Пороскун Олена Олегівна ПМ-81

Лабораторна робота 8

Natural join

1. Створіть звіт який містить ім'я співробітника і місто, в якому він працює.

**select ename name, loc city**

**from emp**

**natural join dept;**

****

2. Виконайте завдання 2 тричі: використовуючи

NATURAL JOIN, JOIN ... ON і JOIN ... USING:

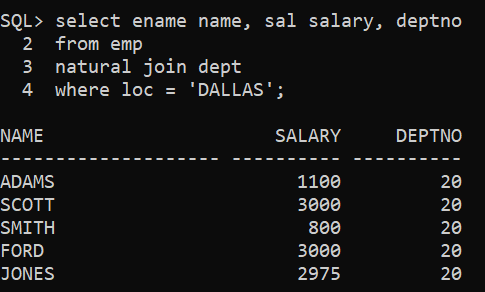
Відділу кадрів потрібен звіт, якій містить ім'я, зарплату, номер відділу для всіх співробітників з Dallas.

**select ename name, sal salary, deptno**

**from emp**

**natural join dept**

**where loc = 'DALLAS';**

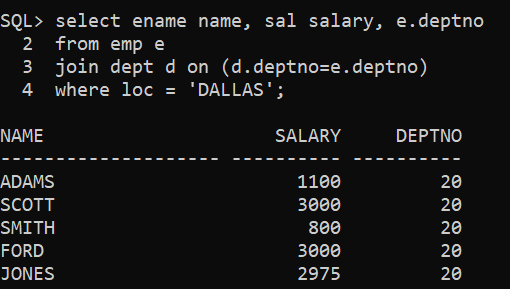


**select ename name, sal salary, e.deptno**

**from emp e**

**join dept d on (d.deptno=e.deptno)**

**where loc = 'DALLAS';**

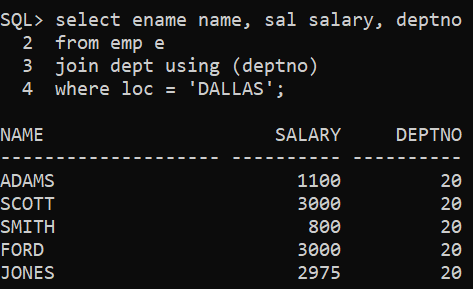


**select ename name, sal salary, deptno**

**from emp e**

**join dept using (deptno)**

**where loc = 'DALLAS';**



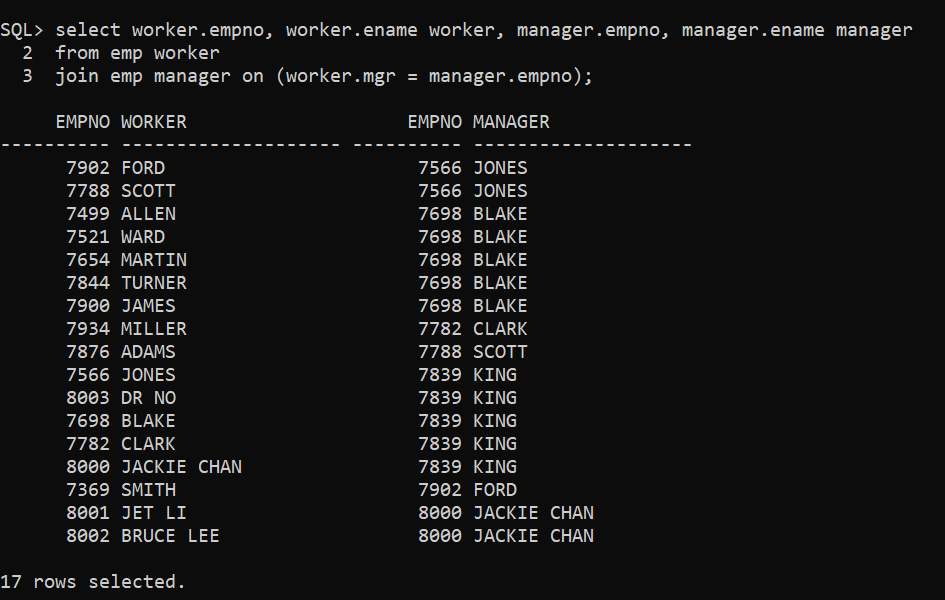
Join з використанням (On або Using)

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

**select worker.empno, worker.ename worker, manager.empno, manager.ename manager**

**from emp worker**

**join emp manager on (worker.mgr = manager.empno);**



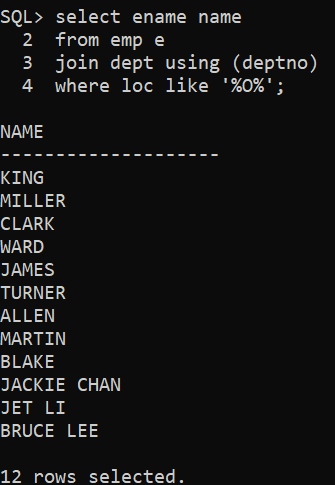
4. Сформувати запит, що виводить співробітників, що працюють в місті, назва якого містить 'O'.

**select ename name**

**from emp e**

**join dept using (deptno)**

**where loc like '%O%';**

****

Left (Right) Join

Виконайте завдання 5 двічі: використовуючи LEFT JOIN і RIGHT JOIN.

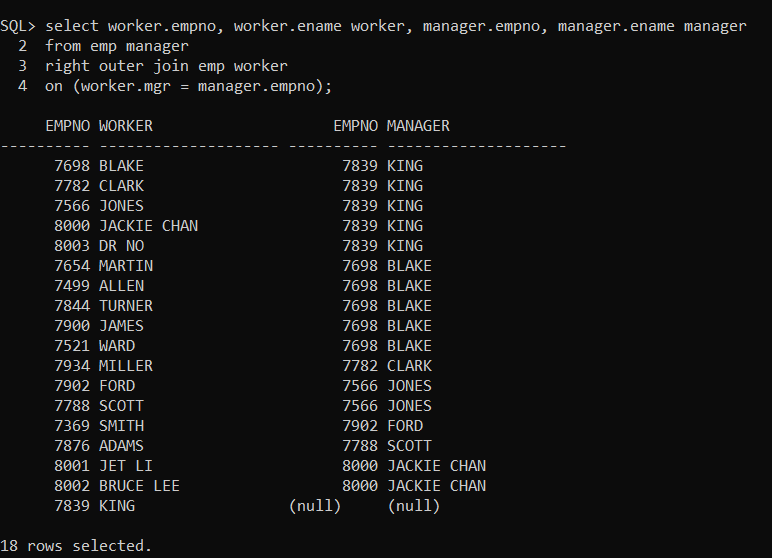
5. Сформувати запит, що виводить номер і прізвище працівника, номер і прізвище його менеджера, включно з працівниками, які не мають менеджерів.

