

# 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.

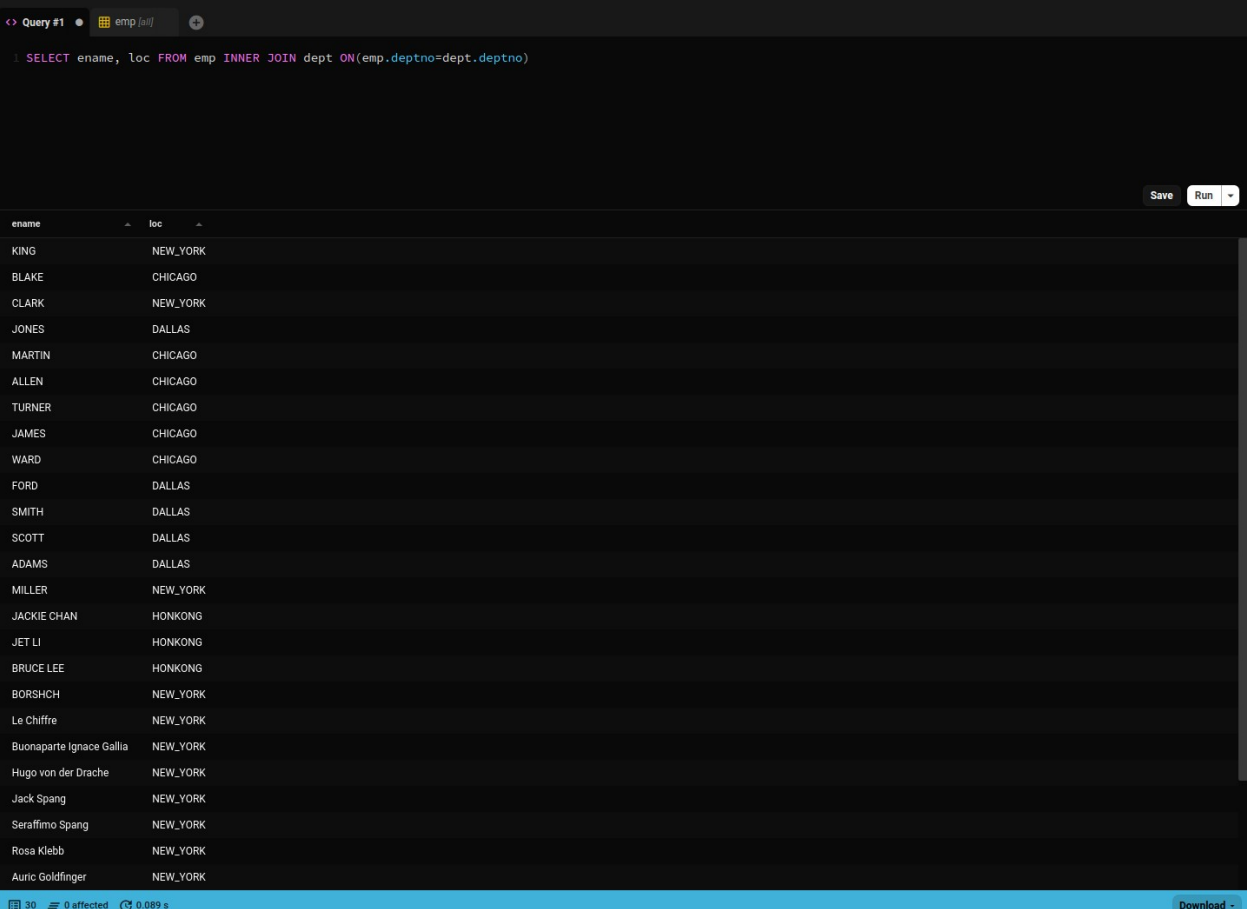
Для завдань/розділів відмічених « » – скріншот – обов'язково. Для інших – за домовленістю із викладачем.

