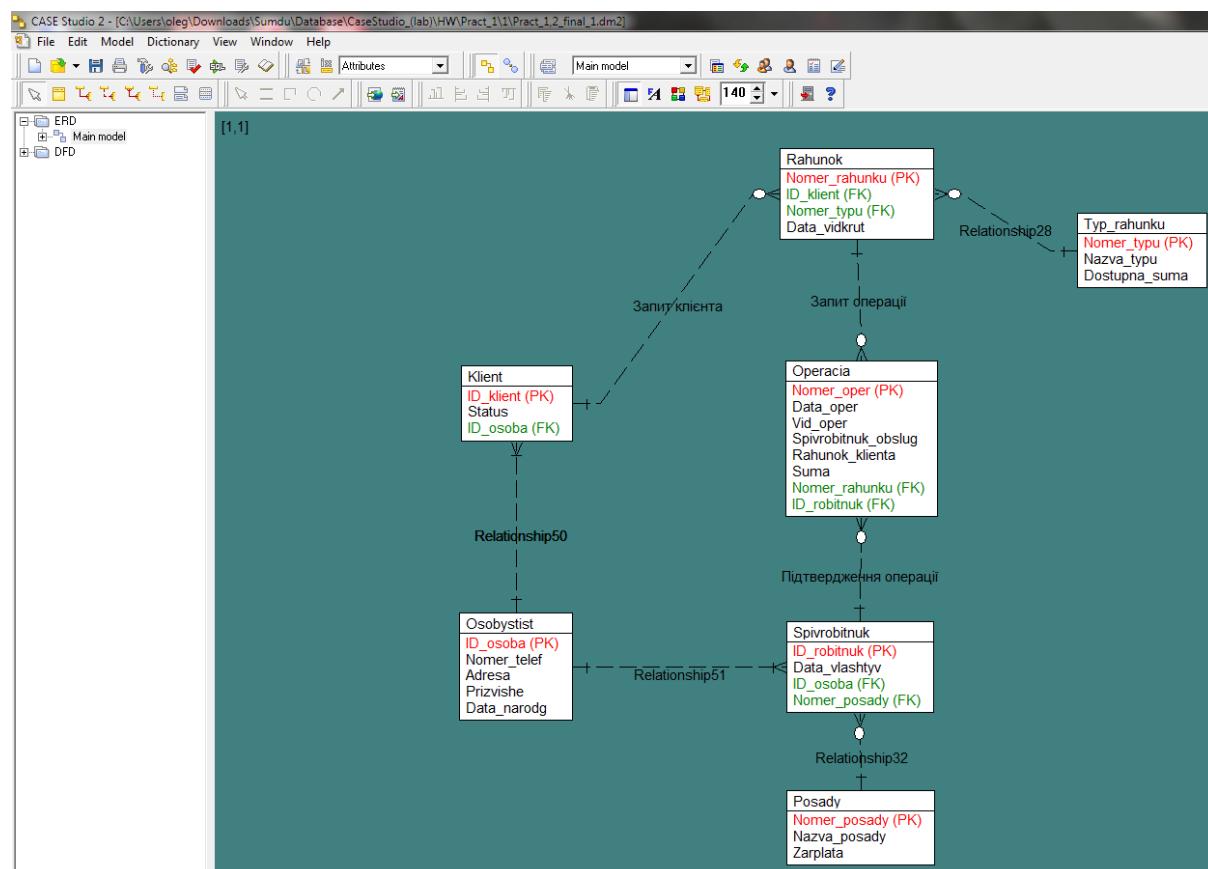


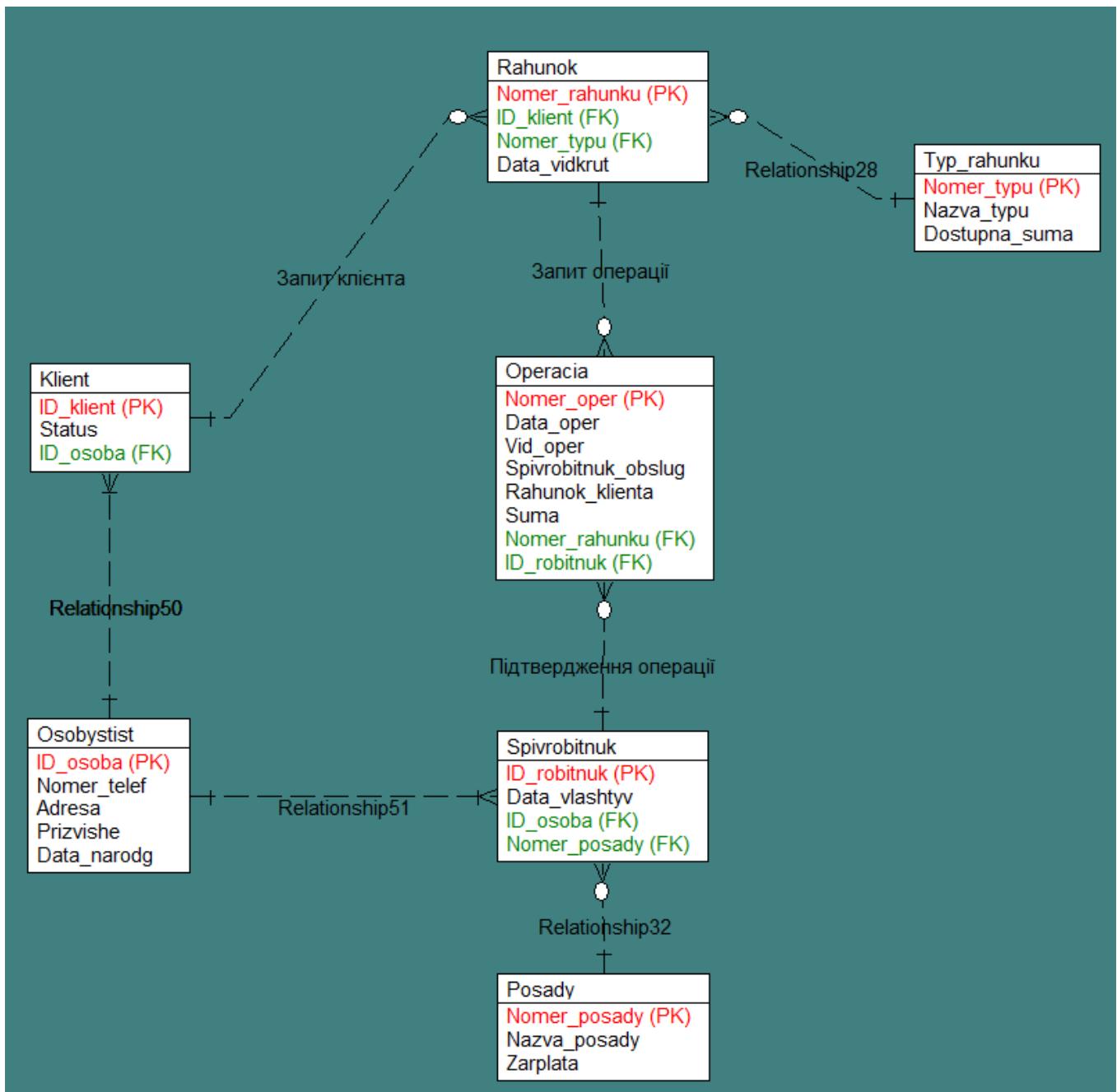
## Лабораторна робота 1

## Завдання 1

Керівник невеликого регіонального банку попросив вас спроектувати і реалізувати базу даних:

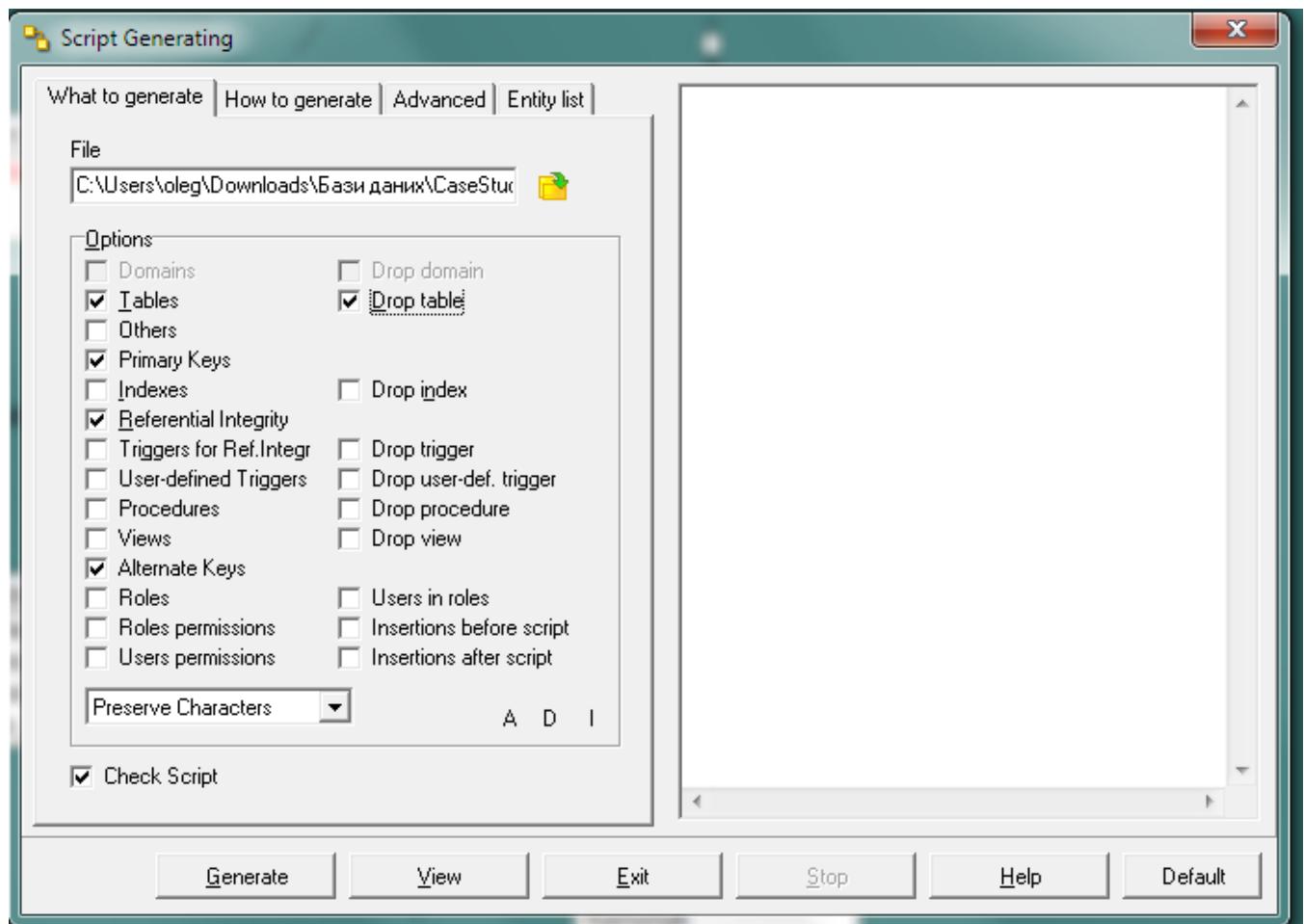
- Клієнти компанії відкривають рахунки, розміщують свої кошти на них, знімають кошти, закривають рахунки тощо;
- Рахунки можуть бути: розрахункові, депозитні, кредитні тощо;
- Клієнти можуть мати довільне число рахунків для проведення фінансових операцій;
- Кожна заявка клієнта на проведення банківських операцій документується і обробляється співробітником банка. Зберігається наступна інформація: хто із співробітників виконав операцію, за яким рахунком, коли здійснена банківська операція, сума операції, клієнт - власник рахунку (на «5» - як ми при цьому гарантуємо що вказана людина дійсно власник рахунку);
- Штат банку складається з N працівників. Штатний розклад банку передбачає посади: керівника, заступника керівника, головного бухгалтера, бухгалтера, операціоніста, контролера, касира, юриста. Співробітник може займати тільки одну посаду;
- Співробітник може бути клієнтом банку (на «5»);
- Побудуйте ER-діаграму ІС, яка супроводжує роботу банку.