**select worker.empno, worker.ename worker, manager.empno, manager.ename manager**

**from emp manager**

**right outer join emp worker**

**on (worker.mgr = manager.empno);**

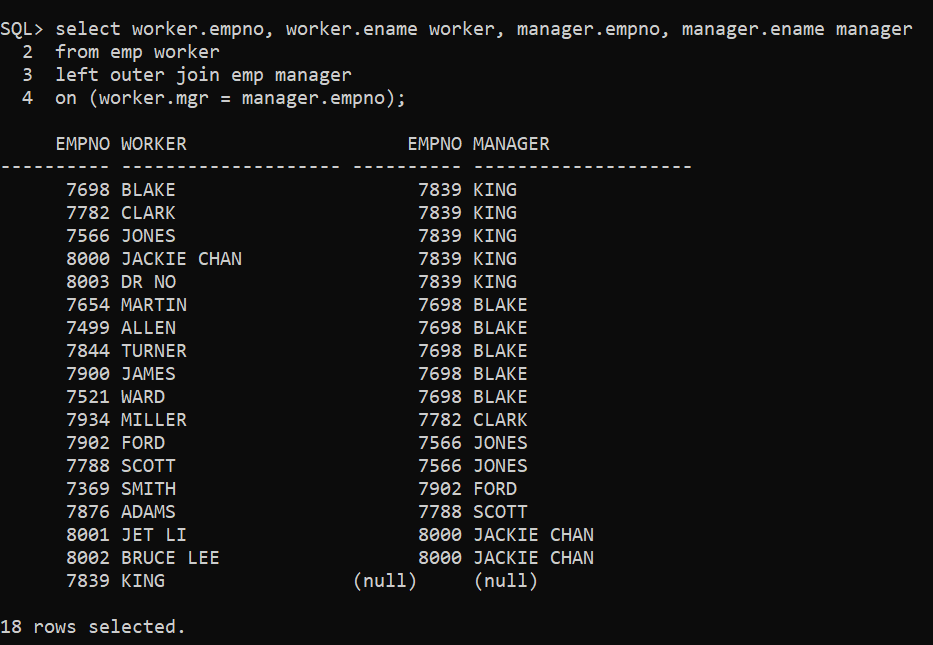


**select worker.empno, worker.ename worker, manager.empno, manager.ename manager**

**from emp worker**

**left outer join emp manager**

**on (worker.mgr = manager.empno);**

****

6. Створіть запит, який повертає номер і назву відділу, в якому немає співробітників.

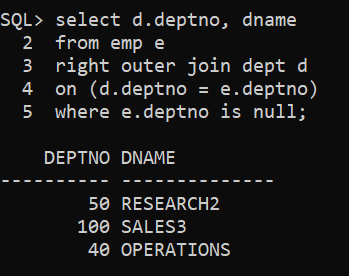
**select d.deptno, dname**

**from emp e**

**right outer join dept d**

**on (d.deptno = e.deptno)**

**where e.deptno is null;**

****

7. Створіть запит, який повертає співробітників, у яких немає підлеглих.

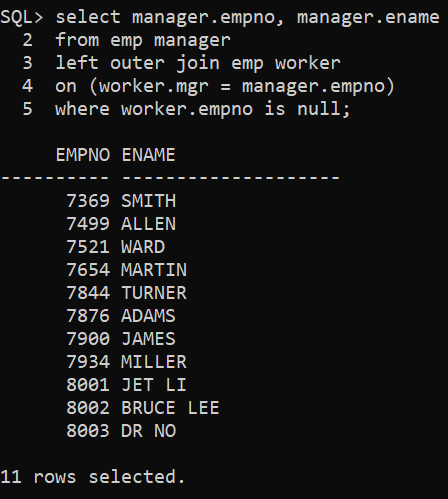
**select manager.empno, manager.ename**

**from emp manager**

**left outer join emp worker**

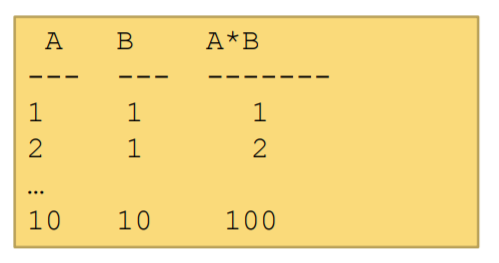
**on (worker.mgr = manager.empno)**

**where worker.empno is null;**

****

Cross Join

8. Виведіть на екран таблицю множення для чисел від 1 до 10. (Можливо для цього вам доведеться створити додаткову таблицю. Не забудьте її видалити)