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)
```



The screenshot shows a SQL query execution interface. The query is: `SELECT ename, loc FROM emp INNER JOIN dept ON (emp.deptno=dept.deptno)`. The results are displayed in a table with two columns: `ename` and `loc`. The table contains 20 rows of data, listing employee names and their locations. The interface includes a 'Query #1' tab, a 'Run' button, and a status bar at the bottom indicating 30 rows affected and a execution time of 0.089 s.

ename	loc
KING	NEW_YORK
BLAKE	CHICAGO
CLARK	NEW_YORK
JONES	DALLAS
MARTIN	CHICAGO
ALLEN	CHICAGO
TURNER	CHICAGO
JAMES	CHICAGO
WARD	CHICAGO
FORD	DALLAS
SMITH	DALLAS
SCOTT	DALLAS
ADAMS	DALLAS
MILLER	NEW_YORK
JACKIE CHAN	HONKONG
JET LI	HONKONG
BRUCE LEE	HONKONG
BORSHCH	NEW_YORK
Le Chiffre	NEW_YORK
Buonaparte Ignace Gallia	NEW_YORK
Hugo von der Drache	NEW_YORK
Jack Spang	NEW_YORK
Seraffimo Spang	NEW_YORK
Rosa Klebb	NEW_YORK
Auric Goldfinger	NEW_YORK

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';
```

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

ename	sal	deptno
JONES	2975	20
FORD	3000	20
SMITH	800	20
SCOTT	3000	20
ADAMS	1100	20

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

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

ename	sal	deptno
JONES	2975	20
FORD	3000	20
SMITH	800	20
SCOTT	3000	20
ADAMS	1100	20

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

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

ename	sal	deptno
JONES	2975	20
FORD	3000	20
SMITH	800	20
SCOTT	3000	20
ADAMS	1100	20

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)
```

```
1 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)|
```

empno	ename	mgrno	mgrname
7839	KING	(NULL)	(NULL)
7698	BLAKE	7839	KING
7782	CLARK	7839	KING
7566	JONES	7839	KING
7654	MARTIN	7698	BLAKE
7499	ALLEN	7698	BLAKE
7844	TURNER	7698	BLAKE
7900	JAMES	7698	BLAKE
7521	WARD	7698	BLAKE
7902	FORD	7566	JONES
7369	SMITH	7902	FORD
7788	SCOTT	7566	JONES
7876	ADAMS	7788	SCOTT
7934	MILLER	7782	CLARK
8000	JACKIE CHAN	7839	KING
8001	JET LI	8000	JACKIE CHAN
8002	BRUCE LEE	8000	JACKIE CHAN
8003	DR NO	7839	KING
8004	BORSHCH	7839	KING
13	Le Chiffre	7839	KING
14	Buonaparte Ignace Gallia	7839	KING
15	Hugo von der Drache	7839	KING
16	Jack Spang	7839	KING
17	Seraffimo Spang	7839	KING
18	Rosa Klebb	7839	KING

31 rows affected (0.348 s)

```
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)
```

```
1 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)
```

mgrno	mgrname	empno	ename
(NULL)	(NULL)	7839	KING
7839	KING	7698	BLAKE
7839	KING	7782	CLARK
7839	KING	7566	JONES
7698	BLAKE	7654	MARTIN
7698	BLAKE	7499	ALLEN
7698	BLAKE	7844	TURNER
7698	BLAKE	7900	JAMES
7698	BLAKE	7521	WARD
7566	JONES	7902	FORD
7902	FORD	7369	SMITH
7566	JONES	7788	SCOTT
7788	SCOTT	7876	ADAMS
7782	CLARK	7934	MILLER
7839	KING	8000	JACKIE CHAN
8000	JACKIE CHAN	8001	JET LI
8000	JACKIE CHAN	8002	BRUCE LEE
7839	KING	8003	DR NO
7839	KING	8004	BORSHCH
7839	KING	13	Le Chiffre
7839	KING	14	Buonaparte Ignace Gallia
7839	KING	15	Hugo von der Drache
7839	KING	16	Jack Spang
7839	KING	17	Seraffimo Spang
7839	KING	18	Rosa Klebb

31 0 affected 0.343 s

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, не забудьте 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
```

```
1 SELECT (RPAD((A || ' '), 2, ' ') || ' x ' || RPAD((B || ' '), 2, ' ') || ' = ' || A*B) AS "A x B" FROM  
2 UNNEST(ARRAY[1,2,3,4,5,6,7,8,9,10]) A  
3 CROSS JOIN UNNEST(ARRAY[1,2,3,4,5,6,7,8,9,10]) B  
4
```

Save Run

A x B

1 x 1 = 1  
1 x 2 = 2  
1 x 3 = 3  
1 x 4 = 4  
1 x 5 = 5  
1 x 6 = 6  
1 x 7 = 7  
1 x 8 = 8  
1 x 9 = 9

