT(c.empno) AS clerk\_count

FROM Subordinates s1

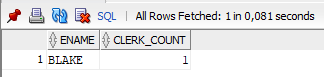
JOIN Emp c ON c.mgr = s1.empno AND c.job = 'CLERK'

WHERE s1.lvl = 2

GROUP BY s1.empno, s1.ename

ORDER BY clerk\_count DESC

FETCH FIRST 1 ROW ONLY;



**Некоректний запит!**

7. Завдання.

* Створіть таблицю T, заповнену числами -3..10.

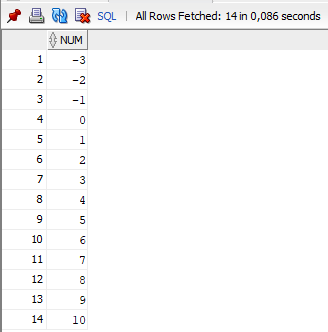
CREATE TABLE T (num NUMBER);

INSERT INTO T (num)

SELECT LEVEL - 4 FROM DUAL CONNECT BY LEVEL <= 14;

Изображение выглядит как текст

Автоматически созданное описание

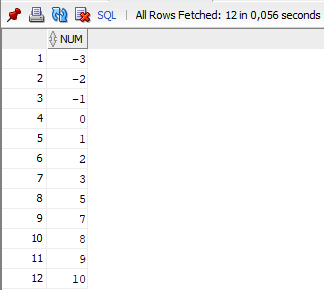


* Видаліть з таблиці числа 4 і 6. Уявімо, що ми не знаємо, які саме числа були видалені, але точно НЕ перше чи останнє.

DELETE FROM T

WHERE num = 4 OR num = 6;





* Напишіть запит, який виведе видалені числа.

WITH recursive\_nums (num) AS (

SELECT -3 AS num FROM DUAL

UNION ALL

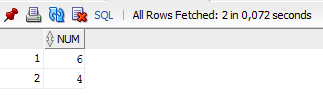
SELECT num + 1 AS num FROM recursive\_nums WHERE num < 10

)

SELECT num

FROM recursive\_nums

WHERE num NOT IN (SELECT num FROM T);



* Як буде виглядати запит, якщо максимальний елемент наперед невідомий?

WITH recursive\_nums (num) AS (

SELECT -3 AS num FROM DUAL

UNION ALL

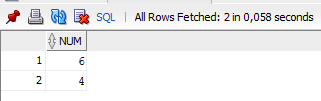
SELECT num + 1 AS num FROM recursive\_nums WHERE num < (SELECT MAX(num) FROM T)

)

SELECT num

FROM recursive\_nums

WHERE num NOT IN (SELECT num FROM T);



* Як буде виглядати запит, якщо максимальний і мінімальний елемент наперед невідомі?

WITH recursive\_nums (num) AS (

SELECT (SELECT MIN(num) FROM T) AS num FROM DUAL

UNION ALL

SELECT num + 1 AS num FROM recursive\_nums

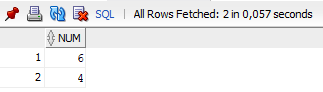
WHERE num < (SELECT MAX(num) FROM T)

)

SELECT num

FROM recursive\_nums

WHERE num NOT IN (SELECT num FROM T);



* Видаліть таблицю Т.

DROP TABLE T;



Втекти звідси

Створіть таблицю відстаней між містами (не менше 8 рядків):

CREATE TABLE Cities (

CityA Varchar2(30),

CityB Varchar2(30),

Distance Number

);

INSERT INTO Cities VALUES ('New York', 'Los Angeles', 2800);

INSERT INTO Cities VALUES ('Los Angeles', 'New York', 2800);

INSERT INTO Cities VALUES ('New York', 'Chicago', 800);

INSERT INTO Cities VALUES ('Chicago', 'New York', 800);

INSERT INTO Cities VALUES ('Los Angeles', 'Chicago', 2100);

INSERT INTO Cities VALUES ('Chicago', 'Los Angeles', 2100);

INSERT INTO Cities VALUES ('New York', 'Houston', 1600);

