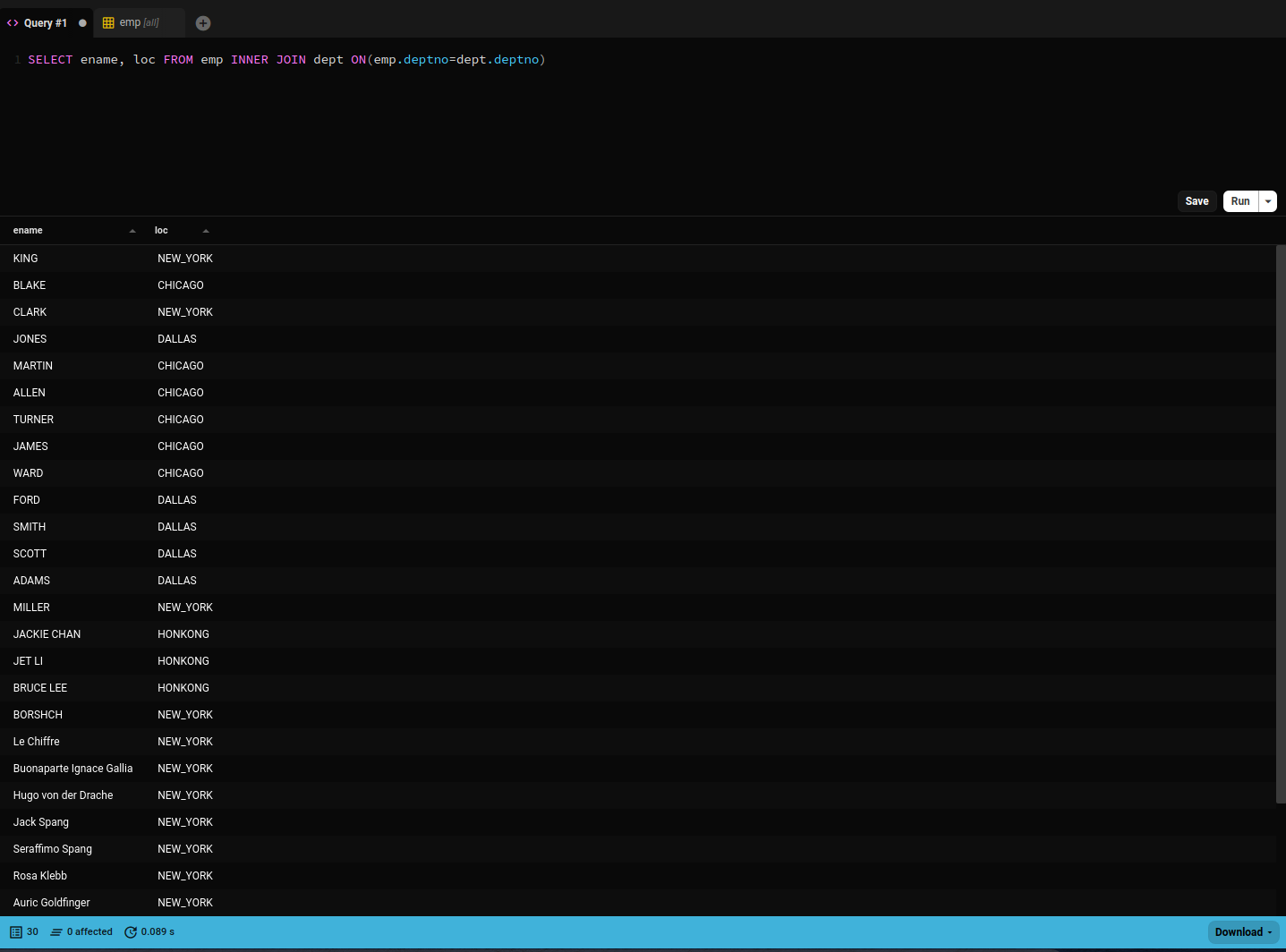
ЛАБОРАТОРНА РОБОТА 8 «JOIN»  
⚠️ Зверніть увагу, що в рамках цієї лабораторної роботи ми не використовуємо агрегатні та віконні (аналітичні ( f() over (...) ) функції, групування (GROUP BY), добуток таблиць (кілька таблиць у FROM). Підзапити – у виключних випадках.  
⚠️ Для завдань/розділів відмічених «📷» – скріншот – обов’язково. Для інших – за домовленістю із викладачем.

Завдання лабораторної роботи виконую всі

Основне завдання  
INNER JOIN  
1. 📷 Створіть звіт який містить ім'я співробітника та місто, в якому він працює.

SELECT ename, loc FROM emp INNER JOIN dept ON (emp.deptno=dept.deptno)

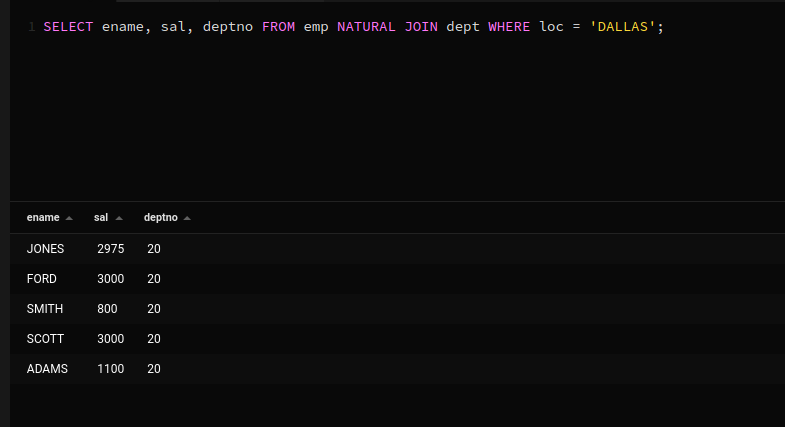


2. Сформувати запит, який виводить співробітників, які працюють у місті, назва якого містить 'O'.