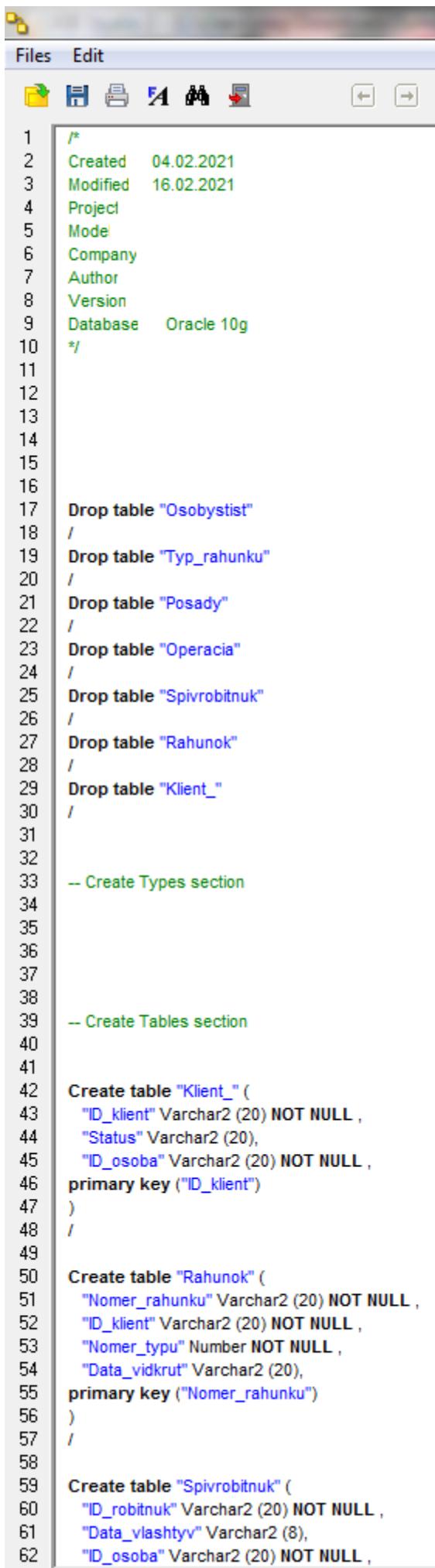




## Завдання 2

- Створіть SQL скрипт генерації таблиць по побудованої ER- діаграмі





The screenshot shows a software application window with a menu bar (Files, Edit) and a toolbar with icons for file operations. The main area is a code editor displaying an Oracle SQL script. The script includes comments at the top, a series of DROP TABLE statements, and three CREATE TABLE statements. The code is color-coded for syntax.

```
/*
Created 04.02.2021
Modified 16.02.2021
Project
Mode
Company
Author
Version
Database Oracle 10g
*/
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 Drop table "Osobystist"
18 /
19 Drop table "Typ_rahunku"
20 /
21 Drop table "Posady"
22 /
23 Drop table "Operacia"
24 /
25 Drop table "Spivrobitnuk"
26 /
27 Drop table "Rahunok"
28 /
29 Drop table "Klient_"
30 /
31
32
33 -- Create Types section
34
35
36
37
38
39 -- Create Tables section
40
41
42 Create table "Klient_" (
43     "ID_klient" Varchar2 (20) NOT NULL ,
44     "Status" Varchar2 (20),
45     "ID_osoba" Varchar2 (20) NOT NULL ,
46     primary key ("ID_klient")
47 )
48 /
49
50 Create table "Rahunok" (
51     "Nomer_rahunku" Varchar2 (20) NOT NULL ,
52     "ID_klient" Varchar2 (20) NOT NULL ,
53     "Nomer_typy" Number NOT NULL ,
54     "Data_vidkrut" Varchar2 (20),
55     primary key ("Nomer_rahunku")
56 )
57 /
58
59 Create table "Spivrobitnuk" (
60     "ID_robitnuk" Varchar2 (20) NOT NULL ,
61     "Data_vlashtyv" Varchar2 (8),
62     "ID_osoba" Varchar2 (20) NOT NULL ,
```

```
--  
59 Create table "Spivrobitnuk" (  
60     "ID_robitnuk" Varchar2 (20) NOT NULL ,  
61     "Data_vlashtyv" Varchar2 (8),  
62     "ID_osoba" Varchar2 (20) NOT NULL ,  
63     "Nomer_posady" Varchar2 (20) NOT NULL ,  
64 primary key ("ID_robitnuk")  
65 )  
66 /  
67  
68 Create table "Operacia" (  
69     "Nomer_oper" Number NOT NULL ,  
70     "Data_oper" Varchar2 (8),  
71     "Vid_oper" Varchar2 (20),  
72     "Spivrobitnuk_obslug" Varchar2 (20),  
73     "Rahunok_klienta" Varchar2 (20),  
74     "Suma" Number,  
75     "Nomer_rahunku" Varchar2 (20) NOT NULL ,  
76     "ID_robitnuk" Varchar2 (20) NOT NULL ,  
77 primary key ("Nomer_oper")  
78 )  
79 /  
80  
81 Create table "Posady" (  
82     "Nomer_posady" Varchar2 (20) NOT NULL ,  
83     "Nazva_posady" Varchar2 (20),  
84     "Zarplata" Number,  
85 primary key ("Nomer_posady")  
86 )  
87 /  
88  
89 Create table "Typ_rahunku" (  
90     "Nomer_typu" Number NOT NULL ,  
91     "Nazva_typu" Varchar2 (20),  
92     "Dostupna_suma" Number,  
93 primary key ("Nomer_typu")  
94 )  
95 /  
96  
97 Create table "Osobystist" (  
98     "ID_osoba" Varchar2 (20) NOT NULL ,  
99     "Nomer_telef" Number,  
100    "Adresa" Varchar2 (20),  
101    "Prizvishe" Varchar2 (20),  
102    "Data_narodg" Varchar2 (20),  
103 primary key ("ID_osoba")  
104 )  
105 /  
106  
107  
108 -- Create Alternate keys section  
109  
110  
111 -- Create section for PKs, AKs and Unique constraints, which have to be generated after Indexes  
112  
113
```

The screenshot shows a software application window with a menu bar at the top labeled "Files Edit". Below the menu is a toolbar with icons for file operations like Open, Save, Print, and a magnifying glass. A dropdown menu is open next to the "SQL" button. The main area is a code editor containing SQL statements. The code is color-coded: green for comments, blue for keywords, and black for identifiers. The code is organized into sections by line numbers and section headers.

```
113  
114  
115 -- Create Foreign keys section  
116  
117 Alter table "Rahunok" add foreign key ("ID_klient") references "Klient_" ("ID_klient")  
118 /  
119  
120 Alter table "Operacia" add foreign key ("Nomer_rahunku") references "Rahunok" ("Nomer_rahunku")  
121 /  
122  
123 Alter table "Operacia" add foreign key ("ID_robitnuk") references "Spivrobitnuk" ("ID_robitnuk")  
124 /  
125  
126 Alter table "Spivrobitnuk" add foreign key ("Nomer_posady") references "Posady" ("Nomer_posady")  
127 /  
128  
129 Alter table "Rahunok" add foreign key ("Nomer_typy") references "Typ_rahunku" ("Nomer_typy")  
130 /  
131  
132 Alter table "Klient_" add foreign key ("ID_osoba") references "Osobystist" ("ID_osoba")  
133 /  
134  
135 Alter table "Spivrobitnuk" add foreign key ("ID_osoba") references "Osobystist" ("ID_osoba")  
136 /  
137  
138  
139 -- Create Object Tables section  
140  
141  
142  
143 -- Create XMLType Tables section  
144  
145  
146  
147 -- Create Functions section  
148  
149  
150  
151 -- Create Sequences section  
152  
153  
154  
155  
156 -- Create Packages section  
157  
158  
159  
160  
161  
162 -- Create Synonyms section  
163  
164  
165  
166 -- Create Table comments section  
167  
168  
169 -- Create Attribute comments section  
170
```

### Завдання 3

- Зіставте ER-діаграму, побудовану в завданні 1 і отриманий SQL-код.
- \* Як в коді записуються первинні і зовнішні ключі?
- \* Як в коді записуються складені первинні і зовнішні ключі?

Первинний ключ в сутності Тип рахунку позначається в ER-діаграмі як *Nomer\_typu PK*, а в отриманому SQL-коді як *primary key ("Nomer\_typu")* :

```
Create table "Typ_rahunku" (
    "Nomer_typu" Number NOT NULL ,
    "Nazva_typu" Varchar2 (20),
    "Dostupna_suma" Number,
    primary key ("Nomer_typu")
)
```

В сутності Рахунок первинний ключ позначається як *"Nomer\_rahunku"(PK)*, зовнішні ключі - як *"ID\_klient"(FK)*, *"Nomer\_typu"(FK)*.

В SQL-коді:

```
Create table "Rahunok" (
    "Nomer_rahunku" Varchar2 (20) NOT NULL ,
    "ID_klient" Varchar2 (20) NOT NULL ,
    "Nomer_typu" Number NOT NULL ,
    "Data_vidkrut" Varchar2 (20),
    primary key ("Nomer_rahunku")
)
```

/

```
Alter table "Rahunok" add foreign key ("ID_klient") references "Klient_" ("ID_klient")
```

```
/
```

```
Alter table "Rahunok" add foreign key ("Nomer_typu") references "Typ_rahunku" ("Nomer_typu")
```

Маємо *"Nomer\_rahunku"* – первинний та *"ID\_klient"*, *"Nomer\_typu"* – вторинні ключі відповідно.

## Лабораторна робота 2

## Завдання 2. Прості запити

У лабораторній роботі використовуються таблиці :

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	SMITH	CLERK	7566	03-DEC-81	3000		20
7902	FORD	ANALYST	7566	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7788	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



```
C:\WINDOWS\system32\cmd.exe
```

```
SQL> SELECT * FROM emp:
```

```
2
```

```
SQL> SELECT * FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7788	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

18 rows selected.

Таблиця emp.

1. Виведемо всі записи з таблиці DEPT:

**SELECT \* FROM Dept;**

```
Выбрать C:\WINDOWS\system32\cmd.exe
SQL> SELECT * FROM Dept;

    DEPTNO  DNAME          LOC
-----  -----
        10  ACCOUNTING    NEW_YORK
        20  RESEARCH      DALLAS
        30  SALES         CHICAGO
        40  OPERATIONS   BOSTON
        50  RESEARCH2    HONKONG
        60  SALES2       HONKONG
       100  SALES3     NEW_YORK

7 rows selected.
```

2. Переглянемо структуру таблиці DEPT:

**DESCRIBE dept;**

```
C:\WINDOWS\system32\cmd.exe
SQL> DESCRIBE dept;
Name          Null?    Type
-----  -----
DEPTNO        NOT NULL NUMBER(3)
DNAME         NOT NULL VARCHAR2(14)
LOC           VARCHAR2(13)
```

3. Виведемо всі назви відділів з таблиці Dept

**SELECT dname FROM Dept;**

```
Выбрать C:\WINDOWS\system32\cmd.exe
SQL> SELECT dname FROM Dept;

DNAME
-----
ACCOUNTING
OPERATIONS
RESEARCH
RESEARCH2
SALES
SALES2
SALES3

7 rows selected.