INSERT INTO Cities VALUES ('Houston', 'New York', 1600);

INSERT INTO Cities VALUES ('Chicago', 'Houston', 940);

INSERT INTO Cities VALUES ('Houston', 'Chicago', 940);

INSERT INTO Cities VALUES ('Los Angeles', 'Houston', 1500);

INSERT INTO Cities VALUES ('Houston', 'Los Angeles', 1500);

INSERT INTO Cities VALUES ('New York', 'Miami', 1300);

INSERT INTO Cities VALUES ('Miami', 'New York', 1300);

INSERT INTO Cities VALUES ('Chicago', 'Miami', 1200);

INSERT INTO Cities VALUES ('Miami', 'Chicago', 1200);

INSERT INTO Cities VALUES ('Los Angeles', 'Miami', 2700);

INSERT INTO Cities VALUES ('Miami', 'Los Angeles', 2700);

INSERT INTO Cities VALUES ('Houston', 'Miami', 960);

INSERT INTO Cities VALUES ('Miami', 'Houston', 960);

8. Виведіть дані з Вашої таблиці.

SELECT \* FROM Cities;

Изображение выглядит как стол

Автоматически созданное описание

9. Куди можна доїхати за 2 пересадки з обраного вами міста?

WITH route (start\_city, end\_city, Hops) AS (

SELECT CityA, CityB, 0 AS Hops

FROM Cities

WHERE CityA = 'New York'

UNION ALL

SELECT route.start\_city, Cities.CityB, route.Hops + 1

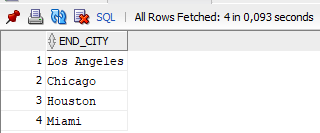
FROM route

JOIN Cities ON route.end\_city = Cities.CityA

WHERE route.Hops + 1 = 2

)

SELECT DISTINCT end\_city FROM route;



10. Знайдіть маршрут з мінімальною кількістю пересадок між двома обраними вами містами.

WITH city\_connections (origin, destination, transfers) AS (

SELECT CityA, CityB, 0 FROM Cities WHERE CityA = 'New York'

UNION ALL

SELECT c.CityA, c.CityB, cc.transfers+1 FROM Cities c

INNER JOIN city\_connections cc ON c.CityA = cc.destination

WHERE c.CityB = 'Miami'

)

SELECT MIN(transfers) AS minimum\_transfers

FROM city\_connections;

Изображение выглядит как стол

Автоматически созданное описание

Або:

WITH Route(CityA, CityB, Transfers, Path) AS (

SELECT CityA, CityB, 0, CityA || ' -> ' || CityB

FROM Cities

WHERE CityA = 'New York' AND CityB = 'Miami'

UNION ALL

SELECT Cities.CityA, Route.CityB, Route.Transfers + 1, Route.Path || ' -> ' || Cities.CityA

FROM Cities, Route

WHERE Cities.CityB = Route.CityA

AND Route.Path NOT LIKE '%' || Cities.CityA || '%'

)

SELECT Path, Transfers

FROM Route

WHERE CityA = 'New York' AND CityB = 'Miami'

AND Transfers = (SELECT MIN(Transfers) FROM Route WHERE CityA = 'New York' AND CityB = 'Miami');

Изображение выглядит как стол

Автоматически созданное описание

11. Знайдіть найкоротший маршрут між вибраними вами містами.

WITH Route(CityA, CityB, Distance, Path) AS (

SELECT CityA, CityB, Distance, CityA || ' -> ' || CityB

FROM Cities

WHERE CityA = 'New York' AND CityB = 'Miami'

UNION ALL

SELECT Cities.CityA, Route.CityB, Route.Distance + Cities.Distance, Route.Path || ' -> ' || Cities.CityA

FROM Cities, Route

WHERE Cities.CityB = Route.CityA

AND Route.Path NOT LIKE '%' || Cities.CityA || '%'

)

SELECT Path, Distance

FROM Route

WHERE CityA = 'New York' AND CityB = 'Miami'

AND Distance = (SELECT MIN(Distance) FROM Route WHERE CityA = 'New York' AND CityB = 'Miami');

Изображение выглядит как стол

Автоматически созданное описание

Видаліть таблицю.