SELECT emp.\* FROM emp INNER JOIN dept ON(dept.loc LIKE '%O%' AND emp.deptno=dept.deptno)

3. 📷 Виконайте завдання тричі (використовуючи NATURAL JOIN, JOIN ... ON та JOIN ... USING ):  
Відділу кадрів потрібен звіт, що містить ім'я, зарплату, номер відділу для всіх співробітників з Dallas.

SELECT ename, sal, deptno FROM emp NATURAL JOIN dept WHERE loc = 'DALLAS';



SELECT ename, sal, emp.deptno FROM emp INNER JOIN dept ON (emp.deptno = dept.deptno AND dept.loc = 'DALLAS')



SELECT ename, sal, deptno FROM emp INNER JOIN dept USING (deptno) WHERE loc = 'DALLAS';

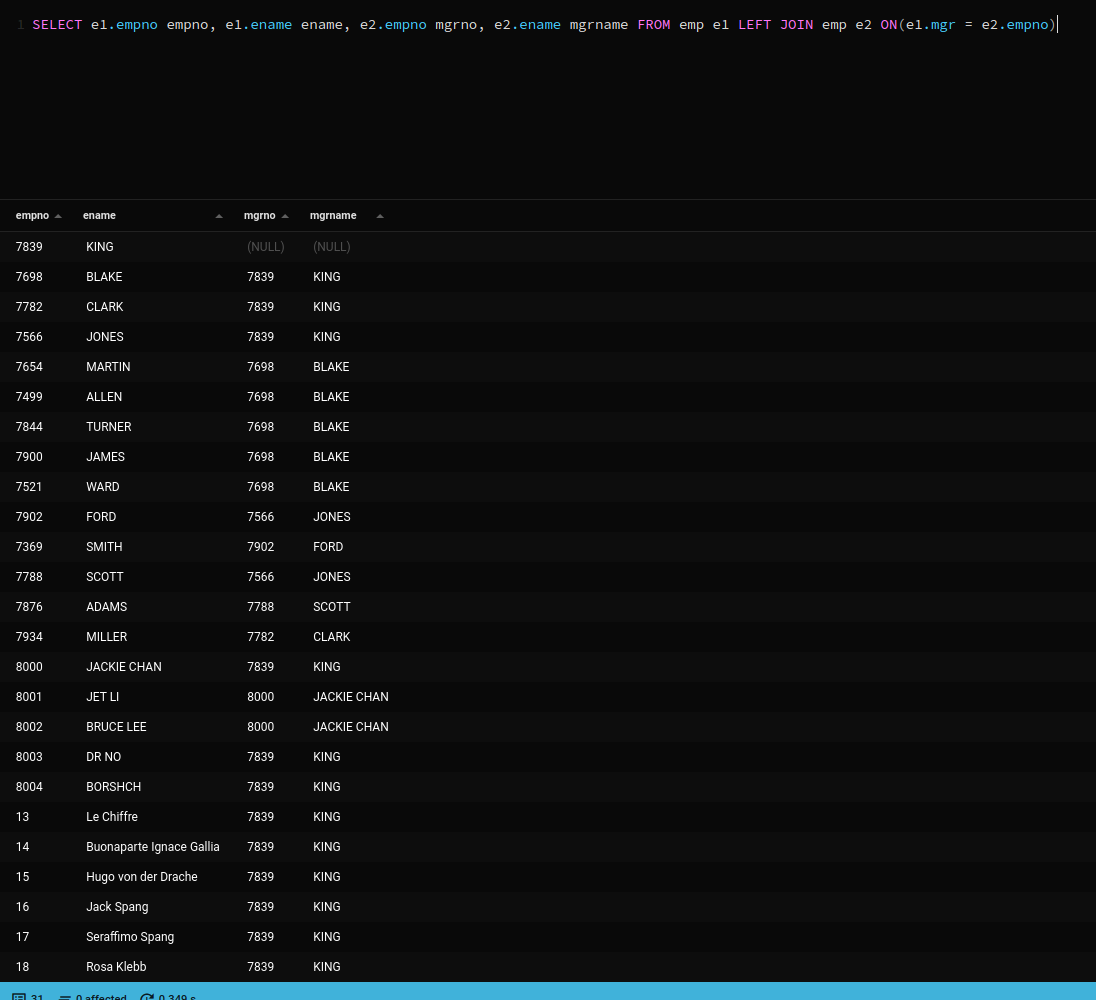


4. Сформувати запит, що виводить номер та прізвище працівника, номер та прізвище його менеджера для тих співробітників, у яких є менеджер.