SQL> -
```

4. Виведемо всі назви відділів та їх розташування з таблиці Dept:

SELECT dname, loc FROM Dept;

```
Выбрать C:\WINDOWS\system32\cmd.exe
SQL> SELECT dname, loc FROM Dept;

DNAME          LOC
-----
ACCOUNTING    NEW_YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON
RESEARCH2      HONKONG
SALES2         HONKONG
SALES3         NEW_YORK

7 rows selected.
```

5. Запит завдання 1 вивів стовпці в послідовності deptno, dname, loc. Скорегуйте запит так, щоб він виводив стовпці в порядку deptno, loc, dname.

SELECT deptno, loc, dname FROM Dept;

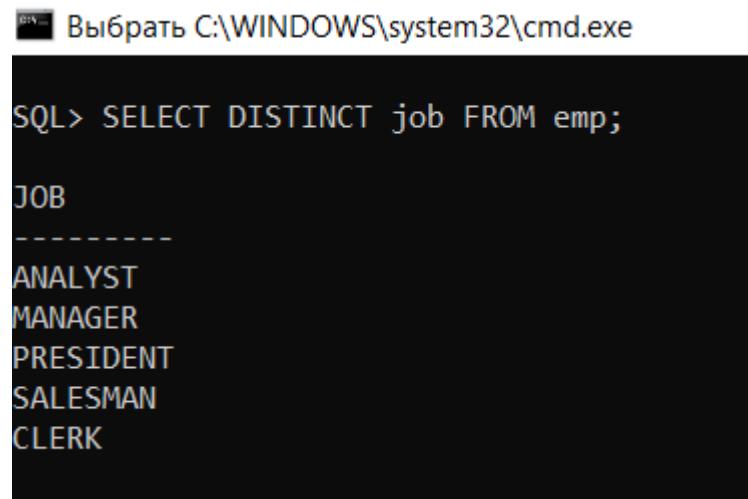
```
SQL> SELECT deptno, loc, dname FROM Dept;

DEPTNO LOC          DNAME
-----
10   NEW_YORK      ACCOUNTING
20   DALLAS        RESEARCH
30   CHICAGO       SALES
40   BOSTON        OPERATIONS
50   HONKONG       RESEARCH2
60   HONKONG       SALES2
100  NEW_YORK      SALES3

7 rows selected.
```

6. Складіть запит, який поверне всі можливі посади співробітників *без повторень*.

SELECT DISTINCT job FROM emp;



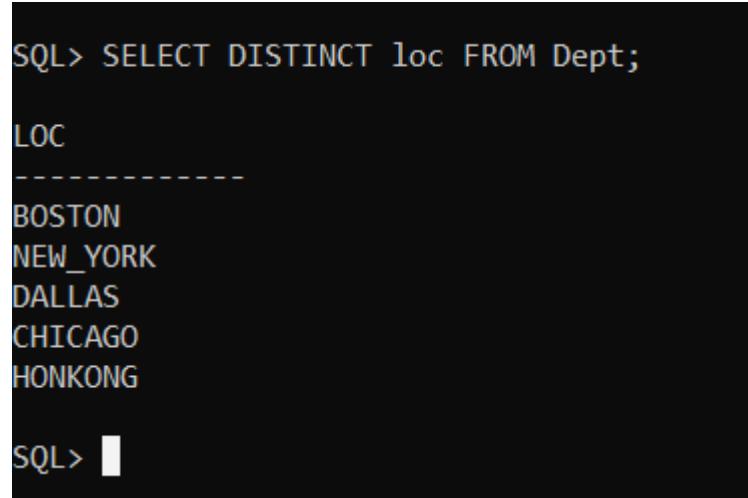
```
Выбрать C:\WINDOWS\system32\cmd.exe

SQL> SELECT DISTINCT job FROM emp;

JOB
-----
ANALYST
MANAGER
PRESIDENT
SALESMAN
CLERK
```

7. Складіть запит, який поверне всі можливі міста розміщення відділів *без повторень*.

SELECT DISTINCT loc FROM Dept;



```
SQL> SELECT DISTINCT loc FROM Dept;

LOC
-----
BOSTON
NEW_YORK
DALLAS
CHICAGO
HONKONG

SQL> █
```

## Завдання 4. Створення таблиць

### Створення таблиць з обмеженнями

- Візьміть за основу код SQL для створення таблиці *STUDENT* і створіть таблицю *STUDENT1*:
- Пара значень (Прізвище, Курс) є унікальною;
- Стипендія не може бути менше ніж 5000;
- Поле *Student\_id* не може бути порожнім;
- За замовчуванням *UNIV\_ID* – 0;
- Переглянте структуру таблиці *STUDENT1*;
- Видаліть таблицю *STUDENT1*.

```
SQL> CREATE TABLE STUDENT1(
 2 STUDENT_ID INTEGER NOT NULL,
 3 SURNAME CHAR (25),
 4 NAME CHAR (10),
 5 STIPEND INTEGER CHECK (STIPEND >= 5000),
 6 KURS INTEGER,
 7 CITY CHAR (15),
 8 BIRTHDAY DATE,
 9 UNIV_ID INTEGER DEFAULT 0,
10 UNIQUE (SURNAME, KURS)
11 );

Table created.

SQL> DESCRIBE STUDENT1;
Name          Null?    Type
-----        -----
STUDENT_ID    NOT NULL NUMBER(38)
SURNAME       CHAR(25)
NAME          CHAR(10)
STIPEND       NUMBER(38)
KURS          NUMBER(38)
CITY          CHAR(15)
BIRTHDAY      DATE
UNIV_ID       NUMBER(38)

SQL> DROP TABLE STUDENT1;
Table dropped.

SQL>
```

```
SQL> CREATE TABLE STUDENT1(
 2 STUDENT_ID INTEGER NOT NULL,
 3 SURNAME CHAR (25),
 4 NAME CHAR (10),
 5 STIPEND INTEGER CHECK (STIPEND >= 5000),
 6 KURS INTEGER,
 7 CITY CHAR (15),
 8 BIRTHDAY DATE,
 9 UNIV_ID INTEGER DEFAULT 0,
10 UNIQUE (SURNAME, KURS)
11 );

Table created.

SQL> DESCRIBE STUDENT1;
Name          Null?    Type
-----        -----
STUDENT_ID    NOT NULL NUMBER(38)
SURNAME       CHAR(25)
NAME          CHAR(10)
STIPEND       NUMBER(38)
KURS          NUMBER(38)
CITY          CHAR(15)
BIRTHDAY      DATE
UNIV_ID       NUMBER(38)

SQL> DROP TABLE STUDENT1;
Table dropped.

SQL>
```

## Лабораторна робота 3

У лабораторній роботі використовуються таблиці :

EMP							
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
	JAMES	CLERK	7698	03-DEC-81	950		30
	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

DEPT		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

## Арифметичні дії

1. Створіть запит, який поверне для всіх співробітників ім'я та розмір його відрахувань у пенсійний фонд. Розмір відрахувань складає 6% від заробітної плати. Назвіть стовпець з відрахуванням deduction.

*Ім'я та зарплата без відрахувань до пенсійного фонду:*

```
SQL> select ename, sal
  2  from emp;

ENAME          SAL
-----
KING           5000
BLAKE          2850
CLARK          1500
JONES          2975
MARTIN         1250
ALLEN          1600
TURNER         1500
JAMES           950
WARD            1250
FORD            3000
SMITH           800
SCOTT           3000
ADAMS           1100
MILLER          1300
JACKIE CHAN    2250
JET LI           2600
BRUCE LEE       2500
DR NO            2500

18 rows selected.
```

*Ім'я та відрахування у пенсійний фонд:*

```
SQL> select ename,
  2  (sal/100)*6
  3  as "deduction"
  4  from emp;

ENAME          deduction
-----
KING             300
BLAKE            171
CLARK             90
JONES            178.5
MARTIN            75
ALLEN             96
TURNER            90
JAMES              57
WARD              75
FORD              180
SMITH              48
SCOTT              180
ADAMS              66
MILLER             78
JACKIE CHAN      135
JET LI              156
BRUCE LEE          150
DR NO              150

18 rows selected.
```

Запити:

```
select ename, sal  
from emp;
```

```
select ename,  
(sal/100)*6  
as "deduction"  
from emp;
```

2. \* Бухгалтерії потрібен звіт, який містить номер співробітника, його зарплату і зарплату збільшенню на 20 % (назвіть цю колонку New Salary), стовпець Increase, у якому відображається на скільки збільшилася зарплата.

```
select empno, sal,  
sal + (sal/100)*20 as "New Salary",  
(sal/100)*20 as "Increase"  
from emp;
```

```
SQL> select empno, sal,  
2  sal + (sal/100)*20 as "New Salary",  
3  (sal/100)*20 as "Increase"  
4  from emp;  
  
EMPNO          SAL New Salary   Increase  
-----  
    7839        5000      6000     1000  
    7698        2850      3420      570  
    7782        1500      1800      300  
    7566        2975      3570      595  
    7654        1250      1500      250  
    7499        1600      1920      320  
    7844        1500      1800      300  
    7900        950       1140      190  
    7521        1250      1500      250  
    7902        3000      3600      600  
    7369        800       960       160  
    7788        3000      3600      600  
    7876        1100      1320      220  
    7934        1300      1560      260  
    8000        2250      2700      450  
    8001        2600      3120      520  
    8002        2500      3000      500  
    8003        2500      3000      500  
  
18 rows selected.
```

СУБД замість калькулятора

3. Обчисліть вираз:  $(5+5)*20-3/2$ .

```
SELECT (5+5)*20-3/2 FROM DUAL;
```

```
SQL> SELECT (5+5)*20-3/2 FROM DUAL;  
  
(5+5)*20-3/2  
-----  
           198.5
```

## Об'єднання рядків

4. Складіть запит, який виведе інформацію про всіх співробітників у форматі:  
Ім'я - Посада (ім'я дефіс посада)

```
select ename || '-' || job  
as "ename - job"  
from emp;
```

```
SQL> select ename || '-' || job  
  2  as "ename - job"  
  3  from emp;  
  
ename - job  
-----  
KING - PRESIDENT  
BLAKE - MANAGER  
CLARK - MANAGER  
JONES - MANAGER  
MARTIN - SALESMAN  
ALLEN - SALESMAN  
TURNER - SALESMAN  
JAMES - CLERK  
WARD - SALESMAN  
FORD - ANALYST  
SMITH - CLERK  
SCOTT - ANALYST  
ADAMS - CLERK  
MILLER - CLERK  
JACKIE CHAN - SALESMAN  
JET LI - SALESMAN  
BRUCE LEE - SALESMAN  
DR NO - ANALYST  
  
18 rows selected.
```

5. \* Складіть запит, який виведе інформацію о всіх підрозділах у форматі:  
Місто -> Назва

```
select loc || '-' || dname  
as "loc -> dname"  
from dept;
```

```
SQL> select loc || '-' || dname  
  2  as "loc -> dname"  
  3  from dept;  
  
loc -> dname  
-----  
NEW_YORK -> ACCOUNTING  
DALLAS -> RESEARCH  
CHICAGO -> SALES  
BOSTON -> OPERATIONS  
HONKONG -> RESEARCH2  
HONKONG -> SALES2  
NEW_YORK -> SALES3  
  
7 rows selected.
```

## Квотування рядків

6. \* Складіть запит, який виведе інформацію про всі підрозділи у форматі:

*department 'Назва' is placed at 'розташування'*

Дайте об'єднаним стовпцям заголовок.

```
select 'Department'|||dname|||' is placed at'|||loc|||"
```

```
as "Department is placed at..."
```

```
from dept;
```

```
SQL> select 'Department '|||dname|||'' is placed at '|||loc|||'  
  2  as "Department is placed at..."  
  3  from dept;  
  
Department is placed at...  
-----  
Department 'ACCOUNTING' is placed at NEW_YORK  
Department 'RESEARCH' is placed at DALLAS  
Department 'SALES' is placed at CHICAGO  
Department 'OPERATIONS' is placed at BOSTON  
Department 'RESEARCH2' is placed at HONKONG  
Department 'SALES2' is placed at HONKONG  
Department 'SALES3' is placed at NEW_YORK  
  
7 rows selected.
```

## Порівняння значень

7. Виведіть імена и заробітну плату співробітників, у яких заробітна плата більше ніж \$3000.

```
select ename, sal
```

```
from emp
```

```
where sal > 3000;
```

```
SQL> select ename, sal  
  2  from emp  
  3  where sal > 3000;  
  
ENAME          SAL  
-----  
KING           5000
```

8. \* Виведіть імена співробітників, у яких комісія (премія) більше ніж 20 % зарплати.

```
select ename
```

```
from emp
```

```
where comm > (sal/100)*20;
```

```

SQL> select ename
  2  from emp
  3  where comm > (sal/100)*20;

ENAME
-----
MARTIN
WARD
JACKIE CHAN
JET LI

```

Підстановка значень

9. Виведіть ім'я співробітника і номер департаменту в якому працює службовець з empno = 7839, де 7839 вводиться користувачем.