## Використання кількох Join в одному запиті

9. Виведіть 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')
```

```
1 SELECT DISTINCT dept.* FROM dept
2 JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')
3 JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')
4
5
```

deptno	dname	loc
10	ACCOUNTING	NEW_YORK
20	RESEARCH	DALLAS

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;
```

```
1 SELECT dept.loc FROM dept  
2 LEFT JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')  
3 LEFT JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')  
4 WHERE e2.empno is NULL AND NOT e1.empno is NULL;  
5
```

loc
CHICAGO

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')
```

```
1 SELECT DISTINCT d1.loc FROM dept d1  
2 JOIN dept d2 ON (d1.loc = d2.loc)  
3 JOIN emp e1 ON (d1.deptno = e1.deptno AND e1.job = 'MANAGER')  
4 JOIN emp e2 ON (d2.deptno = e2.deptno AND e2.job = 'CLERK')  
5
```

loc
DALLAS
NEW_YORK



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;
```

```
1 SELECT dept.loc FROM dept  
2 LEFT JOIN emp e1 ON (dept.deptno = e1.deptno AND e1.job = 'MANAGER')  
3 LEFT JOIN emp e2 ON (dept.deptno = e2.deptno AND e2.job = 'CLERK')  
4 WHERE e2.empno is NULL AND NOT e1.empno is NULL;  
5
```

loc

CHICAGO

## 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);
```

```
1 SELECT (emp.ename || ' earns $' || emp.sal || ' monthly but wants $' || round(emp.sal + emp.sal*0.103, 2)  
2 LEFT JOIN salgrade ON (emp.sal BETWEEN salgrade.losal AND salgrade.hisal)  
3 LEFT JOIN UNNEST(ARRAY[' and got it!']) ending ON (emp.sal + emp.sal*0.103 BETWEEN salgrade.losal AND salg  
4
```

Save Run

Output

KING earns \$5000 monthly but wants \$5515.00 and got it!  
BLAKE earns \$2850 monthly but wants \$3143.55  
CLARK earns \$1500 monthly but wants \$1654.50 and got it!  
JONES earns \$2975 monthly but wants \$3281.43  
MARTIN earns \$1250 monthly but wants \$1378.75 and got it!  
ALLEN earns \$1600 monthly but wants \$1764.80 and got it!  
TURNER earns \$1500 monthly but wants \$1654.50 and got it!  
WARD earns \$1250 monthly but wants \$1378.75 and got it!  
FORD earns \$3000 monthly but wants \$3309.00  
SMITH earns \$800 monthly but wants \$882.40 and got it!  
SCOTT earns \$3000 monthly but wants \$3309.00

# 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)
```

```
1 SELECT emp.* FROM emp  
2 JOIN job_salegrade ON (emp.sal > job_salegrade.maxsal AND emp.job = job_salegrade.job)  
3  
4
```

empno	ename	job	mgr	hiredate	sal	comm	deptno
7698	BLAKE	MANAGER	7839	2011-05-01	2850	(NULL)	30
7566	JONES	MANAGER	7839	2011-04-02	2975	(NULL)	20

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)
```

```
1 SELECT emp.* FROM emp  
2 JOIN job_salegrade ON ((emp.sal NOT BETWEEN job_salegrade.minsal AND job_salegrade.maxsal) AND emp.job = j  
3 |
```

Save Run

empno	ename	job	mgr	hiredate	sal	comm	deptno
7698	BLAKE	MANAGER	7839	2011-05-01	2850	(NULL)	30
7566	JONES	MANAGER	7839	2011-04-02	2975	(NULL)	20
7876	ADAMS	CLERK	7788	2013-01-12	400	(NULL)	20

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

```
CREATE VIEW empl_and_salegrade AS  
SELECT * FROM emp JOIN job_salegrade USING(job);
```