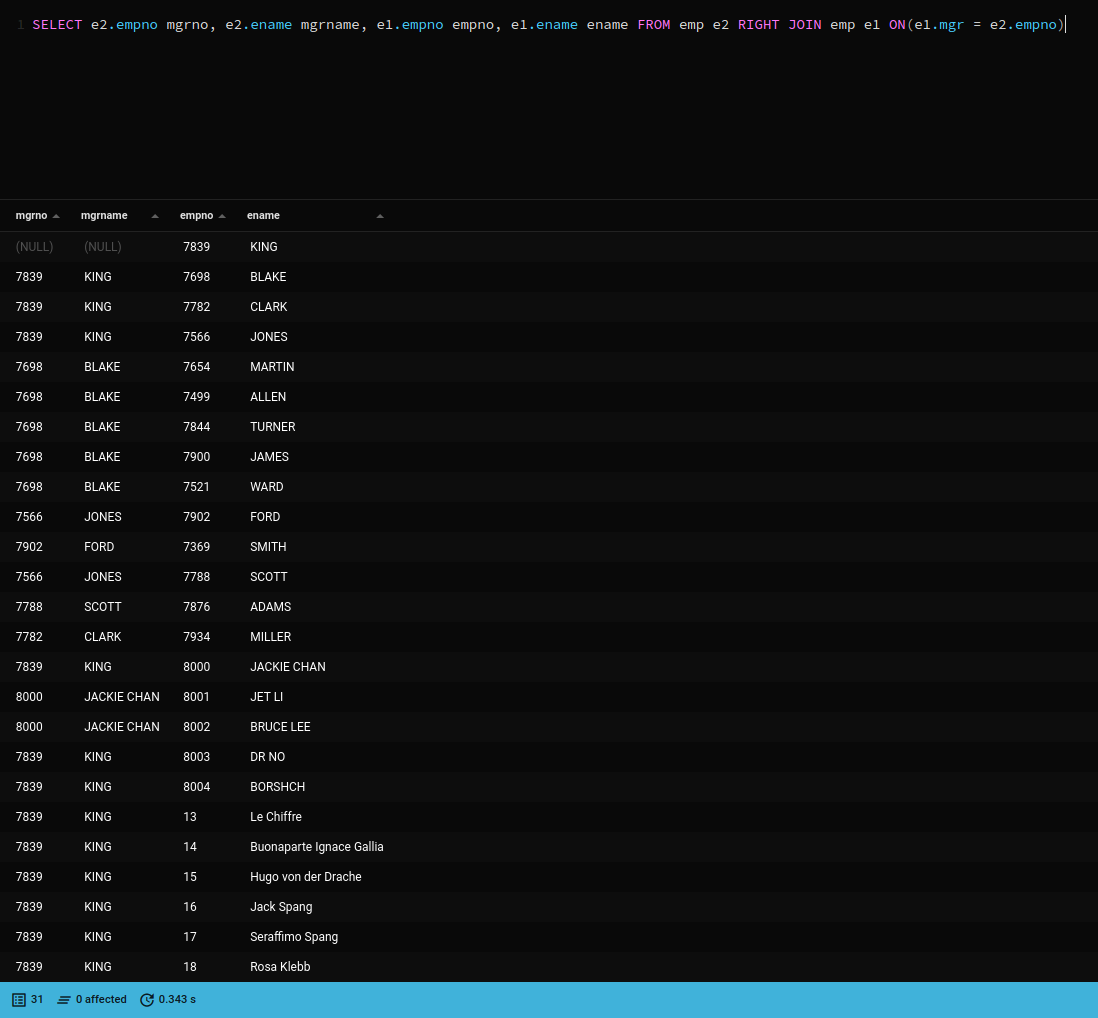
SELECT e1.empno empno, e1.ename ename, e2.empno mgrno, e2.ename mgrname FROM emp e1 INNER JOIN emp e2 ON(e1.mgr = e2.empno)

Left (Right) Join  
5. 📷 Виконайте завдання двічі (використовуючи LEFT та RIGHT JOIN) Сформувати запит, що виводить номер та прізвище працівника, номер та прізвище його менеджера, включаючи працівників, які не мають менеджерів.

SELECT e1.empno empno, e1.ename ename, e2.empno mgrno, e2.ename mgrname FROM emp e1 LEFT JOIN emp e2 ON(e1.mgr = e2.empno)



SELECT e2.empno mgrno, e2.ename mgrname, e1.empno empno, e1.ename ename FROM emp e2 RIGHT JOIN emp e1 ON(e1.mgr = e2.empno)



6. Сформуйте запит, який повертає номер та назву відділу, в якому немає співробітників.

SELECT dept.deptno, dept.dname FROM dept

LEFT JOIN emp ON (emp.deptno = dept.deptno)

WHERE emp.empno is NULL;