DROP TABLE Cities;

Изображение выглядит как текст

Автоматически созданное описание

Завдання за варіантами

**Варіант 1**

1. У співробітниках додайте поле «керівник» - хто з співробітників є керівником для даного. У випадку, якщо клієнт незадоволений швидкістю проведення операції, він може звернутись до керівника. Значи операцію виведіть весь перелік керівників – до кого можна звернутись із скаргою.

2. Для двох співробітників визначить, хто є їх спільним найближчим керівником.

**Розв’язання**

1. У співробітниках додайте поле «керівник» - хто з співробітників є керівником для даного. У випадку, якщо клієнт незадоволений швидкістю проведення операції, він може звернутись до керівника. Знаючи операцію виведіть весь перелік керівників – до кого можна звернутись із скаргою.

WITH operation\_employee AS (

SELECT "PassportInfo"

FROM "Operation"

WHERE "Operation\_ID" = :operation\_id

)

SELECT "Employee"."PassportInfo", "Manager"."PassportInfo" AS "Manager\_PassportInfo"

FROM "Employee"

JOIN "Person" ON "Employee"."PassportInfo" = "Person"."PassportInfo"

JOIN "Person" "Manager" ON "Employee"."Manager" = "Manager"."PassportInfo"

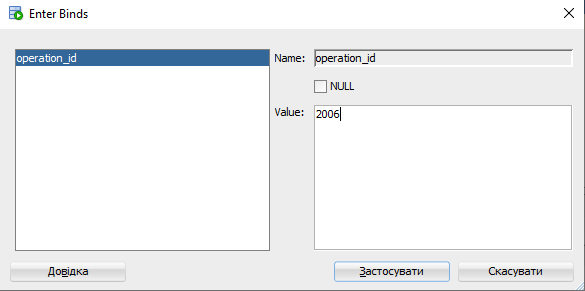
WHERE "Employee"."PassportInfo" IN (

SELECT "PassportInfo" FROM operation\_employee

)

CONNECT BY PRIOR "Manager"."PassportInfo" = "Employee"."PassportInfo"

START WITH "Employee"."Manager" IS NULL;



Изображение выглядит как текст

Автоматически созданное описание

**Некоректний запит!**

2. Для двох співробітників визначить, хто є їх спільним найближчим керівником.

WITH employee\_cte AS (

SELECT "Employee"."PassportInfo", "Employee"."Manager"

FROM "Employee"

WHERE "Employee"."PassportInfo" IN (:emp1\_passport\_info, :emp2\_passport\_info)

), manager\_cte ("PassportInfo", "Manager") AS (

SELECT "Employee"."PassportInfo", "Employee"."Manager"

FROM "Employee"

JOIN employee\_cte ON "Employee"."PassportInfo" = employee\_cte."Manager"

WHERE "Employee"."Manager" IS NOT NULL

UNION ALL

SELECT manager\_cte."PassportInfo", "Employee"."Manager"

FROM "Employee"

JOIN manager\_cte ON "Employee"."PassportInfo" = manager\_cte."Manager"

WHERE "Employee"."Manager" IS NOT NULL

)

SELECT "Person"."PassportInfo", "Person"."Name"

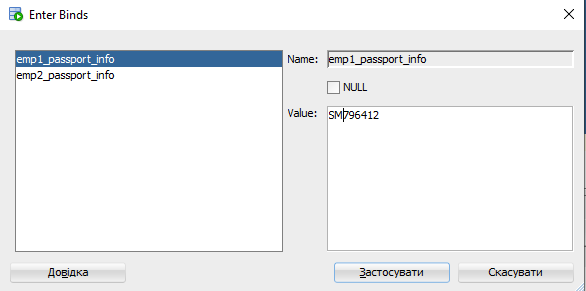
FROM "Person"

JOIN manager\_cte ON "Person"."PassportInfo" = manager\_cte."Manager"

WHERE manager\_cte."PassportInfo" IN (

SELECT "PassportInfo" FROM employee\_cte

);



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

**Некоректний запит!**