```
1 CREATE VIEW empl_and_salegrade AS  
2 SELECT * FROM emp JOIN job_salegrade USING(job);  
3  
4 SELECT * FROM empl_and_salegrade;
```

job	empno	ename	mgr	hiredate	sal	comm	deptno	minsal	maxsal
MANAGER	7698	BLAKE	7839	2011-05-01	2850	(NULL)	30	1000	2000
MANAGER	7782	CLARK	7839	2011-06-09	1500	(NULL)	10	1000	2000
MANAGER	7566	JONES	7839	2011-04-02	2975	(NULL)	20	1000	2000
SALESMAN	7654	MARTIN	7698	2011-09-28	1250	1400	30	1000	5000
SALESMAN	7499	ALLEN	7698	2011-02-20	1600	300	30	1000	5000
SALESMAN	7844	TURNER	7698	2011-09-08	1500	0	30	1000	5000
SALESMAN	7521	WARD	7698	2011-02-22	1250	500	30	1000	5000
ANALYST	7902	FORD	7566	2011-12-03	3000	(NULL)	20	2000	5000
CLERK	7369	SMITH	7902	2010-12-17	800	(NULL)	20	500	1200
ANALYST	7788	SCOTT	7566	2012-12-09	3000	(NULL)	20	2000	5000
SALESMAN	8000	JACKIE CHAN	7839	2011-09-28	2250	1700	60	1000	5000

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)
```

```
1 SELECT DISTINCT d1.loc FROM dept d1  
2 JOIN emp ON (emp.deptno = d1.deptno)  
3 LEFT JOIN job_salegrade ON ((emp.sal NOT BETWEEN job_salegrade.minsal AND job_salegrade.maxsal) AND emp.job  
4 JOIN dept d2 ON (NOT job_salegrade.job is NULL)
```

Save Run

loc
DALLAS
CHICAGO

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)
)
```

```
1 SELECT DISTINCT d1.loc FROM empl_and_salegrade e1
2 JOIN dept d1 USING (deptno)
3 WHERE d1.loc NOT IN (
4     SELECT d2.loc FROM dept d2
5     JOIN empl_and_salegrade e2 ON ((e2.sal NOT BETWEEN e2.minsal AND e2.maxsal) AND e2.deptno = d2.deptno)
6 )
```

Save

Run

loc

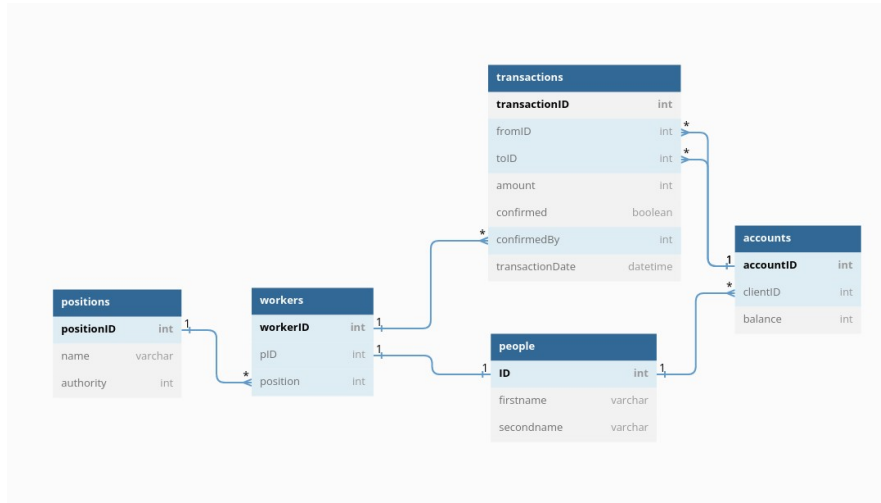
HONKONG

NEW\_YORK

# 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))
    )
  
```

```

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 Run

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

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;
```

```
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. 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  
)
```

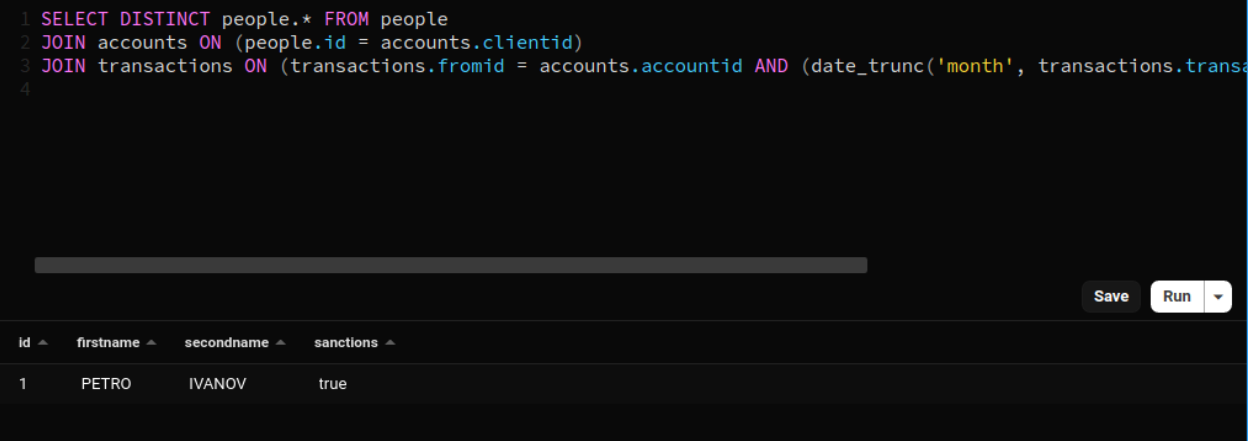
```
1 SELECT people.* FROM people  
2 JOIN workers ON (people.id = workers.pid)  
3 JOIN accounts ON (people.id = accounts.clientid)  
4 JOIN transactions ON ((  
5     transactions.fromid = accounts.accountid OR  
6     transactions.toid = accounts.accountid) AND  
7     transactions.confirmedby = workers.workerid  
8 )  
9
```