7. \* Сформулюйте запит, який повертає співробітників, які не мають підлеглих

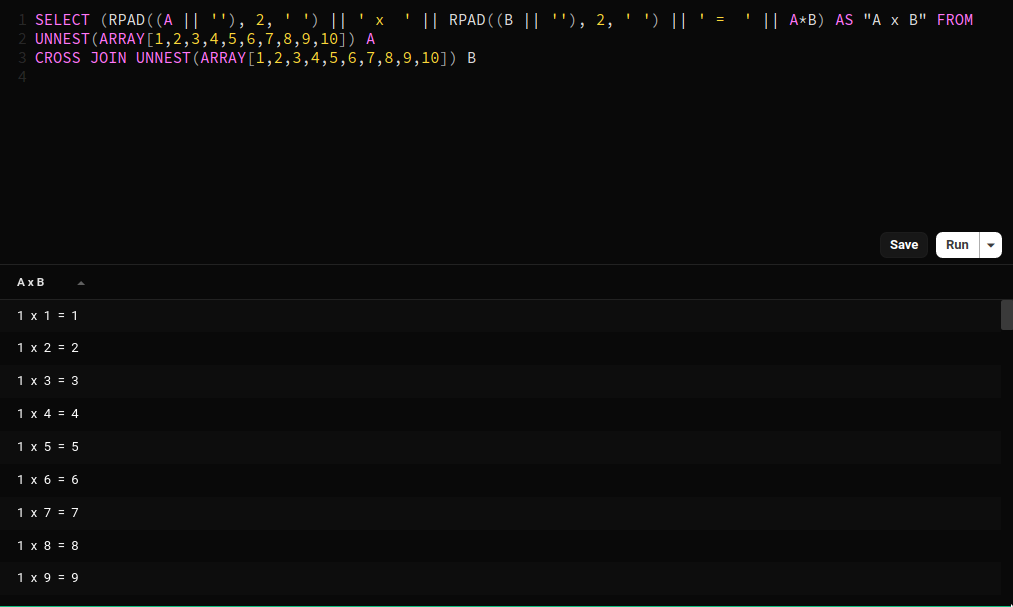
SELECT e2.\* FROM emp e1 RIGHT JOIN emp e2 ON (e1.mgr = e2.empno) WHERE e1.mgr is NULL

Cross Join  
8. 📷 Виведіть на екран таблицю множення для чисел 1 до 10. (Можливо для цього доведеться створити додаткову таблицю, незабудьте її видалити)  
(для «\*» - зверніть увагу на вирівнювання)

SELECT (RPAD((A || ''), 2, ' ') || ' x ' || RPAD((B || ''), 2, ' ') || ' = ' || A\*B) AS "A x B" FROM

UNNEST(ARRAY[1,2,3,4,5,6,7,8,9,10]) A

CROSS JOIN UNNEST(ARRAY[1,2,3,4,5,6,7,8,9,10]) B



Використання кількох Join в одному запиті  
9. Виведете імена співробітників, у яких начальники працюють в інших містах.

SELECT worker.ename FROM emp worker

JOIN emp manager ON (worker.mgr = manager.empno)

JOIN dept d1 ON (d1.deptno = worker.deptno)

JOIN dept d2 ON (d2.deptno = manager.deptno AND d1.loc != d2.loc)

10. Виведіть імена начальників, у яких співробітники працюють в інших містах.

SELECT DISTINCT manager.ename FROM emp manager

JOIN emp worker ON (manager.empno = worker.mgr)

JOIN dept d1 ON (d1.deptno = worker.deptno)

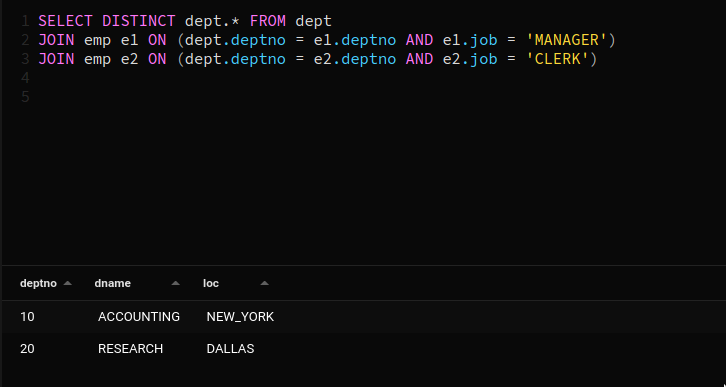
JOIN dept d2 ON (d2.deptno = manager.deptno AND d1.loc != d2.loc)

11. \* 📷 Виведіть відділ в якому працює і менеджер і клерк (менеджер та клерк – посади)

SELECT DISTINCT dept.\* FROM dept

JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')

JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')



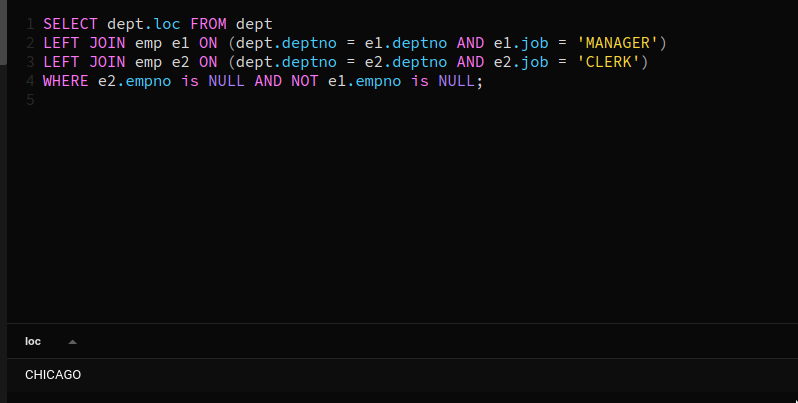
12. \* 📷 Виведіть відділ в якому працює і менеджер але нема клерка.