```

select ename, deptno
from emp
where empno = &my_number;

```

```

SQL> select ename, deptno
  2  from emp
  3  where empno = &my_number;
Enter value for my_number: 7839
old   3: where empno = &my_number
new   3: where empno = 7839

ENAME              DEPTNO
-----
KING                10

```

Використання AND, OR, NOT (без IN)

10. Виведіть імена і заробітну плату співробітників, для яких заробітна плата знаходиться поза діапазону \$ 1500 і \$ 2500. (*без between*)

```

select ename, sal
from emp
where sal < 1500
or   sal > 2500;

```

```

SQL> select ename, sal
  2  from emp
  3  where sal < 1500
  4 or   sal > 2500;

ENAME          SAL
-----
KING           5000
BLAKE          2850
JONES          2975
MARTIN         1250
JAMES           950
WARD            1250
FORD            3000
SMITH           800
SCOTT           3000
ADAMS           1100
MILLER          1300
JET LI          2600

12 rows selected.

```

11. \* Виведіть імена всіх MANAGER-ів і ANALYST-ів, в яких заробітна плата не дорівнює \$2850 та \$1500.

```
select ename  
from emp  
where (job = 'MANAGER'  
or job = 'ANALYST')  
and (sal != 2850 and sal != 1500);
```

```
SQL> select ename  
  2  from emp  
  3  where (job = 'MANAGER' or job = 'ANALYST')  
  4  and (sal != 2850 and sal != 1500);  
  
ENAME  
-----  
JONES  
FORD  
SCOTT  
DR NO
```

Перевірка:

```
select ename, job, sal  
from emp  
where (job = 'MANAGER'  
or job = 'ANALYST')  
and (sal != 2850  
and sal != 1500);
```

```
SQL> select ename, job, sal  
  2  from emp  
  3  where (job = 'MANAGER' or job = 'ANALYST')  
  4  and (sal != 2850 and sal != 1500);  
  
ENAME          JOB        SAL  
-----  
JONES         MANAGER    2975  
FORD          ANALYST   3000  
SCOTT         ANALYST   3000  
DR NO         ANALYST   2500
```

Використання Between

12. Виведіть імена і з/п співробітників для яких з/п знаходиться поза діапазону \$ 1500 і \$ 2500.

```
select ename, sal  
from emp  
where sal not between 1500 and 2500;
```

```
SQL> select ename, sal  
  2  from emp  
  3  where sal not between 1500 and 2500;  
  
ENAME          SAL  
-----  
KING           5000  
BLAKE          2850  
JONES          2975  
MARTIN         1250  
JAMES           950  
WARD            1250  
FORD            3000  
SMITH            800  
SCOTT           3000  
ADAMS           1100  
MILLER          1300  
JET LI          2600  
  
12 rows selected.
```

## Сортування результатів

13. Виведіть імена та номери департаментів співробітників 10 і 20 департаментів, виконайте сортування імен за алфавітом.

```
select ename, deptno  
from emp  
where deptno in (10, 20)  
order by ename;
```

```
SQL> select ename, deptno  
  2  from emp  
  3  where deptno in (10, 20)  
  4  order by ename;
```

ENAME	DEPTNO
ADAMS	20
CLARK	10
FORD	20
JONES	20
KING	10
MILLER	10
SCOTT	20
SMITH	20

8 rows selected.

## Використання IN

14. Виведіть з таблиці співробітників ім'я, посаду та дату прийому на роботу співробітників SCOTT, ADAMS, ALLEN в порядку їх прийому на роботу.

```
select ename, job, hiredate  
from emp  
where ename in ('SCOTT', 'ADAMS', 'ALLEN')  
order by hiredate;
```

```
SQL> select ename, job, hiredate  
  2  from emp  
  3  where ename in ('SCOTT', 'ADAMS', 'ALLEN')  
  4  order by hiredate;
```

ENAME	JOB	HIREDATE
ALLEN	SALESMAN	20-FEB-11
SCOTT	ANALYST	09-DEC-12
ADAMS	CLERK	12-JAN-13

15. \* Виведіть імена, номери відділів та з/п співробітників для яких з/п знаходиться в діапазоні \$ 1500 і \$ 5000 і працюють у відділах з номерами 10 і 20.

```
select ename, deptno, sal  
from emp  
where sal between 1500 and 5000  
and deptno in (10, 20);
```

```

SQL> select ename, deptno, sal
  2  from emp
  3  where sal between 1500 and 5000
  4  and deptno in (10, 20);

```

ENAME	DEPTNO	SAL
KING	10	5000
CLARK	10	1500
JONES	20	2975
FORD	20	3000
SCOTT	20	3000

### Умови з використанням Null-значення

16. Виведіть імена та посади співробітників у яких немає начальників.

```

select ename, job
from emp
where mgr is null;

```

```

SQL> select ename, job
  2  from emp
  3  where mgr is null;

ENAME          JOB
-----
KING           PRESIDENT

```

17. \* Виведіть посади співробітників, які не мають начальника або премії, за умови, що у виведеному списку посади не повторюються.

```

select distinct job
from emp
where mgr is null
or comm is null;

```

```

SQL> select distinct job
  2  from emp
  3  where mgr is null
  4  or    comm is null;

JOB
-----
ANALYST
MANAGER
PRESIDENT
CLERK
SALESMAN

```

18. \*Виведіть імена і премію співробітників, у яких є премія.

*Перевірка премії тільки на null:*

```
select ename, comm  
from emp  
where comm is not null;
```

```
SQL> select ename, comm  
  2  from emp  
  3  where comm is not null;  
  
ENAME          COMM  
-----  
MARTIN        1400  
ALLEN         300  
TURNER         0  
WARD           500  
JACKIE CHAN   1700  
JET LI          600  
  
6 rows selected.
```

*Перевірка на те, чи є премія > 0:*

```
select ename, comm  
from emp  
where comm is not null  
and comm != 0;
```

```
SQL> select ename, comm  
  2  from emp  
  3  where comm is not null  
  4  and comm != 0;
```

```
ENAME          COMM  
-----  
MARTIN        1400  
ALLEN         300  
WARD           500  
JACKIE CHAN   1700  
JET LI          600
```

Запити з Like

19. Виведіть імена співробітників, у яких третя літера ‘A’.

```
select ename  
from emp  
where ename like '__A%';
```

```
SQL> select ename  
  2  from emp  
  3  where ename like '__A%';  
  
ENAME  
-----  
BLAKE  
CLARK  
ADAMS
```

20. Виведіть імена співробітників, у яких друга літера ‘L’, остання – ‘N’.

```
select ename  
from emp  
where ename like '_L%N';
```

```
SQL> select ename  
  2  from emp  
  3  where ename like '_L%N';  
  
ENAME  
-----  
ALLEN
```

21. \* Знайдіть назви міст, в яких використовується символ «\_».

```
select loc  
from dept  
where loc like '%/_%' escape '/';
```

```
SQL> select loc  
  2  from dept  
  3  where loc like '%/_%' escape '/';  
  
LOC  
-----  
NEW_YORK  
NEW_YORK
```

У лабораторній роботі використовуються таблиці :

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
<b>DEPT</b>		JAMES	CLERK	7698 03-DEC-81	950		30
WARD			SALESMAN	7698 22-FEB-81	1250	500	30
DEPTNO	DNAME	LOC		7566 03-DEC-81	3000		20
10	ACCOUNTING	NEW YORK		7902 17-DEC-80	800		20
20	RESEARCH	DALLAS		7566 09-DEC-82	3000		20
30	SALES	CHICAGO		7788 12-JAN-83	1100		20
40	OPERATIONS	BOSTON		7782 23-JAN-82	1300		10



### Робота з числами

- Відділу кадрів потрібен звіт, який містить ім'я співробітника, поточну зарплату і зарплату збільшенну на 13,33%. Значення останнього стовпчика повинні бути округлені до цілого.

```
select ename, sal, round((sal + (sal/100)*13.33), 0) increased_sal
```

```
from emp;
```

ENAME	SAL	INCREASED_SAL
KING	5000	5667
BLAKE	2850	3230
CLARK	1500	1700
JONES	2975	3372
MARTIN	1250	1417
ALLEN	1600	1813
TURNER	1500	1700
JAMES	950	1077
WARD	1250	1417
FORD	3000	3400
SMITH	800	907
SCOTT	3000	3400
ADAMS	1100	1247
MILLER	1300	1473
JACKIE CHAN	2250	2550
JET LI	2600	2947
BRUCE LEE	2500	2833
DR NO	2500	2833

18 rows selected.

2. Порівняйте результати, які повертають запити:

1. SELECT TRUNC(55.52,2) FROM DUAL;
2. SELECT TRUNC(55.52) FROM DUAL;
3. SELECT TRUNC(55.52,-1) FROM DUAL;
4. SELECT ROUND(55.52,2) FROM DUAL;
5. SELECT ROUND(55.52) FROM DUAL;
6. SELECT ROUND(55.52,-1) FROM DUAL;

```
SQL> SELECT TRUNC(55.52,2) FROM DUAL;
TRUNC(55.52,2)
-----
      55.52

SQL> SELECT TRUNC(55.52) FROM DUAL;
TRUNC(55.52)
-----
      55

SQL> SELECT TRUNC(55.52,-1) FROM DUAL;
TRUNC(55.52,-1)
-----
      50

SQL> SELECT ROUND(55.52,2) FROM DUAL;
ROUND(55.52,2)
-----
      55.52

SQL> SELECT ROUND(55.52) FROM DUAL;
ROUND(55.52)
-----
      56

SQL> SELECT ROUND(55.52,-1) FROM DUAL;
ROUND(55.52,-1)
-----
      60
```

Функція **TRUNC** відсікла значення

1. до сотих;
2. цілих;
3. десятка числа, обнуливши нульовий разряд числа(одиниці).

Тобто, якщо було б число 355.52 та наступний запит:

**SELECT TRUNC(355.52,-2) FROM DUAL;**

ми б отримали 300 (обнулились би розряди десятих та одиниць).

Або отримаємо 350 в цьому випадку:

**SELECT TRUNC(355.52,-1) FROM DUAL;**

Функція **ROUND** округлила значення до виду:

4. сотих;
5. цілого числа;
6. округленого десятка числа, обнуливши нульовий разряд числа(одиниці).

Тобто, якщо було б число 355.52 та наступний запит:

**SELECT ROUND(355.52,-2) FROM DUAL;**

ми б отримали 400 (обнулились би розряди десятих та одиниць).

Або отримаємо 360 в цьому випадку:

**SELECT ROUND(355.52,-1) FROM DUAL;**

## Робота з рядками (запити без Like)

3. Виберіть ім'я та довжину імені всіх співробітників, ім'я яких починається з «A», «C» або «J». Дайте стовпцю з довжиною імені псевдонімом.

```
select      ename,      length(ename)
length_ename
from emp
where substr(ename, 1, 1) = 'A'
or   substr(ename, 1, 1) = 'C'
or   substr(ename, 1, 1) = 'J';
```

```
SQL> select ename, length(ename) length_ename
  2  from emp
  3  where substr(ename, 1, 1) = 'A'
  4  or   substr(ename, 1, 1) = 'C'
  5  or   substr(ename, 1, 1) = 'J';
```

ENAME	LENGTH_ENAME
CLARK	5
JONES	5
ALLEN	5
JAMES	5
ADAMS	5
JACKIE CHAN	11
JET LI	6

7 rows selected.

4. Виведіть міста, в назвах яких 4-а буква — «C» (CHICAGO).

```
select loc
from dept
where substr(loc, 4, 1) = 'C';
```

```
SQL> select loc
  2  from dept
  3  where substr(loc, 4, 1) = 'C';
```

LOC
CHICAGO

5. \* Виведіть перші слова в назвах міст, якщо назва міста складається з 2-х слів.  
(Назви зберігаються в стовбці loc таблиці Dept)

- Наприклад для «New\_York» виведіть «New». «Dallas» - складається з одного слова і не виводиться.