****

**create table my\_table(num integer default 1);**

**insert into my\_table values (1);**

**insert into my\_table values (2);**

**insert into my\_table values (3);**

**insert into my\_table values (4);**

**insert into my\_table values (5);**

**insert into my\_table values (6);**

**insert into my\_table values (7);**

**insert into my\_table values (8);**

**insert into my\_table values (9);**

**insert into my\_table values (10);**

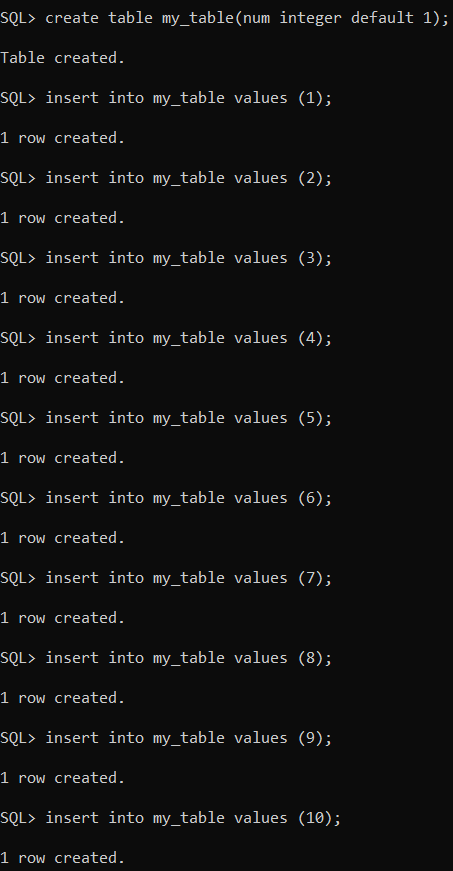
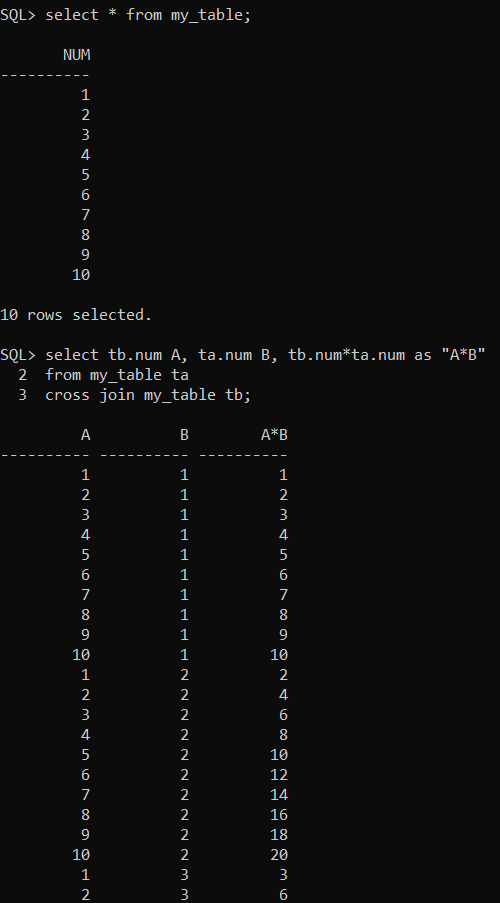
**select \* from my\_table;**

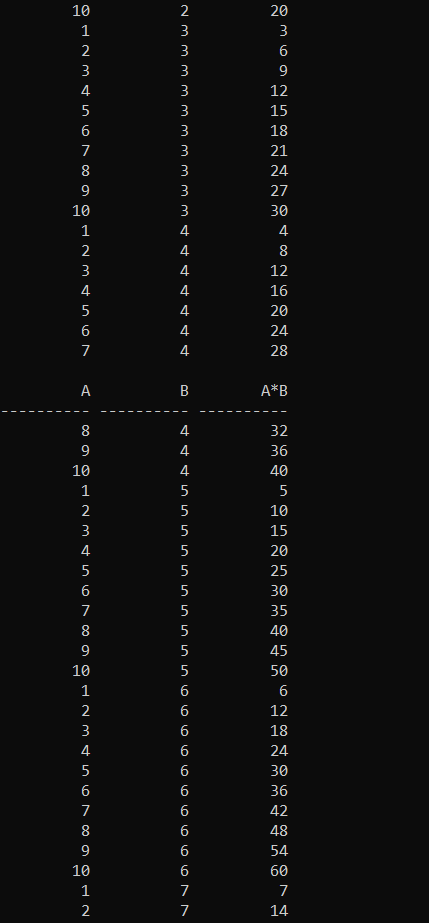
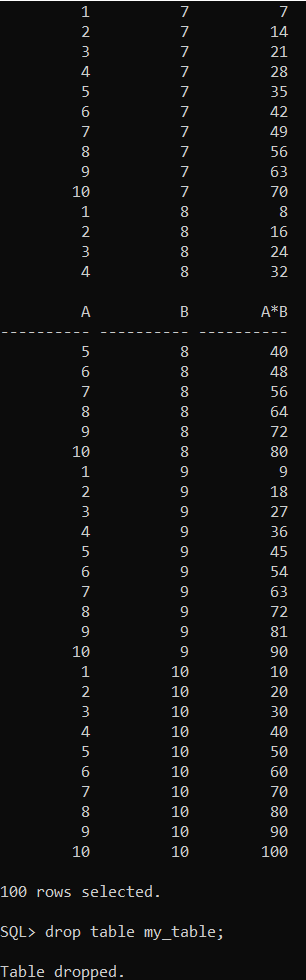
**select tb.num A, ta.num B, tb.num\*ta.num as "A\*B"**

**from my\_table ta**

**cross join my\_table tb;**

**drop table my\_table;**

Використання декількох Join в одному запиті

9. Виведіть імена співробітників, у яких начальники працюють в інших містах.

**select worker.ename worker**

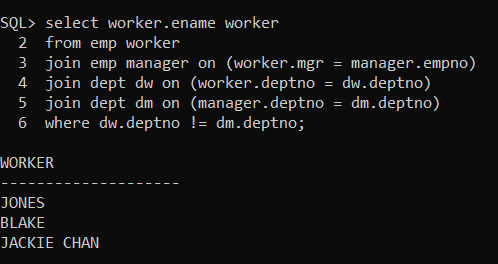
**from emp worker**

**join emp manager on (worker.mgr = manager.empno)**

**join dept dw on (worker.deptno = dw.deptno)**

**join dept dm on (manager.deptno = dm.deptno)**

**where dw.deptno != dm.deptno;**

****

Перевірка:

**select worker.empno, worker.ename worker, dw.loc, manager.empno, manager.ename manager, dm.loc**

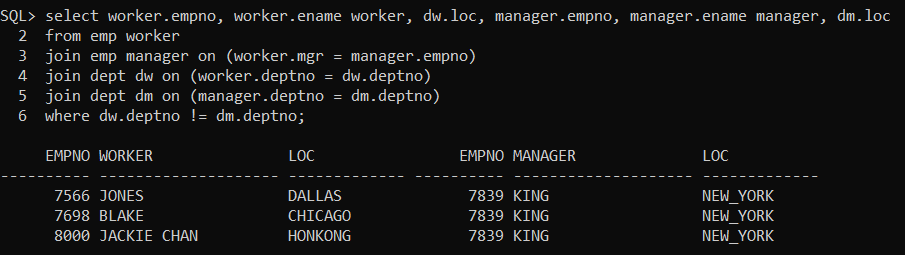
**from emp worker**

**join emp manager on (worker.mgr = manager.empno)**

**join dept dw on (worker.deptno = dw.deptno)**

**join dept dm on (manager.deptno = dm.deptno)**

**where dw.deptno != dm.deptno;**

****

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

**select manager.ename manager**

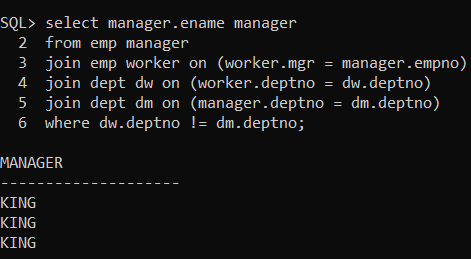
**from emp manager**

**join emp worker on (worker.mgr = manager.empno)**

**join dept dw on (worker.deptno = dw.deptno)**

**join dept dm on (manager.deptno = dm.deptno)**

**where dw.deptno != dm.deptno;**

****

Перевірка:

**select worker.empno, worker.ename worker, dw.loc, manager.empno, manager.ename manager, dm.loc**

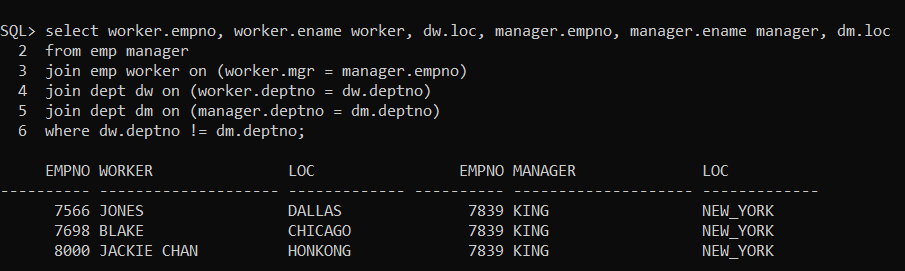
**from emp manager**

**join emp worker on (worker.mgr = manager.empno)**

**join dept dw on (worker.deptno = dw.deptno)**

**join dept dm on (manager.deptno = dm.deptno)**

**where dw.deptno != dm.deptno;**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Комплексне завдання 1. «Межі»

1. Створіть таблицю Job\_Salegrade:



**create table Job\_Salegrade(job varchar(20) not null,**

**min\_salary NUMBER(4) not null,**

**max\_salary NUMBER(4) not null);**

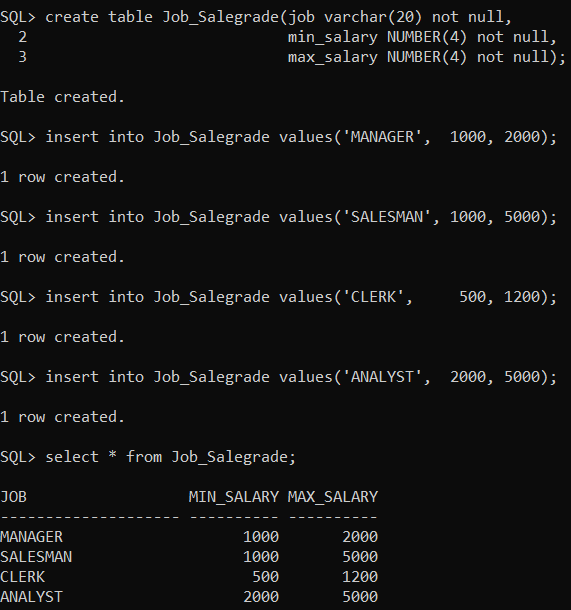
**insert into Job\_Salegrade values('MANAGER', 1000, 2000);**

**insert into Job\_Salegrade values('SALESMAN', 1000, 5000);**

**insert into Job\_Salegrade values('CLERK', 500, 1200);**

**insert into Job\_Salegrade values('ANALYST', 2000, 5000);**

**select \* from Job\_Salegrade;**

****

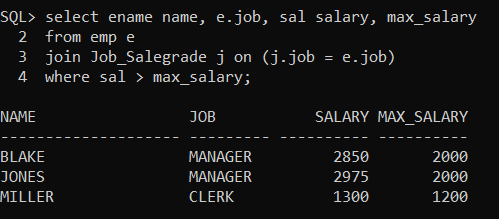
2. Знайдіть співробітників, які отримують більше, ніж максимальна зарплата за посадою.

**select ename name, e.job, sal salary, max\_salary**

**from emp e**

**join Job\_Salegrade j on (j.job = e.job)**

**where sal > max\_salary;**

****

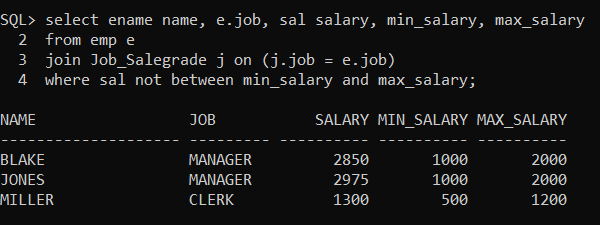
3. Знайдіть співробітників, які отримують більше або менше встановлених меж зарплати.

**select ename name, e.job, sal salary, min\_salary, max\_salary**

**from emp e**

**join Job\_Salegrade j on (j.job = e.job)**

**where sal not between min\_salary and max\_salary;**



4. Створіть представлення (view) empl\_and\_salegrade на основі з'єднання таблиць Job\_Salegrage і emp.