SELECT dept.loc FROM dept

LEFT JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')

LEFT JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')

WHERE e2.empno is NULL AND NOT e1.empno is NULL;



13. \* 📷 Виведіть місто в якому працює і менеджер і клерк (у одному місті може бути декілька відділів)

SELECT DISTINCT d1.loc FROM dept d1

JOIN dept d2 ON (d1.loc = d2.loc)

JOIN emp e1 ON (d1.deptno = e1.deptno AND e1.job = 'MANAGER')

JOIN emp e2 ON (d2.deptno = e2.deptno AND e2.job = 'CLERK')



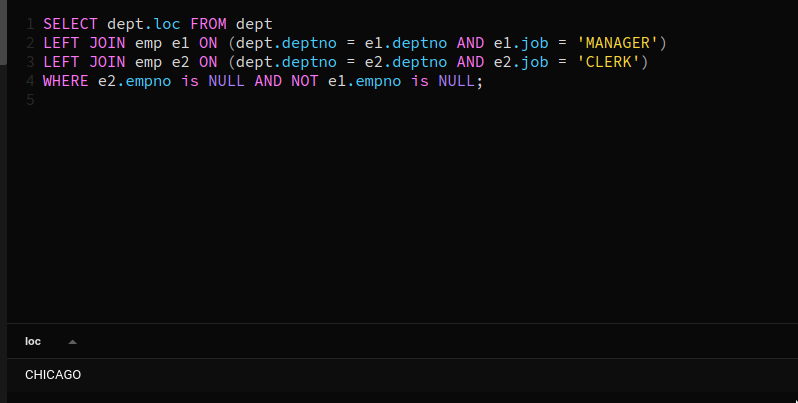
14. \*\* 📷 Виведіть відділ в якому працює і менеджер але нема клерка.

SELECT dept.loc FROM dept

LEFT JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')

LEFT JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')

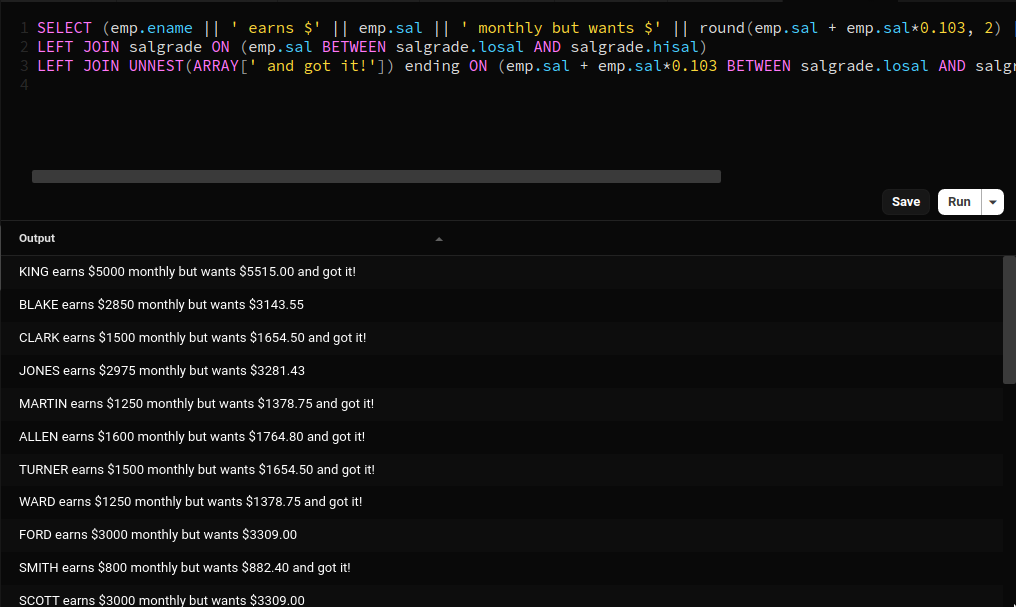
WHERE e2.empno is NULL AND NOT e1.empno is NULL;



Тренувальні вправи  
15. \* 📷 Заповніть таблицю Salegrade значеннями.  
Керівник може підвищити зарплату співробітнику у межах його Salegrade. Наприклад, якщо зарплата співробітника 1405, то його зарплата належить 3-му тарифному розряду від 1401 до 2000. Керівник може збільшити її лише до 2000.  
Опитування показало, що співробітники хочуть отримувати на 10,3% більше, ніж зараз. Створіть звіт, який для кожного співробітника поверне рядок (1 стовпчик):  
<Ename> earns $<salary> monthly but wants $<salary+10.3% > and got it!  
якщо зарплату можна отримати, і  
<Ename> earns $<salary> monthly but wants $<salary+10.3%>  
якщо плату підвищити не можна.  
Наприклад:  
SMITH earns $800.00 monthly but wants $882.40 and got it!  
Зарплату виводити з точністю до 2-х знаків після коми (тобто у виведенні має бути 2400.00, а не 2400).

SELECT (emp.ename || ' earns $' || emp.sal || ' monthly but wants $' || round(emp.sal + emp.sal\*0.103, 2) || COALESCE(ending, ' ')) as "Output" FROM emp LEFT JOIN salgrade ON (emp.sal BETWEEN salgrade.losal AND salgrade.hisal)

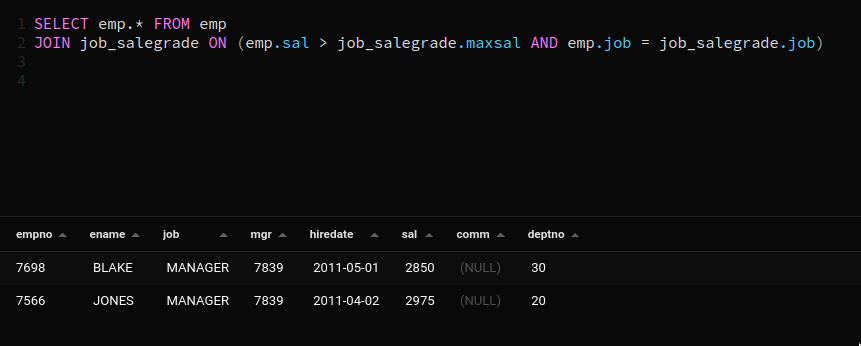
LEFT JOIN UNNEST(ARRAY[' and got it!']) ending ON (emp.sal + emp.sal\*0.103 BETWEEN salgrade.losal AND salgrade.hisal);