```
select substr(loc, 1, instr(loc,'_') - 1) first_city
from dept
where instr(loc, '_') != 0;
```

```
SQL> select substr(loc, 1, instr(loc,'_') - 1) first_city
  2  from dept
  3  where instr(loc, '_') != 0;
```

FIRST_CITY
NEW
NEW

6. \* Створіть запит, який поверне колонку NAME\_AND\_SALARIES. Колонка має містити ім'я працівника і кілька зірочок (\*) по одній зірочці на кожні \$ 250 зарплати.

➤ Наприклад «SMITH \*\*\*» - для працівника SMITH із зарплатою \$800

```
select rpad(ename, trunc(sal/250) + length(ename), '*') NAME_AND_SALARIES  
from emp;
```

```
SQL> select rpad(ename, trunc(sal/250) + length(ename), '*') NAME_AND_SALARIES  
2  from emp;  
  
NAME_AND_SALARIES  
-----  
KING*****  
BLAKE*****  
CLARK*****  
JONES*****  
MARTIN*****  
ALLEN*****  
TURNER*****  
JAMES***  
WARD*****  
FORD*****  
SMITH***  
SCOTT*****  
ADAMS***  
MILLER*****  
JACKIE CHAN*****  
JET LI*****  
BRUCE LEE*****  
DR NO*****  
  
18 rows selected.
```

## Робота з рядками

7. Створіть звіт, який для кожного співробітника поверне рядок:

<ename> earns \$ <salary> monthly but wants \$ <3 \* salary>

Наприклад :

«SMITH earns \$ 800.00 monthly but wants \$ 2400.00»

Зарплату виводити з точністю до 2-х знаків після коми (тобто у виводі повинно бути 2400.00, а не 2400).

Для об'єднання рядків скористайтеся функцією.

```
select concat (ename || ' earns', to_char(sal, '$99,999.99'))  
|| concat (' monthly but wants', to_char(3*sal, '$99,999.99'))  
as "ENAME earns..."  
from emp;
```

```
SQL> select concat (ename || ' earns', to_char(sal, '$99,999.99'))  
  2 || concat (' monthly but wants', to_char(3*sal, '$99,999.99'))  
  3 as "ENAME earns..."  
  4 from emp;  
  
ENAME earns...  
-----  
KING earns $5,000.00 monthly but wants $15,000.00  
BLAKE earns $2,850.00 monthly but wants $8,550.00  
CLARK earns $1,500.00 monthly but wants $4,500.00  
JONES earns $2,975.00 monthly but wants $8,925.00  
MARTIN earns $1,250.00 monthly but wants $3,750.00  
ALLEN earns $1,600.00 monthly but wants $4,800.00  
TURNER earns $1,500.00 monthly but wants $4,500.00  
JAMES earns $950.00 monthly but wants $2,850.00  
WARD earns $1,250.00 monthly but wants $3,750.00  
FORD earns $3,000.00 monthly but wants $9,000.00  
SMITH earns $800.00 monthly but wants $2,400.00  
SCOTT earns $3,000.00 monthly but wants $9,000.00  
ADAMS earns $1,100.00 monthly but wants $3,300.00  
MILLER earns $1,300.00 monthly but wants $3,900.00  
JACKIE CHAN earns $2,250.00 monthly but wants $6,750.00  
JET LI earns $2,600.00 monthly but wants $7,800.00  
BRUCE LEE earns $2,500.00 monthly but wants $7,500.00  
DR NO earns $2,500.00 monthly but wants $7,500.00  
  
18 rows selected.
```

## Робота з датами

8. Напишіть запит, який поверне поточну дату.

**select SYSDATE from dual;**

```
SQL> select SYSDATE from dual;  
  
SYSDATE  
-----  
03-MAR-21
```

9. \* Сформувати запит, що виводить в порядку спадання кількість місяців, що минули з моменту найму кожного працівника. Кількість місяців округлити до найближчого цілого.

**select round(months\_between(SYSDATE, hiredate)) months  
from emp  
order by months desc;**

```

SQL> select round(months_between(SYSDATE, hiredate)) months
  2  from emp
  3  order by months desc;

MONTHS
-----
 123
 120
 120
 120
 119
 118
 117
 114
 114
 114
 113
 113
 112
 111
 111
 109
 99
 98

18 rows selected.

```

10. \* Створіть запит, який поверне імена співробітників, роки коли вони були найняті та дні тижня. Дані впорядкувати по роках за спаданням, по іменах за зростанням.

```

select ename,
to_char(hiredate, 'YYYY') year,
to_char(hiredate, 'DAY') weekday
from emp
order by years desc, ename asc;

```

```

SQL> select ename,
  2  to_char(hiredate, 'YYYY') year,
  3  to_char(hiredate, 'DAY') weekday
  4  from emp
  5  order by year desc, ename asc;

```

ENAME	YEAR	WEEKDAY
ADAMS	2013	SATURDAY
MILLER	2012	MONDAY
SCOTT	2012	SUNDAY
ALLEN	2011	SUNDAY
BLAKE	2011	SUNDAY
BRUCE LEE	2011	THURSDAY
CLARK	2011	THURSDAY
DR NO	2011	SUNDAY
FORD	2011	SATURDAY
JACKIE CHAN	2011	WEDNESDAY
JAMES	2011	SATURDAY
JET LI	2011	SUNDAY
JONES	2011	SATURDAY
KING	2011	THURSDAY
MARTIN	2011	WEDNESDAY
TURNER	2011	THURSDAY
WARD	2011	TUESDAY
SMITH	2010	FRIDAY

```
18 rows selected.
```

## Значення Null

11. Сформувати запит, що виводить прізвище службовця і його комісійні. У разі відсутності комісійних в стовпці має бути виведено "No commissions".

```
select ename,
       coalesce(to_char(comm), 'No commissions') comm
  from emp;
```

```
SQL> select ename,
  2  coalesce(to_char(comm), 'No commissions') comm
  3  from emp;

ENAME          COMM
-----          -----
KING           No commissions
BLAKE          No commissions
CLARK          No commissions
JONES          No commissions
MARTIN         1400
ALLEN          300
TURNER         0
JAMES           No commissions
WARD            500
FORD             No commissions
SMITH            No commissions
SCOTT            No commissions
ADAMS            No commissions
MILLER           No commissions
JACKIE CHAN     1700
JET LI           600
BRUCE LEE        No commissions
DR NO            No commissions

18 rows selected.
```

12. Створіть звіт, який містить ім'я співробітника, заробіток (зарплата + премія) та коментар. Поле коментар містить значення «SAL + COM», якщо у співробітника є премія, або «SAL», якщо премія не встановлена.

```
select ename, (sal+nvl(comm, 0)) income,
       nvl2(comm, 'SAL + COM', 'SAL') comments
  from emp;
```

```
SQL> select ename, (sal+nvl(comm, 0)) income,
  2  nvl2(comm, 'SAL + COM', 'SAL') comments
  3  from emp;

ENAME          INCOME COMMENTS
-----          -----
KING           5000 SAL
BLAKE          2850 SAL
CLARK          1500 SAL
JONES          2975 SAL
MARTIN         2650 SAL + COM
ALLEN          1900 SAL + COM
TURNER         1500 SAL + COM
JAMES            950 SAL
WARD            1750 SAL + COM
FORD             3000 SAL
SMITH            800 SAL
SCOTT            3000 SAL
ADAMS            1100 SAL
MILLER           1300 SAL
JACKIE CHAN     3950 SAL + COM
JET LI           3200 SAL + COM
BRUCE LEE        2500 SAL
DR NO            2500 SAL

18 rows selected.
```

13. \* Створіть звіт, який містить ім'я співробітника і його премію. Якщо премія не встановлена поле повинно містити номер керівника (mgr). Якщо у співробітника немає премії і керівника поле повинно містити «No value»

**select ename,**

**coalesce(to\_char(comm), to\_char(mgr), 'No value') comm**

**from emp;**

```
SQL> select ename,
  2 coalesce(to_char(comm), to_char(mgr), 'No value') comm
  3 from emp;

ENAME          COMM
-----
KING           No value
BLAKE          7839
CLARK          7839
JONES          7839
MARTIN         1400
ALLEN          300
TURNER          0
JAMES           7698
WARD            500
FORD             7566
SMITH           7902
SCOTT           7566
ADAMS           7788
MILLER          7782
JACKIE CHAN    1700
JET LI           600
BRUCE LEE       8000
DR NO            7839

18 rows selected.
```

#### Використання CASE і DECODE

14. \* Всім співробітникам вирішили надати рівень допуску, згідно з посадою:

- PRESIDENT – A
- MANAGER – B
- SALESMAN – C
- Для всіх інших – 0.

Використовуючи CASE складіть звіт, який містить ім'я співробітника і його рівень допуску.

```

select ename,
case job
when 'PRESIDENT' then 'A'
when 'MANAGER' then 'B'
when 'SALESMAN' then 'C'
else '0' end l
from emp;

```

```

SQL> select ename,
2 case job
3 when 'PRESIDENT' then 'A'
4 when 'MANAGER' then 'B'
5 when 'SALESMAN' then 'C'
6 else '0' end l
7 from emp;

ENAME          L
-----
KING           A
BLAKE          B
CLARK          B
JONES          B
MARTIN         C
ALLEN          C
TURNER         C
JAMES           0
WARD            C
FORD             0
SMITH            0
SCOTT            0
ADAMS            0
MILLER           0
JACKIE CHAN      C
JET LI           C
BRUCE LEE        C
DR NO            0

18 rows selected.

```

15. \* Реалізуйте попереднє завдання за допомогою DECODE.

```

select ename,
decode (job,
'PRESIDENT', 'A',
'MANAGER', 'B',
'SALESMAN', 'C',
'0') l
from emp;

```

```

SQL> select ename,
2 decode (job,
3 'PRESIDENT', 'A',
4 'MANAGER', 'B',
5 'SALESMAN', 'C',
6 '0') l
7 from emp;

ENAME          L
-----
KING           A
BLAKE          B
CLARK          B
JONES          B
MARTIN         C
ALLEN          C
TURNER         C
JAMES           0
WARD            C
FORD             0
SMITH            0
SCOTT            0
ADAMS            0
MILLER           0
JACKIE CHAN      C
JET LI           C
BRUCE LEE        C
DR NO            0

18 rows selected.

```

## 16\* Завдання

Однією командою SELECT вивести інформацію про підрозділи компанії, що мають в назві більше одного слова.

Звіт повинен містити:

- a. ідентифікатор підрозділу;
- b. друге слово в назві підрозділу.

Слова в назві підрозділу розділяються пробілами.

Дані необхідно впорядкувати по зростанню ідентифікаторів підрозділів компанії.

У назві підрозділу може бути **більше 2-х слів**, потрібно тільки 2-е.

Для виконання завдання можестати в нагоді функція INSTR.

*Для виконання цього завдання використаємо наступний код:*

```
drop table Dept_name;
CREATE TABLE Dept_name( -- описание подразделения
    deptno NUMBER(3) NOT NULL, -- № подразделения
    dname VARCHAR(20) NOT NULL, -- название подразделения
    loc VARCHAR(13), -- местоположение подразделения
    CONSTRAINT dept_pk_ PRIMARY KEY (deptno),
    CONSTRAINT dept_naem_uk_ UNIQUE (dname)
);
INSERT INTO Dept_name VALUES (10,'ACCOUNTING','NEW_YORK');
INSERT INTO Dept_name VALUES (20,'RESEARCH NEW','DALLAS');
INSERT INTO Dept_name VALUES (30,'SALES','CHICAGO');
INSERT INTO Dept_name VALUES (40,'OPERATIONS','BOSTON');
INSERT INTO Dept_name VALUES (50,'RESEARCH2 plus','HONKONG');
INSERT INTO Dept_name VALUES (60,'SALES 2','HONKONG');
INSERT INTO Dept_name VALUES (100,'SALES NEW 3 ','NEW_YORK');
INSERT INTO Dept_name VALUES (90,'PROJECT ABC ZZZ','NEW_YORK');
commit;
SELECT * FROM Dept_name;
/
```

```
SQL> CREATE TABLE Dept_name( -- описание подразделения
  2      deptno NUMBER(3) NOT NULL, -- № подразделения
  3      dname VARCHAR(20) NOT NULL, -- название подразделения
  4      loc VARCHAR(13), -- местоположение подразделения
  5      CONSTRAINT dept_pk_ PRIMARY KEY (deptno),
  6      CONSTRAINT dept_naem_uk_ UNIQUE (dname)
  7  );
Table created.

SQL> INSERT INTO Dept_name  VALUES (10,'ACCOUNTING','NEW_YORK');

1 row created.

SQL> INSERT INTO Dept_name VALUES (20,'RESEARCH NEW','DALLAS');

1 row created.