Save Run

id ^	firstname ^	secondname ^	sanctions ^
8	DANIL	FOVSKIY	(NULL)

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)))
```



The screenshot shows a SQL query editor with the following code:

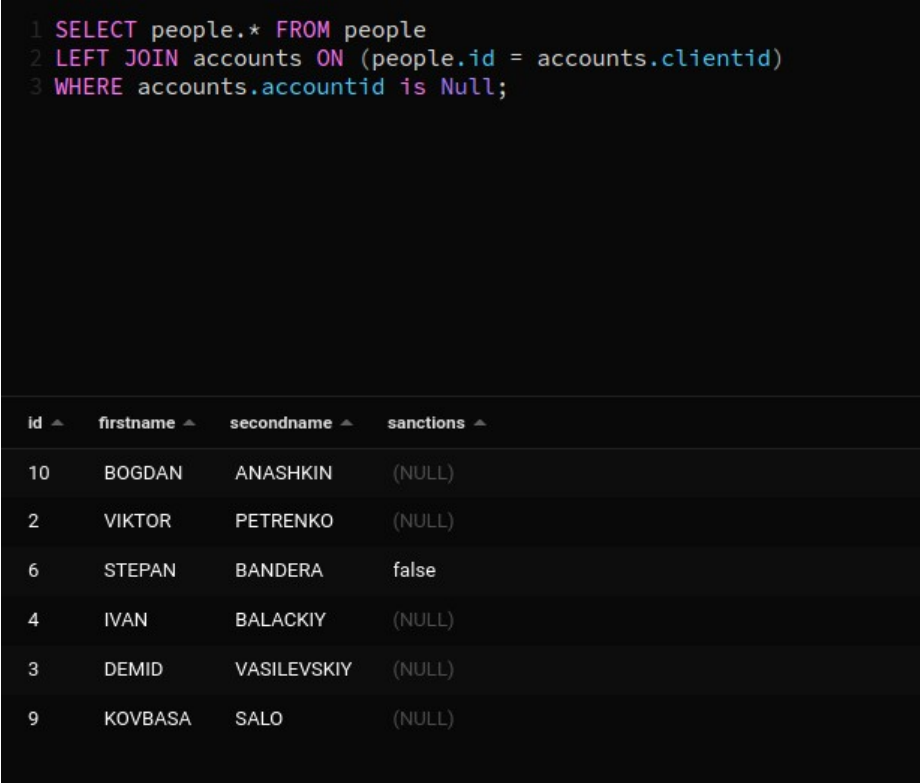
```
1 SELECT DISTINCT people.* FROM people
2 JOIN accounts ON (people.id = accounts.clientid)
3 JOIN transactions ON (transactions.fromid = accounts.accountid AND (date_trunc('month', transactions.transa
4
```

Below the editor is a results table with the following data:

id	firstname	secondname	sanctions
1	PETRO	IVANOV	true

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;
```



The screenshot shows a SQL query editor with the following code:

```
1 SELECT people.* FROM people
2 LEFT JOIN accounts ON (people.id = accounts.clientid)
3 WHERE accounts.accountid is Null;
```

Below the editor is a results table with the following data:

id	firstname	secondname	sanctions
10	BOGDAN	ANASHKIN	(NULL)
2	VIKTOR	PETRENKO	(NULL)
6	STEPAN	BANDERA	false
4	IVAN	BALACKIY	(NULL)
3	DEMID	VASILEVSKIY	(NULL)
9	KOVBASA	SALO	(NULL)