Завдання по варіантах, частина 1  
\* Варіант 1 «Межі» 📷  
Створіть таблицю Job\_Salegrade   
1. Знайдіть співробітників, які отримують більше, ніж максимальна зарплата за посадою

SELECT emp.\* FROM emp

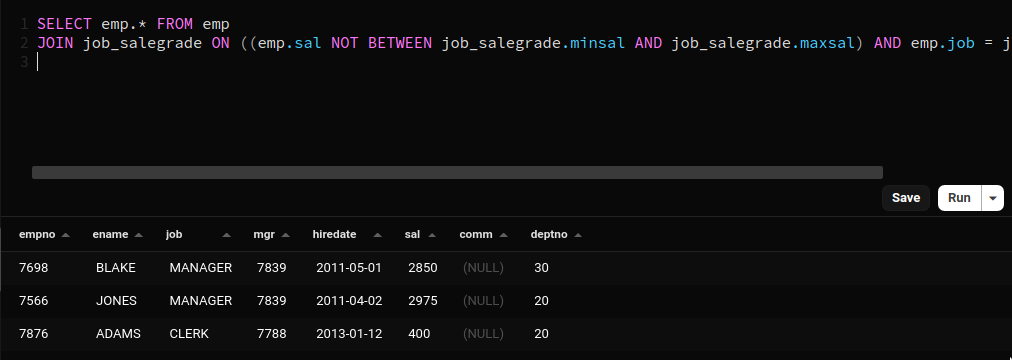
JOIN job\_salegrade ON (emp.sal > job\_salegrade.maxsal AND emp.job = job\_salegrade.job)



2. Знайдіть співробітників, які отримують більше або менше встановлених меж зарплати

SELECT emp.\* FROM emp

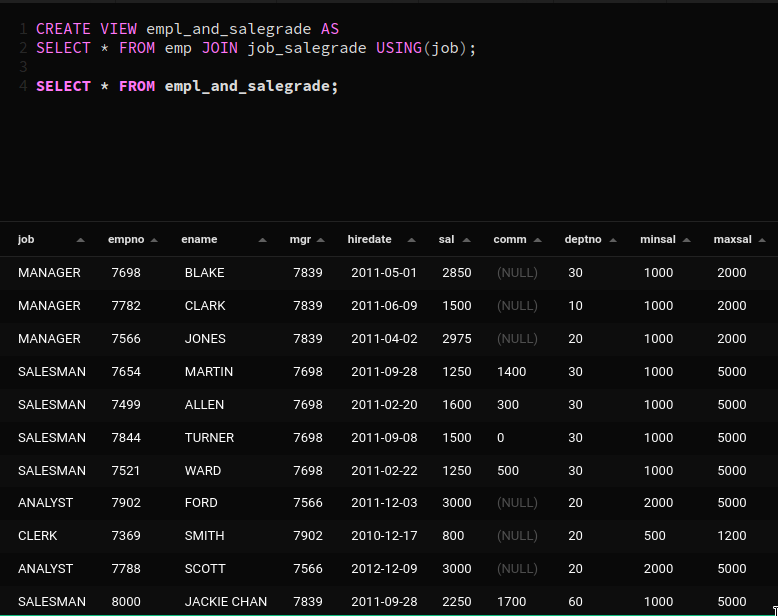
JOIN job\_salegrade ON ((emp.sal NOT BETWEEN job\_salegrade.minsal AND job\_salegrade.maxsal) AND emp.job = job\_salegrade.job)



3. Створіть уявлення (view) empl\_and\_salegrade на основі об'єднання таблиць Job\_Salegrage та emp.

CREATE VIEW empl\_and\_salegrade AS

SELECT \* FROM emp JOIN job\_salegrade USING(job);



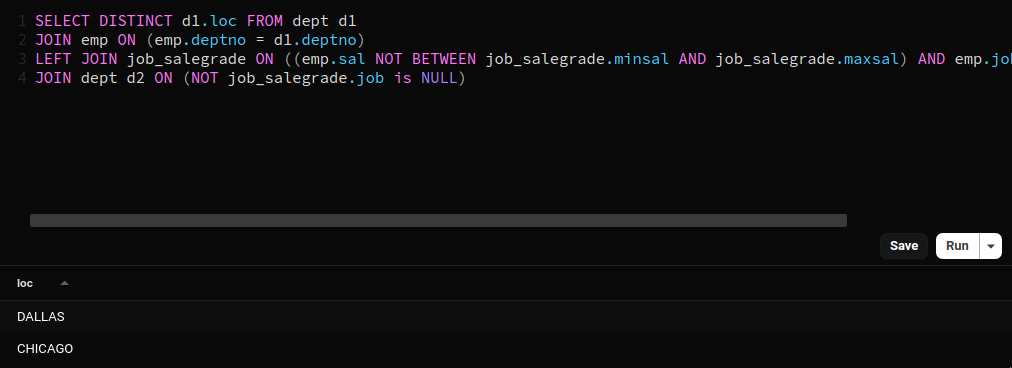
4. Визначте, у яких містах службовцям недоплачують чи переплачують:

SELECT DISTINCT d1.loc FROM dept d1

JOIN emp ON (emp.deptno = d1.deptno)

LEFT JOIN job\_salegrade ON ((emp.sal NOT BETWEEN job\_salegrade.minsal AND job\_salegrade.maxsal) AND emp.job = job\_salegrade.job)

JOIN dept d2 ON (NOT job\_salegrade.job is NULL)



5. Визначте, у яких містах усі службовці одержують зарплату у встановлених межах.