SQL> INSERT INTO Dept_name VALUES (30,'SALES','CHICAGO');

1 row created.

SQL> INSERT INTO Dept_name VALUES (40,'OPERATIONS','BOSTON');

1 row created.

SQL> INSERT INTO Dept_name VALUES (50,'RESEARCH2 plus','HONKONG');

1 row created.

SQL> INSERT INTO Dept_name VALUES (60,'SALES 2','HONKONG');

1 row created.

SQL> INSERT INTO Dept_name VALUES (100,'SALES NEW 3 ','NEW_YORK');

1 row created.

SQL> INSERT INTO Dept_name  VALUES (90,'PROJECT ABC ZZZ','NEW_YORK');

1 row created.

SQL> commit;

Commit complete.

SQL> SELECT * FROM Dept_name;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW_YORK
20	RESEARCH NEW	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	RESEARCH2 plus	HONKONG
60	SALES 2	HONKONG
100	SALES NEW 3	NEW_YORK
90	PROJECT ABC ZZZ	NEW_YORK

Отримали наступне:

```
SQL> SELECT * FROM Dept_name;

DEPTNO DNAME          LOC
----- -----
 10 ACCOUNTING        NEW_YORK
 20 RESEARCH NEW      DALLAS
 30 SALES             CHICAGO
 40 OPERATIONS        BOSTON
 50 RESEARCH2 plus    HONKONG
 60 SALES 2           HONKONG
100 SALES NEW 3       NEW_YORK
 90 PROJECT ABC ZZZ   NEW_YORK

8 rows selected.
```

Тепер для того, щоб вивести інформацію про підрозділи компанії, що мають в назві більше одного слова, використаємо наступний запит:

```
select deptno,
decode(instr(dname, ' ', instr(dname, ' ')+1),
'0', substr(dname, instr(dname,' ')+1, length(dname)),
substr(dname, instr(dname,' ')+1,
instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' '))))
second_dname
from Dept_name
where instr(dname, ' ') != 0
order by deptno;
```

```
SQL> select deptno,
 2 decode (instr(dname, ' ', instr(dname, ' ')+1),
 3 '0', substr(dname, instr(dname,' ')+1, length(dname)),
 4 substr(dname, instr(dname,' ')+1,
 5 instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' ')))
 6 second_dname
 7 from Dept_name
 8 where instr(dname, ' ') != 0
 9 order by deptno;

DEPTNO SECOND_DNAME
----- -----
 20 NEW
 50 plus
 60 2
 90 ABC
100 NEW
```

Перевірка результату:

```
select deptno, dname,
decode(instr(dname, ' ', instr(dname, ' ')+1),
'0', substr(dname, instr(dname,' ')+1, length(dname)),
substr(dname, instr(dname,' ')+1,
instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' '))))
second_dname
from Dept_name
where instr(dname, ' ') != 0
order by deptno;
```

```
SQL> select deptno, dname,
2 decode (instr(dname, ' ', instr(dname, ' ')+1),
3 '0', substr(dname, instr(dname,' ')+1, length(dname)),
4 substr(dname, instr(dname,' ')+1,
5 instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' '))))
6 second_dname
7 from Dept_name
8 where instr(dname, ' ') != 0
9 order by deptno;
```

DEPTNO	DNAME	SECOND_DNAME
20	RESEARCH NEW	NEW
50	RESEARCH2 plus	plus
60	SALES 2	2
90	PROJECT ABC ZZZ	ABC
100	SALES NEW 3	NEW

У лабораторній роботі використовуються таблиці :

EMP							
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
DEPT		JAMES	CLERK	7698 03-DEC-81	950		30
		WARD	SALESMAN	7698 22-FEB-81	1250	500	30
DEPTNO	DNAME	LOC		7566 03-DEC-81	3000		20
				7902 17-DEC-80	800		20
10	ACCOUNTING	NEW YORK		7566 09-DEC-82	3000		20
20	RESEARCH	DALLAS		7788 12-JAN-83	1100		20
30	SALES	CHICAGO		7782 23-JAN-82	1300		10
40	OPERATIONS	BOSTON					

### Підзапит в умові WHERE

- Створіть запит, що виводить імена всіх підлеглих KING.

**select ename**

**from emp**

**where mgr =**

**(select empno**

**from emp**

**where ename = 'KING');**

```
SQL> select ename
  2  from emp
  3  where mgr =
  4      (select empno
  5       from emp
  6       where ename = 'KING');

ENAME
-----
BLAKE
CLARK
JONES
JACKIE CHAN
DR NO
```

2. Створіть запит, що виводить список співробітників компанії, які працюють в одному відділі зі SMITH.

**select ename, deptno**

**from emp**

**where deptno =**

**(select deptno**

**from emp**

**where ename = 'SMITH');**

```
SQL> select ename, deptno
  2  from emp
  3  where deptno =
  4          (select deptno
  5           from emp
  6           where ename = 'SMITH');
```

ENAME	DEPTNO
JONES	20
FORD	20
SMITH	20
SCOTT	20
ADAMS	20

3. \* Сформулюйте запит, що виводить імена співробітників, найдятих після SMITH в його відділі.

**select ename**

**from emp**

**where hiredate >**

**(select hiredate**

**from emp**

**where ename = 'SMITH')**

**and deptno =**

**(select deptno**

**from emp**

**where ename = 'SMITH');**

```
SQL> select ename
  2  from emp
  3  where hiredate >
  4          (select hiredate
  5           from emp
  6           where ename = 'SMITH')
  7  and deptno =
  8          (select deptno
  9           from emp
 10           where ename = 'SMITH');
```

ENAME
JONES
FORD
SCOTT
ADAMS

## Використання In, Any, All

4. Створіть запит, що виводить імена співробітників найнятих після службовців відділу №30.

```
select ename, hiredate  
from emp  
where hiredate > all  
      (select hiredate  
       from emp  
      where deptno = 30);
```

```
SQL> select ename, hiredate  
  2  from emp  
  3  where hiredate > all  
  4          (select hiredate  
  5            from emp  
  6            where deptno = 30);  
  
ENAME              HIREDATE  
-----  
MILLER           23-JAN-12  
SCOTT            09-DEC-12  
ADAMS            12-JAN-13
```

5. Сформуйте запит, що виводить прізвище, номер підрозділу і зарплату працівників які працюють у відділах, де деяким співробітником виплачують премію.

```
select ename, deptno, sal  
  2  from emp  
  3  where deptno = any  
  4          (select deptno  
  5            from emp  
  6            where comm is not null);  
  
select ename, deptno, sal  
  from emp  
 where deptno = any  
      (select deptno  
       from emp  
      where comm is not null);  
  
ENAME          DEPTNO    SAL  
-----  
BLAKE          30        2850  
MARTIN         30        1250  
ALLEN          30        1600  
TURNER         30        1500  
JAMES           30        950  
WARD            30        1250  
JACKIE CHAN    60        2250  
JET LI          60        2600  
BRUCE LEE       60        2500  
  
9 rows selected.
```

6. \* Створіть запит, що виводить прізвища керівників, у яких всі підлеглі отримують більше \$ 2500.

```
select ename
from emp
where empno in
      (select mgr
       from emp
       where sal > 2500)
and empno not in
      (select mgr
       from emp
       where sal <= 2500);
```

```
SQL> select ename
  2  from emp
  3  where empno in
  4          (select mgr
  5           from emp
  6           where sal > 2500)
  7  and empno not in
  8          (select mgr
  9           from emp
 10          where sal <= 2500);

ENAME
-----
JONES
```

7. Сформуйте запит, що виводить інформацію про працівників, зарплата яких більше зарплати будь-якого з клерків.

*Візьмемо всіх тих працівників, які не є клерками(секретарями) та мають зарплату вищу за всі зарплати клерків:*

```
select *
from emp
where sal > all
      (select sal
       from emp
       where job = 'CLERK');
```

```
SQL> select *
  2  from emp
  3  where sal > all
  4          (select sal
  5           from emp
  6           where job = 'CLERK');

   EMPNO ENAME      JOB         MGR HIREDATE      SAL     COMM  DEPTNO
----- -----  -----
    7844 TURNER    SALESMAN    7698 08-SEP-11    1500      0      30
    7782 CLARK     MANAGER    7839 09-JUN-11    1500 (null)    10
    7499 ALLEN     SALESMAN    7698 20-FEB-11    1600     300      30
    8000 JACKIE CHAN SALESMAN    7839 28-SEP-11    2250    1700      60
    8002 BRUCE LEE SALESMAN    8000 08-SEP-11    2500 (null)      60
    8003 DR NO     ANALYST    7839 11-SEP-11    2500 (null) (null)
    8001 JET LI     SALESMAN    8000 20-FEB-11    2600     600      60
    7698 BLAKE     MANAGER    7839 01-MAY-11    2850 (null)      30
    7566 JONES     MANAGER    7839 02-APR-11    2975 (null)      20
    7788 SCOTT     ANALYST    7566 09-DEC-12    3000 (null)      20
    7902 FORD      ANALYST    7566 03-DEC-11    3000 (null)      20
    7839 KING      PRESIDENT  (null) 17-NOV-11    5000 (null)      10

12 rows selected.
```

Тепер виберемо тих, хто мають зарплату вищу за якогось з клерків(включаючи до результату і клерків):

```
select *
from emp
where sal > any
      (select sal
       from emp
      where job = 'CLERK');
```

```
SQL> select *
  2  from emp
  3  where sal > any
  4      (select sal
  5      from emp
  6      where job = 'CLERK');

EMPNO ENAME          JOB            MGR HIREDATE        SAL     COMM   DEPTNO
----- -----
 7839 KING            PRESIDENT      (null) 17-NOV-11    5000  (null)    10
 7788 SCOTT           ANALYST       7566  09-DEC-12   3000  (null)    20
 7902 FORD            ANALYST       7566  03-DEC-11   3000  (null)    20
 7566 JONES           MANAGER      7839  02-APR-11   2975  (null)    20
 7698 BLAKE           MANAGER      7839  01-MAY-11   2850  (null)    30
 8001 JET LI          SALESMAN     8000  20-FEB-11   2600    600     60
 8002 BRUCE LEE       SALESMAN     8000  08-SEP-11   2500  (null)    60
 8003 DR NO           ANALYST      7839  11-SEP-11   2500  (null)  (null)
 8000 JACKIE CHAN    SALESMAN     7839  28-SEP-11   2250    1700    60
 7499 ALLEN           SALESMAN     7698  20-FEB-11   1600     300    30
 7782 CLARK           MANAGER      7839  09-JUN-11   1500  (null)    10
 7844 TURNER          SALESMAN     7698  08-SEP-11   1500     0      30
 7934 MILLER          CLERK        7782  23-JAN-12   1300  (null)    10
 7521 WARD            SALESMAN     7698  22-FEB-11   1250     500    30
 7654 MARTIN          SALESMAN     7698  28-SEP-11   1250    1400    30
 7876 ADAMS            CLERK        7788  12-JAN-13   1100  (null)    20
 7900 JAMES            CLERK        7698  03-DEC-11    950  (null)    30

17 rows selected.
```

8. \* Відділу кадрів потрібен звіт, який містить ім'я і зарплату всіх співробітників з Dallas.

```
select ename, sal
from emp
where deptno = all
```

```
(select deptno
```

```
from dept
```

```
where loc = 'DALLAS');
```

```
SQL> select ename, sal
  2  from emp
  3  where deptno = all
  4      (select deptno
  5      from dept
  6      where loc = 'DALLAS');
```

ENAME	SAL
JONES	2975
FORD	3000
SMITH	800
SCOTT	3000
ADAMS	1100

## Підзапит в списку полів

9. Створіть запит, що виводить номер і прізвище працівника, номер і прізвище його менеджера.

```
select emp.empno, emp.ename worker, mng.empno, mng.ename manager  
from emp, emp mng  
where mng.job = 'MANAGER'  
and emp.mgr = mng.empno;
```

```
SQL> select emp.empno, emp.ename worker, mng.empno, mng.ename manager  
  2  from emp, emp mng  
  3  where mng.job = 'MANAGER'  
  4  and emp.mgr = mng.empno;
```

EMPNO	WORKER	EMPNO	MANAGER
7902	FORD	7566	JONES
7788	SCOTT	7566	JONES
7499	ALLEN	7698	BLAKE
7521	WARD	7698	BLAKE
7654	MARTIN	7698	BLAKE
7844	TURNER	7698	BLAKE
7900	JAMES	7698	BLAKE
7934	MILLER	7782	CLARK

```
8 rows selected.
```

10. \* Створіть запит, що виводить прізвище працівника і місто в якому він працює.

```
select ename,  
       (select loc  
        from dept  
        where emp.deptno = dept.deptno) locs  
  from emp;
```

```

SQL> select ename,
      2          (select loc
      3           from dept
      4           where emp.deptno = dept.deptno) locs
      5    from emp;

ENAME          LOCS
-----
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
DR NO           (null)

18 rows selected.