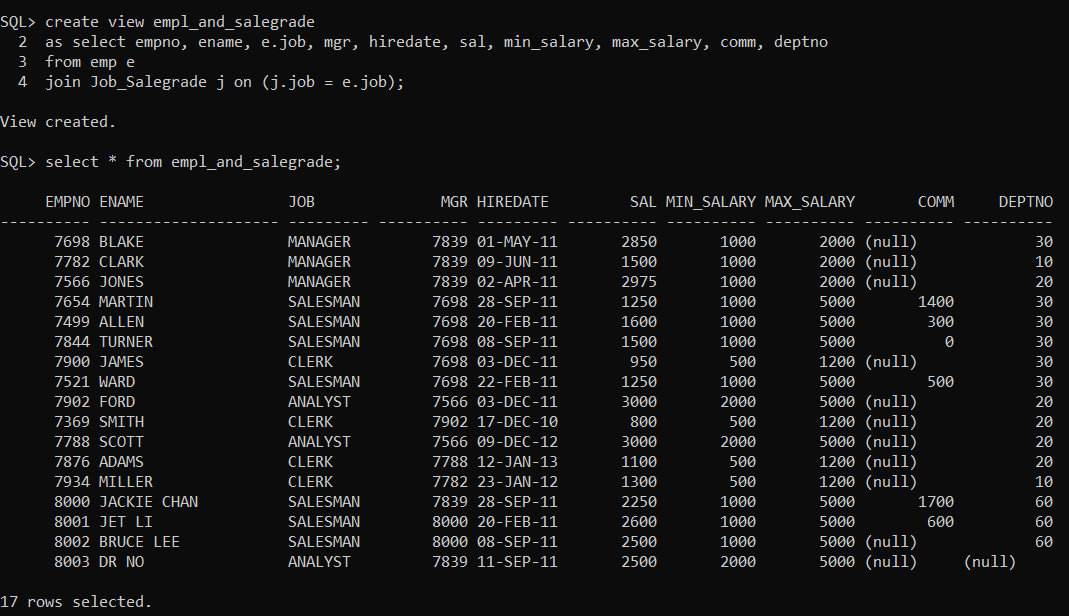
**create view empl\_and\_salegrade**

**as select empno, ename, e.job, mgr, hiredate, sal, min\_salary, max\_salary, comm, deptno**

**from emp e**

**join Job\_Salegrade j on (j.job = e.job);**

**select \* from empl\_and\_salegrade;**



5. Визначте, в яких містах службовцям недоплачують або переплачують:

А) використовуючи з'єднання таблиць

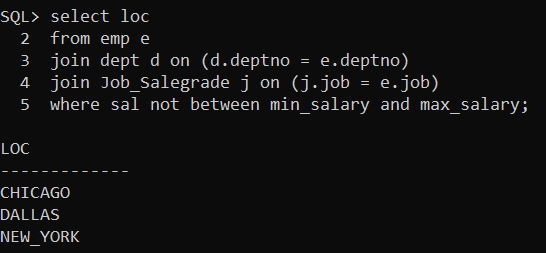
**select loc**

**from emp e**

**join dept d on (d.deptno = e.deptno)**

**join Job\_Salegrade j on (j.job = e.job)**

**where sal not between min\_salary and max\_salary;**

****

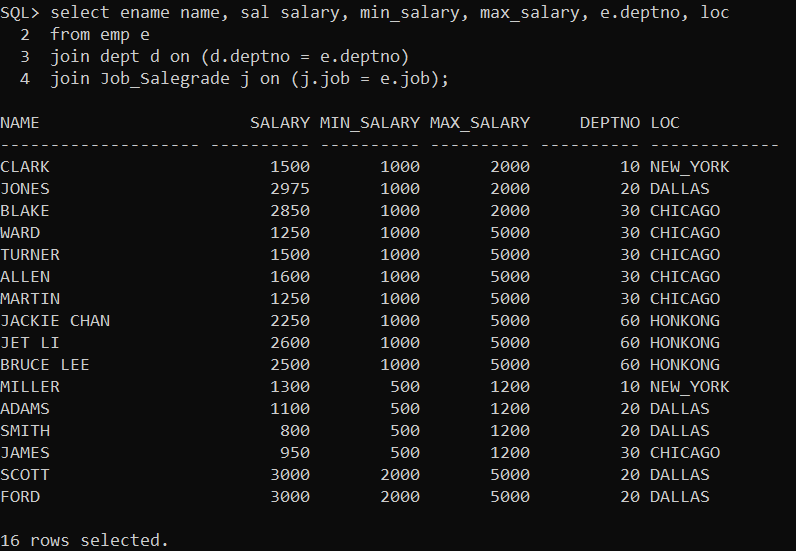
Перевірка:

**select ename name, sal salary, min\_salary, max\_salary, e.deptno, loc**

**from emp e**

**join dept d on (d.deptno = e.deptno)**

**join Job\_Salegrade j on (j.job = e.job);**

****

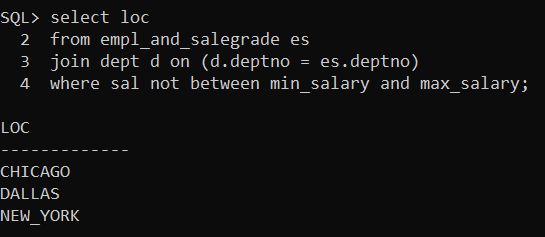
Б) використовуючи таблицю dept і представлення empl\_and\_salegrade.

**select loc**

**from empl\_and\_salegrade es**

**join dept d on (d.deptno = es.deptno)**

**where sal not between min\_salary and max\_salary;**

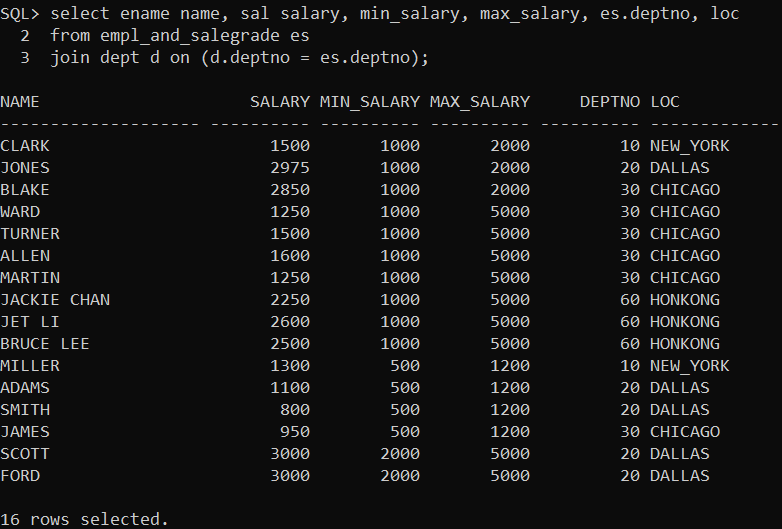
****

Перевірка:

**select ename name, sal salary, min\_salary, max\_salary, es.deptno, loc**

**from empl\_and\_salegrade es**

**join dept d on (d.deptno = es.deptno);**

****

6. Визначте, в яких містах все службовці отримують зарплату в встановлених межах.