SELECT DISTINCT d1.loc FROM empl\_and\_salegrade e1

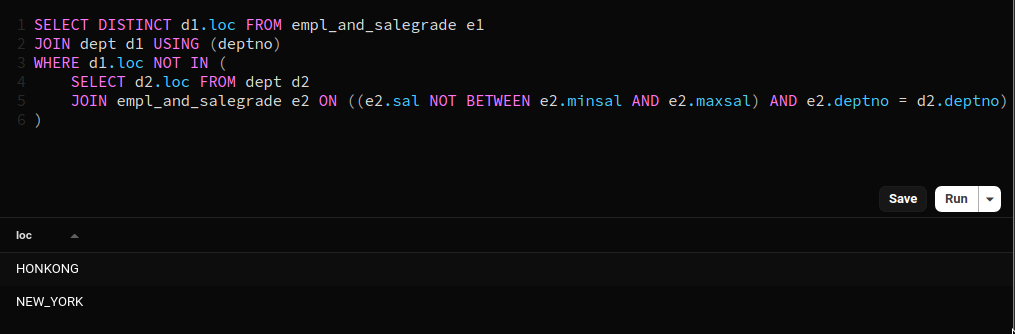
JOIN dept d1 USING (deptno)

WHERE d1.loc NOT IN (

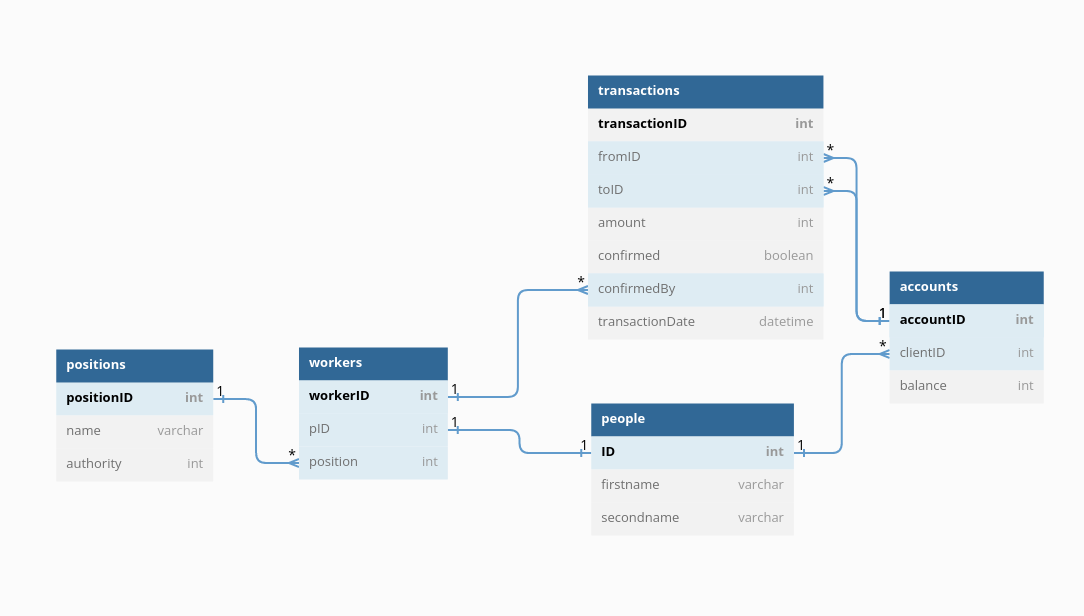
SELECT d2.loc FROM dept d2

JOIN empl\_and\_salegrade e2 ON ((e2.sal NOT BETWEEN e2.minsal AND e2.maxsal) AND e2.deptno = d2.deptno)

)



Завдання за варіантами, частина 2  
\*\* Варіант 1 📷  
Обов’язково наведіть Вашу ER-діаграму.



Оберіть буд-які 5 завдань з переліку.