```

## Підзапит у списку FROM

11. Створіть запит, який виведе імена, номер відділу і зарплату співробітників з відділу №10.

```

select ename, deptno, sal
from (select * from emp where deptno = 10);

```

```

SQL> select ename, deptno, sal
  2  from (select * from emp where deptno = 10);

ENAME          DEPTNO      SAL
-----
KING           10          5000
CLARK          10          1500
MILLER         10          1300

```

12. Створіть запит, який вибере всіх службовців з зарплатою вище \$ 1500 з результатів запиту 11.

```

select ename, deptno, sal
from (select * from emp where
deptno = 10)
where sal > 1500;

```

```

SQL> select ename, deptno, sal
  2  from (select * from emp where deptno = 10)
  3  where sal > 1500;

ENAME          DEPTNO      SAL
-----
KING           10          5000

```

13. \* Створіть представлення my\_view на основі запиту 11.

**create view my\_view**

**as select ename, deptno, sal**

**from (select \* from emp where deptno = 10);**

**select \* from my\_view;**

```
SQL> create view my_view
  2  as select ename, deptno, sal
  3  from (select * from emp where deptno = 10);

View created.

SQL> select * from my_view;

ENAME          DEPTNO      SAL
-----          -----
KING            10        5000
CLARK           10        1500
MILLER          10        1300
```

14. \* Створіть запит, який вибере всіх службовців з зарплатою вище \$ 1500 із представлення my\_view.

**select \* from my\_view**

**where sal > 1500;**

```
SQL> select * from my_view
  2  where sal > 1500;

ENAME          DEPTNO      SAL
-----          -----
KING            10        5000
```

15. \* Видаліть представлення my\_view.

**drop view my\_view;**

```
SQL> drop view my_view;

View dropped.
```

## Лабораторна робота 6

## INSERT

1. Додайте в таблицю My\_emp відомості про нового співробітника:

empno - 665

ename - IVAN

job – SALESMAN

mgr - 7839

hiredate – текущая дата

sal - 9999

comm - NULL

deptno – 20

Переконайтесь, що співробітник доданий.

```
insert into My_emp(empno, ename, job, mgr,
                   hiredate, sal, comm, deptno)
```

```
values      (665, 'IVAN', 'SALESMAN', 7839,
            sysdate, 9999, null, 20);
```

```
select * from My_emp;
```

```
SQL> insert into My_emp(empno, ename, job, mgr,
  2           hiredate, sal, comm, deptno)
  3   values      (665, 'IVAN', 'SALESMAN', 7839,
  4                 sysdate, 9999, null, 20);
```

```
1 row created.
```

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
665	IVAN	SALESMAN	7839	16-MAR-21	9999	(null)	20

## INSERT

2. Додайте в таблицю My\_emp відомості про нового співробітника:

empno - 667

ename - IGOR

hiredate – поточна дата

sal – 9999

deptno – 10

Переконайтесь, що співробітник доданий.

```
insert into My_emp(empno, ename,
                   hiredate, sal, deptno)
values      (667, 'IGOR',
             sysdate, 9999, 10);
```

```
select * from My_emp;
```

```
SQL> insert into My_emp(empno, ename,
  2                     hiredate, sal,    deptno)
  3   values      (667,      'IGOR',
  4                  sysdate, 9999,    10);

1 row created.

SQL> select * from My_emp;

EMPNO ENAME          JOB            MGR HIREDATE        SAL     COMM  DEPTNO
----- -----          -----       -----  -----        -----
 665  IVAN           SALESMAN      7839 16-MAR-21    9999  (null)    20
 667  IGOR          (null)        (null) 16-MAR-21    9999  (null)    10
```

## TRUNCATE

3. Очистіть таблицю My\_emp за допомогою Truncate table.

Переконайтесь, що таблиця My\_emp порожня.

```
truncate table My_emp;
```

```
select * from My_emp;
```

```
SQL> truncate table My_emp;
Table truncated.

SQL> select * from My_emp;
no rows selected
```

## INSERT

4. Додайте всі дані з таблиці Emp в таблицю My\_emp.

Переконайтесь, що всі співробітники додані.

```
insert into My_emp
```

```
select * from emp;
```

```
select * from My_emp;
```

```
SQL> insert into My_emp
  2  select * from emp;
```

18 rows created.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

18 rows selected.

## UPDATE

5. Додайте всім службовцям з таблиці My\_emp \$ 200 зарплати.

**update My\_emp**

```
set sal = sal + 200;
```

```
select * from My_emp;
```

```
SQL> update My_emp
  2  set sal = sal + 200;
```

18 rows updated.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5200	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	3050	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1700	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	3175	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1450	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1800	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1700	0	30
7900	JAMES	CLERK	7698	03-DEC-11	1150	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1450	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3200	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	1000	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3200	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1300	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1500	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2450	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2800	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2700	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2700	(null)	(null)

18 rows selected.

6. \* Одним запитом додайте всім службовцям ще \$ 200 зарплати і встановіть премію \$ 100.

**update My\_emp**

**set sal = sal + 200,**

**comm = 100;**

**select \* from My\_emp;**

```
SQL> update My_emp
  2  set sal = sal + 200,
  3      comm = 100;

18 rows updated.

SQL> select * from My_emp;

  EMPNO ENAME          JOB            MGR HIREDATE        SAL       COMM  DEPTNO
----- -----
    7839 KING            PRESIDENT      (null) 17-NOV-11     5400      100     10
    7698 BLAKE           MANAGER        7839 01-MAY-11    3250      100     30
    7782 CLARK           MANAGER        7839 09-JUN-11    1900      100     10
    7566 JONES           MANAGER        7839 02-APR-11    3375      100     20
    7654 MARTIN          SALESMAN       7698 28-SEP-11   1650      100     30
    7499 ALLEN           SALESMAN       7698 20-FEB-11   2000      100     30
    7844 TURNER          SALESMAN       7698 08-SEP-11   1900      100     30
    7900 JAMES            CLERK          7698 03-DEC-11   1350      100     30
    7521 WARD             SALESMAN       7698 22-FEB-11   1650      100     30
    7902 FORD             ANALYST        7566 03-DEC-11   3400      100     20
    7369 SMITH            CLERK          7902 17-DEC-10   1200      100     20
    7788 SCOTT            ANALYST        7566 09-DEC-12   3400      100     20
    7876 ADAMS            CLERK          7788 12-JAN-13   1500      100     20
    7934 MILLER           CLERK          7782 23-JAN-12   1700      100     10
    8000 JACKIE CHAN      SALESMAN       7839 28-SEP-11   2650      100     60
    8001 JET LI            SALESMAN       8000 20-FEB-11   3000      100     60
    8002 BRUCE LEE         SALESMAN       8000 08-SEP-11   2900      100     60
    8003 DR NO             ANALYST        7839 11-SEP-11   2900      100  (null)

18 rows selected.
```

7. Заберіть у всіх службовців премію. Переконайтесь, що всі зміни проведені правильно.

**update My\_emp**

**set comm = null;**

**select \* from My\_emp;**

```
SQL> update My_emp  
  2  set comm = null;
```

18 rows updated.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5400	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	3250	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1900	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	3375	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1650	(null)	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	2000	(null)	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1900	(null)	30
7900	JAMES	CLERK	7698	03-DEC-11	1350	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1650	(null)	30
7902	FORD	ANALYST	7566	03-DEC-11	3400	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	1200	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3400	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1500	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1700	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2650	(null)	60
8001	JET LI	SALESMAN	8000	20-FEB-11	3000	(null)	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2900	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2900	(null)	(null)

18 rows selected.

## DELETE

8. Видаліть працівників 10 відділу з таблиці My\_Emp.

Переконайтесь, що працівники видалені.

```
SQL> delete from My_emp  
2 where deptno = 10;
```

3 rows deleted.

SQL>

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-11	3250	(null)	30
7566	JONES	MANAGER	7839	02-APR-11	3375	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1650	(null)	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	2000	(null)	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1900	(null)	30
7900	JAMES	CLERK	7698	03-DEC-11	1350	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1650	(null)	30
7902	FORD	ANALYST	7566	03-DEC-11	3400	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	1200	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3400	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1500	(null)	20
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2650	(null)	60
8001	JET LI	SALESMAN	8000	20-FEB-11	3000	(null)	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2900	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2900	(null)	(null)

15 rows selected.

**select \* from My.emp;**

## Завершення роботи

9. Видаліть таблицю My\_emp.

**drop table My\_emp;**

```
SQL> drop table My_emp;  
Table dropped.
```

## NULL vs DEFAULT

10. Створимо таблицю: CREATE TABLE TBL1 (A integer, B integer DEFAULT 5);

Якими будуть результати команд:

a. **Insert into tbl1 values (default, default);**

```
SQL> CREATE TABLE TBL1 (A integer, B integer DEFAULT 5);  
Table created.  
  
SQL> Insert into tbl1 values (default, default);  
1 row created.  
  
SQL> select * from tbl1;  
          A          B  
----- -----  
(null)      5
```

b. **Insert into tbl1 values (null, null);**

```
SQL> Insert into tbl1 values (null, null);  
1 row created.  
  
SQL> select * from tbl1;  
          A          B  
----- -----  
(null)      5  
(null)    (null)
```

c. Insert into tbl1(a) values (default);

```
SQL> Insert into tbl1(a) values (default);

1 row created.

SQL> select * from tbl1;

      A          B
----- -----
(null)      5
(null)  (null)
(null)      5
```

d. Insert into tbl1(b) values (default);

```
SQL> Insert into tbl1(b) values (default);

1 row created.

SQL> select * from tbl1;

      A          B
----- -----
(null)      5
(null)  (null)
(null)      5
(null)      5
```

e. Insert into tbl1(a) values (null);

```
SQL> Insert into tbl1(a) values (null);

1 row created.

SQL> select * from tbl1;

      A          B
----- -----
(null)      5
(null)  (null)
(null)      5
(null)      5
(null)      5
```

f. Insert into tbl1(b) values (null);

```
SQL> Insert into tbl1(b) values (null);
```

```
1 row created.
```

```
SQL> select * from tbl1;
```

A	B
(null)	5
(null)	(null)
(null)	5
(null)	5
(null)	5
(null)	(null)

```
6 rows selected.
```

\*11. Корельовані підзапити в UPDATE і DELETE

a. Створіть таблицю MY\_EMP на основі таблиці EMP і скопіюйте в неї всі дані.

```
create table My_emp as
```

```
select * from emp;
```

```
select * from My_emp;
```

```
SQL> create table My_emp as  
2 select * from emp;
```

```
Table created.
```

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

```
18 rows selected.
```

b. Денормалізуйте таблицю MY\_EMP додавши до неї стовпець dname.

**alter table My\_emp**

**add dname varchar(20);**

```
SQL> alter table My_emp
  2 add dname varchar(20);

Table altered.
```

c. Заповніть цей стовпець на основі даних таблиці DEPT.