**select loc**

**from empl\_and\_salegrade es1**

**join dept d on (d.deptno = es1.deptno)**

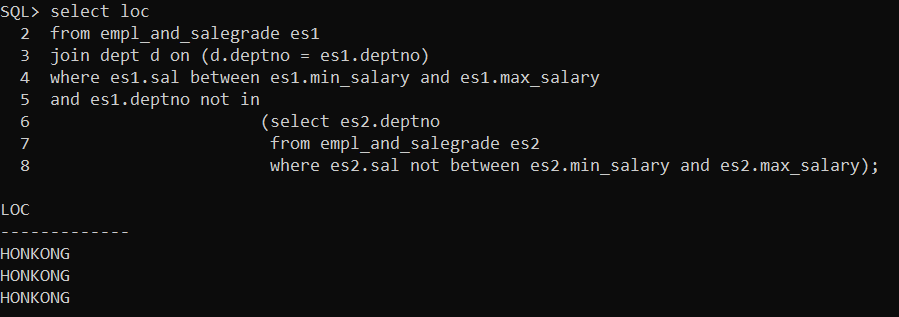
**where es1.sal between es1.min\_salary and es1.max\_salary**

**and es1.deptno not in**

**(select es2.deptno**

**from empl\_and\_salegrade es2**

**where es2.sal not between es2.min\_salary and es2.max\_salary);**

****

Перевірка:

**select es1.ename name, es1.sal salary, es1.min\_salary, es1.max\_salary, es1.deptno, loc**

**from empl\_and\_salegrade es1**

**join dept d on (d.deptno = es1.deptno)**

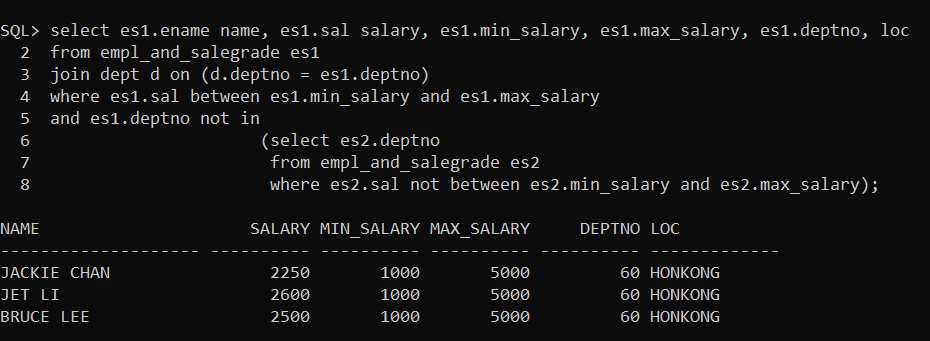
**where es1.sal between es1.min\_salary and es1.max\_salary**

**and es1.deptno not in**

**(select es2.deptno**

**from empl\_and\_salegrade es2**

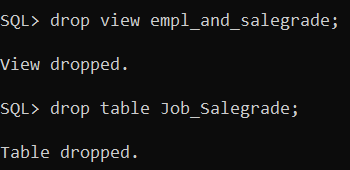
**where es2.sal not between es2.min\_salary and es2.max\_salary);**

****

7. Видаліть представлення empl\_and\_salegrade і таблицю Job\_Salegrade.

drop view empl\_and\_salegrade;

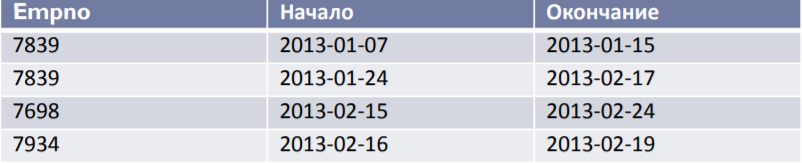
drop table Job\_Salegrade;



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Комплексне завдання 2. «Відпустки»

1. Створіть таблицю Vocation(vacation**)**, приклад файлу доданий до завдання.



vocation. sql:

**CREATE TABLE vacation (empno number, start\_date date, end\_date date);**

**INSERT INTO vacation VALUES (7839, TO\_DATE ('2013-01-07', 'YYYY-MM-DD'), TO\_DATE ('2013-01-15', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7839, TO\_DATE ('2013-01-24', 'YYYY-MM-DD'), TO\_DATE ('2013-02-17', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7698, TO\_DATE ('2013-02-05', 'YYYY-MM-DD'), TO\_DATE ('2013-02-24', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7934, TO\_DATE ('2013-02-16', 'YYYY-MM-DD'), TO\_DATE ('2013-02-19', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7839, TO\_DATE ('2013-01-07', 'YYYY-MM-DD'), TO\_DATE ('2013-01-08', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7782, TO\_DATE ('2015-01-03', 'YYYY-MM-DD'), TO\_DATE ('2015-03-16', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7566, TO\_DATE ('2015-02-07', 'YYYY-MM-DD'), TO\_DATE ('2015-03-08', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (8000, TO\_DATE ('2015-01-19', 'YYYY-MM-DD'), TO\_DATE ('2015-04-05', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7900, TO\_DATE ('2015-03-01', 'YYYY-MM-DD'), TO\_DATE ('2015-03-20', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7844, TO\_DATE ('2015-02-25', 'YYYY-MM-DD'), TO\_DATE ('2015-03-16', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7844, TO\_DATE ('2013-03-07', 'YYYY-MM-DD'), TO\_DATE ('2013-04-14', 'YYYY-MM-DD'));**

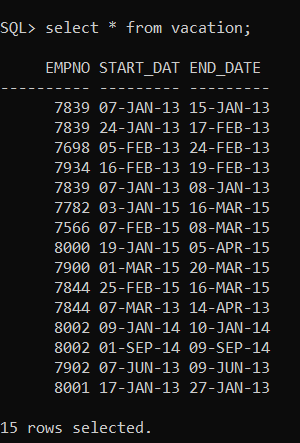
**INSERT INTO vacation VALUES (8002, TO\_DATE ('2014-01-09', 'YYYY-MM-DD'), TO\_DATE ('2014-01-10', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (8002, TO\_DATE ('2014-09-01', 'YYYY-MM-DD'), TO\_DATE ('2014-09-09', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (7902, TO\_DATE ('2013-06-07', 'YYYY-MM-DD'), TO\_DATE ('2013-06-09', 'YYYY-MM-DD'));**

**INSERT INTO vacation VALUES (8001, TO\_DATE ('2013-01-17', 'YYYY-MM-DD'), TO\_DATE ('2013-01-27', 'YYYY-MM-DD'));**

**select \* from vacation;**

****

2. Знайдіть керівників, які сьогодні у відпустці?