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

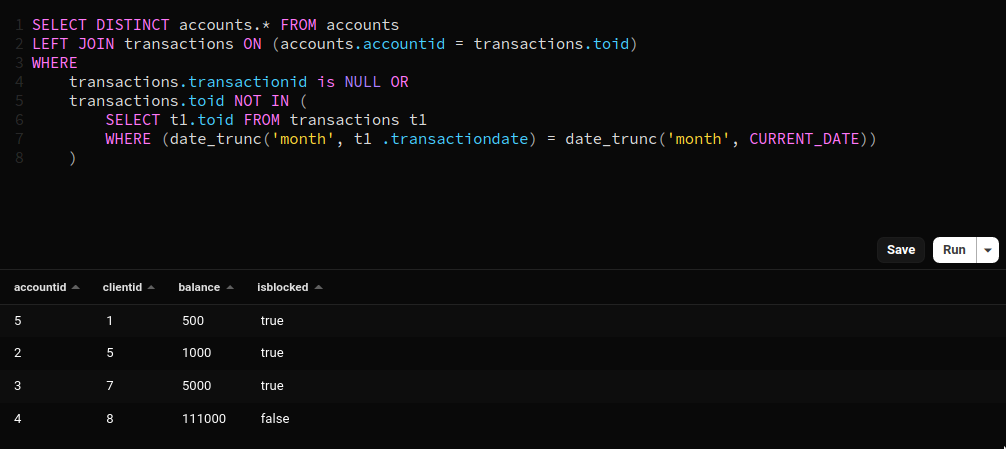
transactions.transactionid is NULL OR

transactions.toid NOT IN (

SELECT t1.toid FROM transactions t1

WHERE (date\_trunc('month', t1 .transactiondate) = date\_trunc('month', CURRENT\_DATE))

)

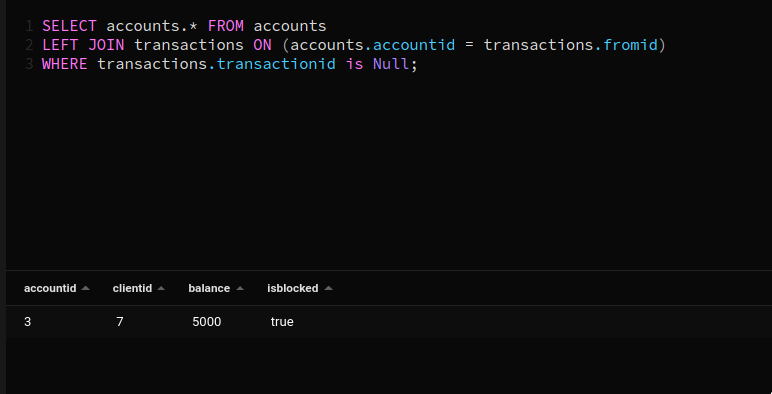


3. «Скрудж». Знайдіть рахунки, на які гроші виключно нараховуються, але не витрачаються.

SELECT accounts.\* FROM accounts

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

WHERE transactions.transactionid is Null;



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

SELECT people.\* FROM people

JOIN workers ON (people.id = workers.pid)

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

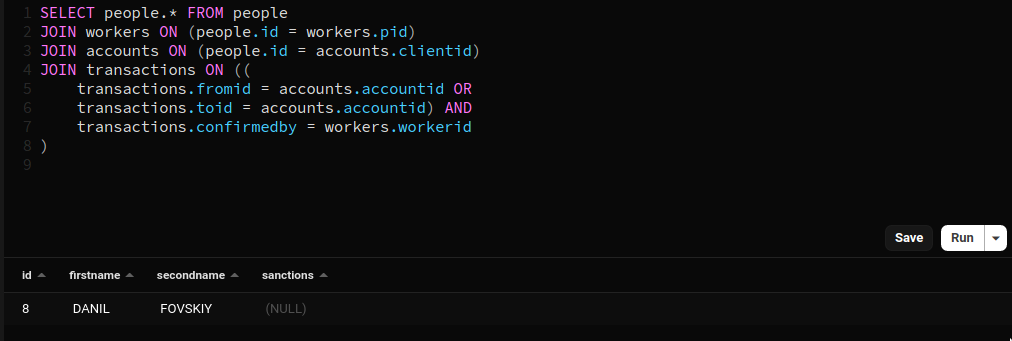
JOIN transactions ON ((

transactions.fromid = accounts.accountid OR

transactions.toid = accounts.accountid) AND

transactions.confirmedby = workers.workerid

)

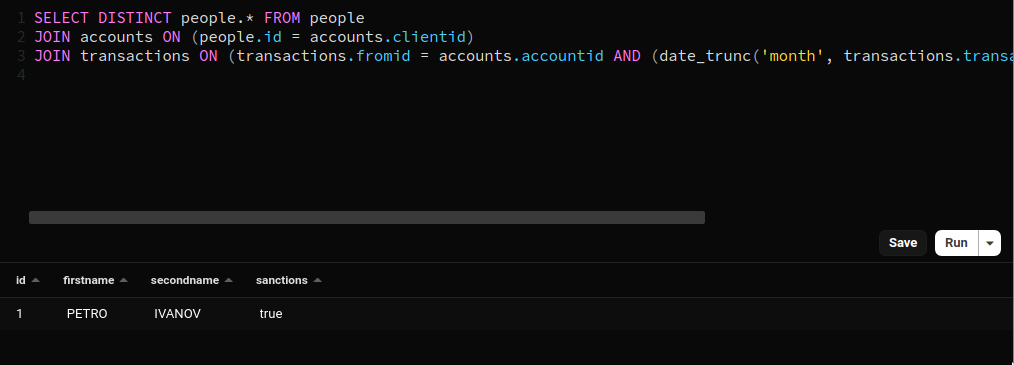


6. Виведіть людей, які витрачали гроші цього місяця.

SELECT DISTINCT people.\* FROM people

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

JOIN transactions ON (transactions.fromid = accounts.accountid AND (date\_trunc('month', transactions.transactiondate) = date\_trunc('month', CURRENT\_DATE)))



12. Люди (всі: співробітники, клієнти, уповноважені особи), які не мають рахунку у банку.

SELECT people.\* FROM people

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

WHERE accounts.accountid is Null;