**select mn.empno, mn.ename manager**

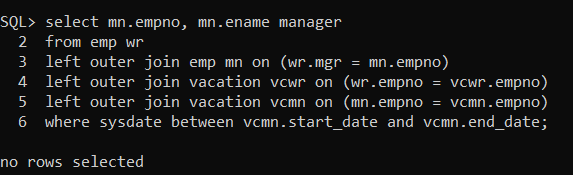
**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**where sysdate between vcmn.start\_date and vcmn.end\_date;**



Перевірка:

**select wr.empno, wr.ename worker, wr.mgr, vcwr.start\_date, vcwr.end\_date,**

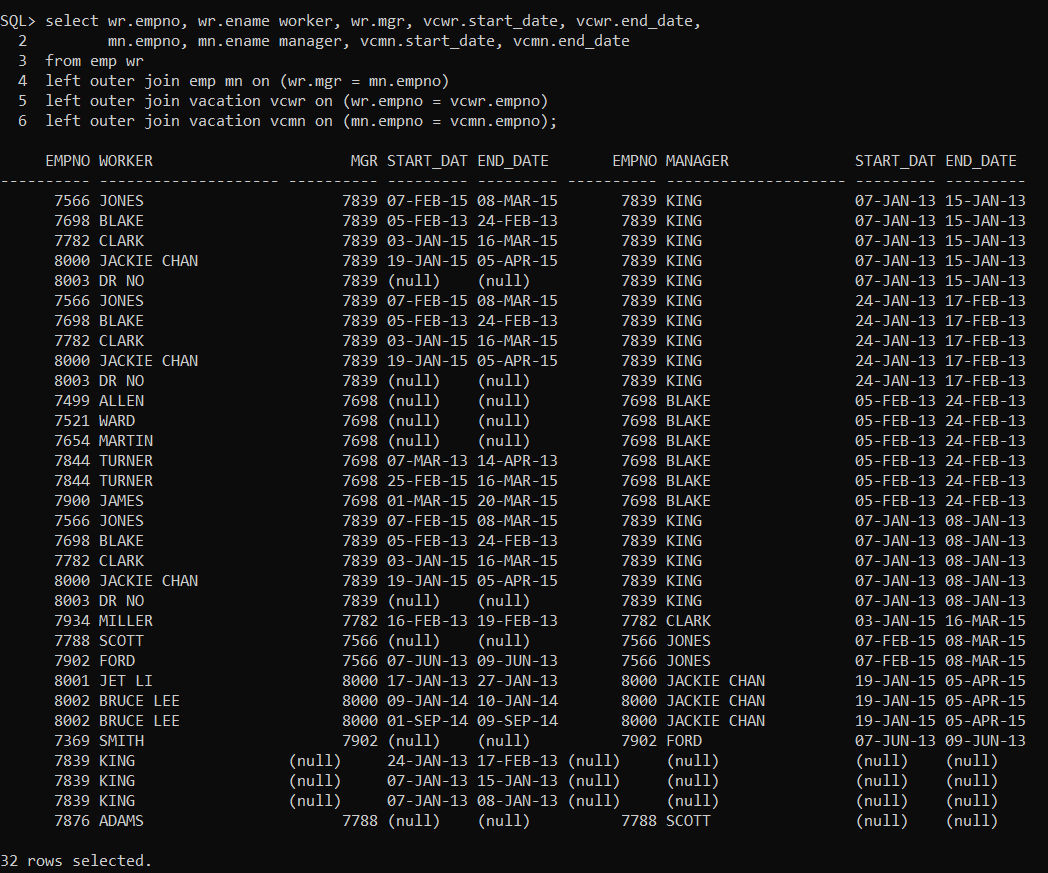
**mn.empno, mn.ename manager, vcmn.start\_date, vcmn.end\_date**

**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno);**

****

3. В яких містах сьогодні керівники у відпустці?

**select dm.loc**

**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

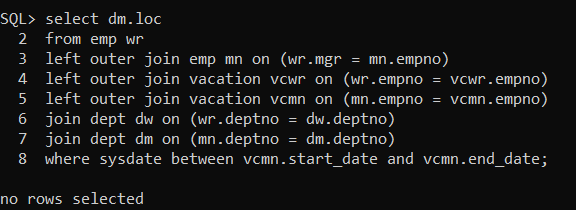
**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**join dept dw on (wr.deptno = dw.deptno)**

**join dept dm on (mn.deptno = dm.deptno)**

**where sysdate between vcmn.start\_date and vcmn.end\_date;**

****

Перевірка:

**select wr.empno, wr.ename worker, wr.mgr, vcwr.start\_date, vcwr.end\_date, dw.loc,**

**mn.empno, mn.ename manager, vcmn.start\_date, vcmn.end\_date , dm.loc**

**from emp wr**

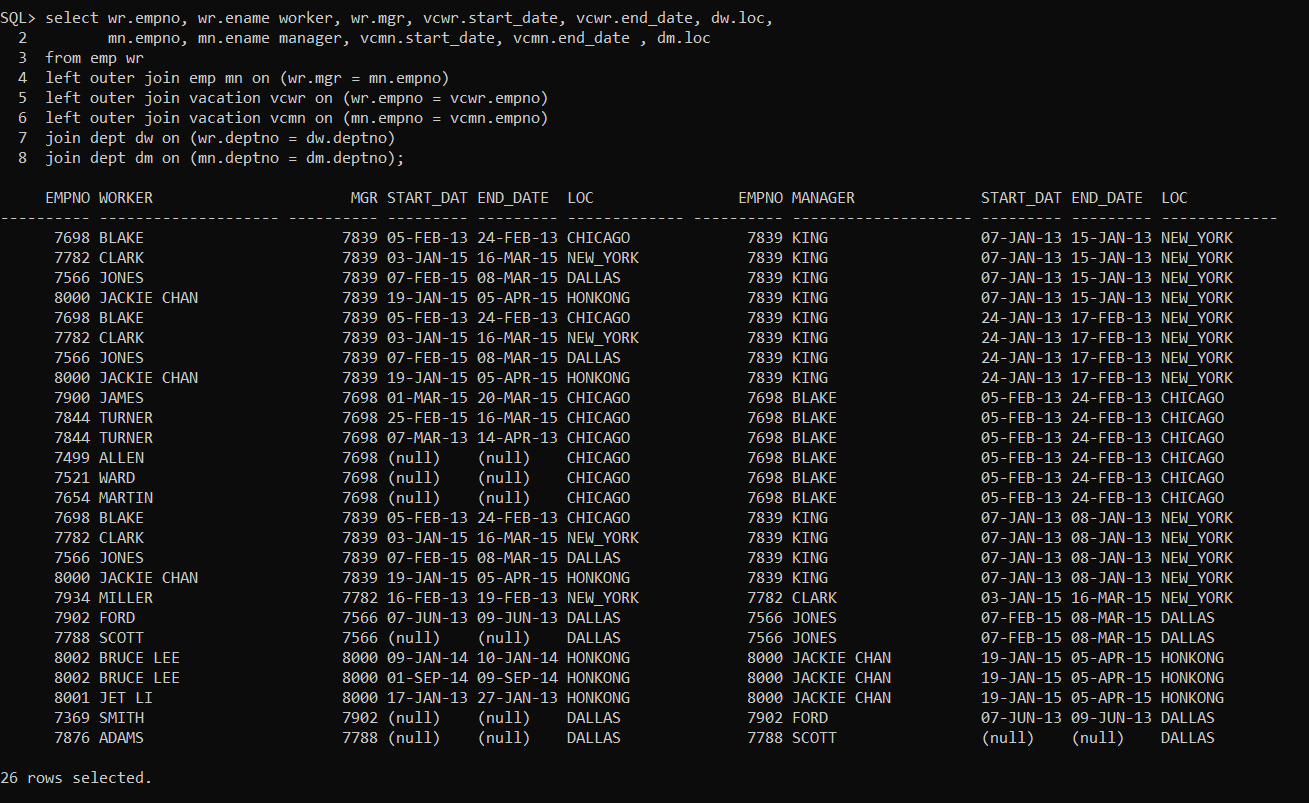
**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**join dept dw on (wr.deptno = dw.deptno)**

**join dept dm on (mn.deptno = dm.deptno);**

****

4. У яких співробітників керівники сьогодні у відпустці?

**select wr.empno, wr.ename worker**

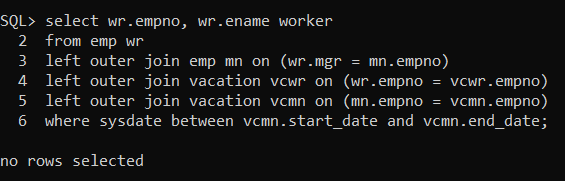
**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**where sysdate between vcmn.start\_date and vcmn.end\_date;**



5. У яких керівників є співробітники, що зараз у відпустці?

**select mn.empno, mn.ename manager**

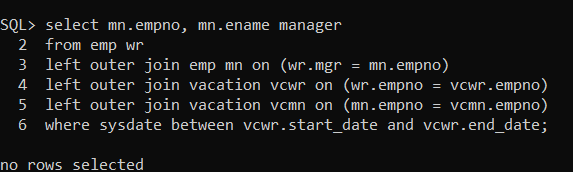
**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**where sysdate between vcwr.start\_date and vcwr.end\_date;**



6. Які співробітники були в цьому місяці у відпустці, але отримують премію?

**select wr.empno, wr.ename worker**

**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

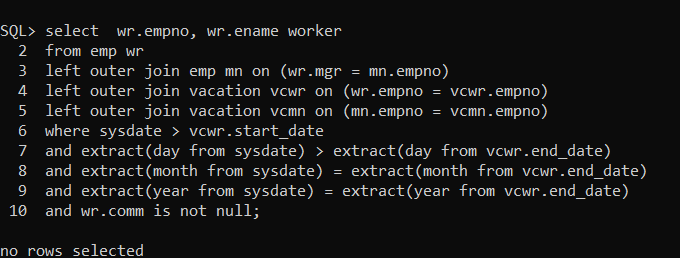
**where sysdate > vcwr.start\_date**

**and extract(day from sysdate) > extract(day from vcwr.end\_date)**

**and extract(month from sysdate) = extract(month from vcwr.end\_date)**

**and extract(year from sysdate) = extract(year from vcwr.end\_date)**

**and wr.comm is not null;**

****

Перевірка:

**select wr.empno, wr.ename worker, wr.comm, sysdate, extract(month from sysdate) mon\_sys,**

**vcwr.start\_date, extract(month from vcwr.start\_date) month\_start,**

**vcwr.end\_date, extract(month from vcwr.end\_date) month\_end**

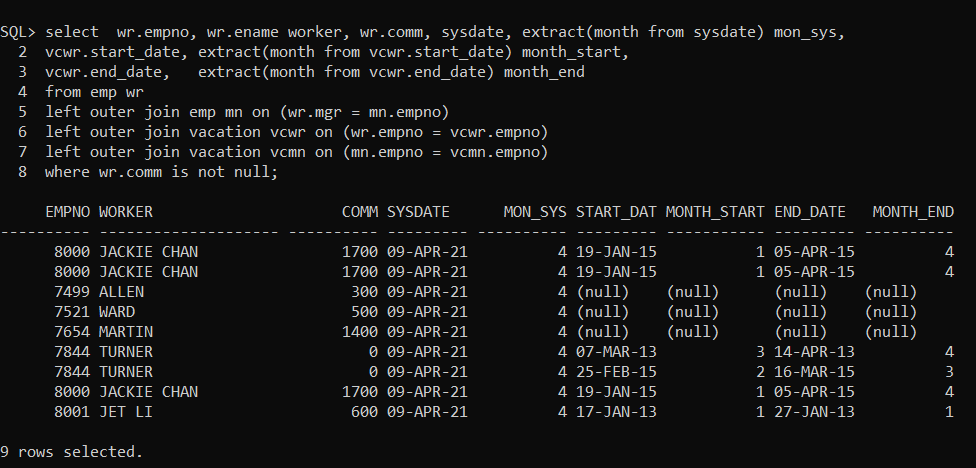
**from emp wr**

**left outer join emp mn on (wr.mgr = mn.empno)**

**left outer join vacation vcwr on (wr.empno = vcwr.empno)**

**left outer join vacation vcmn on (mn.empno = vcmn.empno)**

**where wr.comm is not null;**



7. Видаліть таблицю Vocation(vacation**)**.

**drop table vacation;**