**update My\_emp**

**set dname =**

**(select dname**

**from dept**

**where dept.deptno = My\_emp.deptno);**

**select \* from My\_emp;**

```
5
SQL> update My_emp
 2 set dname =
 3      (select dname
 4       from dept
 5      where dept.deptno = My_emp.deptno);

18 rows updated.

SQL> select * from My_emp;

EMPNO ENAME          JOB            MGR HIREDATE        SAL      COMM    DEPTNO DNAME
----- -----
 7839 KING            PRESIDENT     (null) 17-NOV-11   5000  (null)    10 ACCOUNTING
 7698 BLAKE           MANAGER      7839 01-MAY-11   2850  (null)    30 SALES
 7782 CLARK           MANAGER      7839 09-JUN-11   1500  (null)    10 ACCOUNTING
 7566 JONES           MANAGER      7839 02-APR-11   2975  (null)    20 RESEARCH
 7654 MARTIN          SALESMAN     7698 28-SEP-11   1250   1400    30 SALES
 7499 ALLEN           SALESMAN     7698 20-FEB-11   1600    300     30 SALES
 7844 TURNER          SALESMAN     7698 08-SEP-11   1500     0      30 SALES
 7900 JAMES            CLERK        7698 03-DEC-11   950   (null)    30 SALES
 7521 WARD             SALESMAN     7698 22-FEB-11   1250    500     30 SALES
 7902 FORD             ANALYST     7566 03-DEC-11   3000  (null)    20 RESEARCH
 7369 SMITH            CLERK        7902 17-DEC-10   800   (null)    20 RESEARCH
 7788 SCOTT            ANALYST     7566 09-DEC-12   3000  (null)    20 RESEARCH
 7876 ADAMS            CLERK        7788 12-JAN-13   1100  (null)    20 RESEARCH
 7934 MILLER           CLERK        7782 23-JAN-12   1300  (null)    10 ACCOUNTING
 8000 JACKIE CHAN      SALESMAN     7839 28-SEP-11   2250   1700    60 SALES2
 8001 JET LI            SALESMAN     8000 20-FEB-11   2600    600     60 SALES2
 8002 BRUCE LEE         SALESMAN     8000 08-SEP-11   2500  (null)    60 SALES2
 8003 DR NO             ANALYST     7839 11-SEP-11   2500  (null)  (null)  (null)

18 rows selected.
```

d. Видаліть відомості з таблиці MY\_EMP про службовців, які працюють в HONKONG.

**delete**

**from My\_emp**

**where dname = any**

**(select dname**

**from dept**

**where loc = 'HONKONG');**

```
SQL> delete
  2  from My_emp
  3 where dname = any
  4          (select dname
  5           from dept
  6          where loc = 'HONKONG');
```

3 rows deleted.

SQL>

SQL> select \* from My\_emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10	ACCOUNTING
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30	SALES
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10	ACCOUNTING
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20	RESEARCH
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30	SALES
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30	SALES
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30	SALES
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30	SALES
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30	SALES
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20	RESEARCH
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20	RESEARCH
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20	RESEARCH
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20	RESEARCH
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10	ACCOUNTING
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)	(null)

15 rows selected.

e. Видаліть таблицю MY\_EMP.

**drop table My\_emp;**

```
SQL> drop table My_emp;
```

```
Table dropped.
```

## \*12 Завдання

Напишіть запит, якій виведе Ялинку.

Для виведення зірочок скористайтеся функціями CASE, LPAD(RPAD).

*Не створюйте таблицю, яка заповнена зірочками.*

```
SQL> @ 3.sql
YALINKA
-----
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * * *
12 rows selected.
```

```
create table my_fir1(string integer default 1);
```

```
insert into my_fir1 values (1);
```

```
insert into my_fir1 values (2);
```

```
insert into my_fir1 values (3);
```

```
insert into my_fir1 values (4);
```

```
insert into my_fir1 values (5);
```

```
insert into my_fir1 values (6);
```

```
insert into my_fir1 values (7);
```

```
insert into my_fir1 values (8);
```

```
insert into my_fir1 values (9);
```

```
insert into my_fir1 values (10);
```

```
insert into my_fir1 values (11);
```

```
insert into my_fir1 values (12);
```

```
select * from my_fir1;
```

```
select
```

```
case string when 1 then concat(lpad('*', 11, ' '), rpad(' ', 10, ' '))
```

```
when 2 then concat(lpad('**', 11, ' '), rpad('*', 10, ' '))
```

```
when 3 then concat(lpad('***', 11, ' '), rpad('**', 10, ' '))
```

```
when 4 then concat(lpad('****', 11, ' '), rpad('**', 10, ' '))
```

```
when 5 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))
```

```
when 6 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))
```

```
when 7 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))
```

```
when 8 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))  
)
```

```
when 9 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))  
)
```

```
when 10 then concat(lpad('*****', 11, ' '), rpad('**', 10, ' '))  
)
```

```
when 11 then concat(lpad('*', 11, ' '), rpad(' ', 10, ' '))
```

```
when 12 then concat(lpad('*', 11, ' '), rpad(' ', 10, ' '))
```

```
end FIR
```

```
from my_fir1;
```

```
drop table my_fir1;
```

```
SQL> create table my_fir1(string integer default 1);

Table created.

SQL> insert into my_fir1 values (1);

1 row created.

SQL> insert into my_fir1 values (2);

1 row created.

SQL> insert into my_fir1 values (3);

1 row created.

SQL> insert into my_fir1 values (4);

1 row created.

SQL> insert into my_fir1 values (5);

1 row created.

SQL> insert into my_fir1 values (6);

1 row created.

SQL> insert into my_fir1 values (7);

1 row created.

SQL> insert into my_fir1 values (8);

1 row created.

SQL> insert into my_fir1 values (9);

1 row created.

SQL> insert into my_fir1 values (10);

1 row created.
```

```
SQL> insert into my_fir1 values (11);
1 row created.

SQL> insert into my_fir1 values (12);
1 row created.

SQL>
SQL> select * from my_fir1;

  STRING
-----
   1
   2
   3
   4
   5
   6
   7
   8
   9
  10
  11
  12

12 rows selected.

SQL>
SQL> select
```

```
SQL>
SQL> select
  2 case string when 1 then concat(lpad('**', 11, ' '), rpad(' ', 10, ' '))
  3           when 2 then concat(lpad('**', 11, ' '), rpad('* ', 10, ' '))
  4 when 3 then concat(lpad('***', 11, ' '), rpad('** ', 10, ' '))
  5 when 4 then concat(lpad('****', 11, ' '), rpad('*** ', 10, ' '))
  6 when 5 then concat(lpad('*****', 11, ' '), rpad('**** ', 10, ' '))
  7 when 6 then concat(lpad('*****', 11, ' '), rpad('***** ', 10, ' '))
  8 when 7 then concat(lpad('*****', 11, ' '), rpad('***** ', 10, ' '))
  9 when 8 then concat(lpad('*****', 11, ' '), rpad('***** ', 10, ' '))
10 when 9 then concat(lpad('*****', 11, ' '), rpad('***** ', 10, ' '))
11 when 10 then concat(lpad('*****', 11, ' '), rpad('***** ', 10, ' '))
12 when 11 then concat(lpad('*', 11, ' '), rpad(' ', 10, ' '))
13 when 12 then concat(lpad('*', 11, ' '), rpad(' ', 10, ' '))
14 end FIR
15 from my_fir1;
```

FIR

```
-----
*
***
*****
*****
*****
*****
*****
*****
*****
*****
*
*
```

12 rows selected.

```
SQL> drop table my_fir1;
```

Table dropped.

## Лабораторна робота 7

## Екві-з'єднання

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

```
select ename name, loc city
from emp, dept
where emp.deptno = dept.deptno;
```

SQL> select ename name, loc city	
2 from emp, dept	
3 where emp.deptno = dept.deptno;	
NAME	CITY
-----	
KING	NEW_YORK
MILLER	NEW_YORK
CLARK	NEW_YORK
ADAMS	DALLAS
SCOTT	DALLAS
SMITH	DALLAS
FORD	DALLAS
JONES	DALLAS
WARD	CHICAGO
JAMES	CHICAGO
TURNER	CHICAGO
ALLEN	CHICAGO
MARTIN	CHICAGO
BLAKE	CHICAGO
JACKIE CHAN	HONKONG
JET LI	HONKONG
BRUCE LEE	HONKONG
17 rows selected.	

2. Відділу кадрів потрібен звіт що містить ім'я і зарплату для всіх співробітників з Dallas

```
select ename name, sal salary
from emp, dept
where emp.deptno = dept.deptno
and loc = 'DALLAS';
```

SQL> select ename name, sal salary	
2 from emp, dept	
3 where emp.deptno = dept.deptno	
4 and loc = 'DALLAS';	
NAME	SALARY
-----	
ADAMS	1100
SCOTT	3000
SMITH	800
FORD	3000
JONES	2975

## Не-екві з'єднання

3. Створіть запит, який виведе імена і зарплату всіх співробітників із зарплатою вище ніж у Turner.

```
select e.ename name, e.sal salary  
from emp e, emp t  
where e.sal > t.sal  
and t.ename = 'TURNER';
```

```
SQL> select e.ename name, e.sal salary  
2  from emp e, emp t  
3  where e.sal > t.sal  
4    and t.ename = 'TURNER';  
  
NAME          SALARY  
-----  
KING           5000  
BLAKE          2850  
JONES          2975  
ALLEN          1600  
FORD            3000  
SCOTT           3000  
JACKIE CHAN   2250  
JET LI          2600  
BRUCE LEE      2500  
DR NO           2500  
  
10 rows selected.
```

4. Створіть запит, який виведе співробітників із зарплатою нижче ніж у будь-якого з клерків.

```
select distinct e.ename name, e.job, e.sal salary  
from emp e, emp cl  
where e.sal < cl.sal  
and cl.job = 'CLERK';
```

```
SQL> select distinct e.ename name, e.job, e.sal salary  
2  from emp e, emp cl  
3  where e.sal < cl.sal  
4    and cl.job = 'CLERK';  
  
NAME        JOB          SALARY  
-----  
SMITH       CLERK        800  
JAMES       CLERK        950  
MARTIN     SALESMAN    1250  
ADAMS       CLERK        1100  
WARD        SALESMAN    1250
```

Перевірка:

```
select ename name, job, sal salary  
from emp  
where sal < any  
  
(select sal  
from emp  
where job = 'CLERK');
```

SQL> select ename name, job, sal salary		
2 from emp		
3 where sal < any		
4 (select sal		
5 from emp		
6 where job = 'CLERK');		
NAME	JOB	SALARY
SMITH	CLERK	800
JAMES	CLERK	950
ADAMS	CLERK	1100
MARTIN	SALESMAN	1250
WARD	SALESMAN	1250

Само-з'єднання

5. Виведіть на екран таблицю множення для чисел від 1 до 10 використовуючи само-з'єднання. Можливо для цього вам доведеться створити додаткову таблицю. Видаліть її після завершення роботи.

A	B	A*B
---	---	-----
1	1	1
2	1	2
...		
10	10	100

```
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, my_table tb;

drop table my_table;

```

```

SQL> create table my_table(num integer default 1);

Table created.

SQL> insert into my_table values (1);

1 row created.

SQL> insert into my_table values (2);

1 row created.

SQL> insert into my_table values (3);

1 row created.

SQL> insert into my_table values (4);

1 row created.

SQL> insert into my_table values (5);

1 row created.

SQL> insert into my_table values (6);

1 row created.

SQL> insert into my_table values (7);

1 row created.

SQL> insert into my_table values (8);

1 row created.

SQL> insert into my_table values (9);

1 row created.

SQL> insert into my_table values (10);

1 row created.

```

SQL> select * from my_table;		
NUM		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

10 rows selected.

A	B	A*B
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	1	8
9	1	9
10	1	10
1	2	2
2	2	4
3	2	6
4	2	8
5	2	10
6	2	12
7	2	14
8	2	16
9	2	18
10	2	20
1	3	3
2	3	6
3	3	9
4	3	12
5	3	15
6	3	18
7	3	21
8	3	24
9	3	27
10	3	30

	A	B	A*B		A	B	A*B
10	3	30		10	7	70	
1	4	4		1	8	8	
2	4	8		2	8	16	
3	4	12		3	8	24	
4	4	16		4	8	32	
5	4	20					
6	4	24					
7	4	28					
8	4	32					
9	4	36					
10	4	40					
1	5	5					
2	5	10					
3	5	15					
4	5	20					
5	5	25					
6	5	30					
7	5	35					
8	5	40					
9	5	45					
10	5	50					
1	6	6					
2	6	12					
3	6	18					
4	6	24					
5	6	30					
6	6	36					
7	6	42					
8	6	48					
9	6	54					
10	6	60					
1	7	7					
2	7	14					
3	7	21					
4	7	28					
5	7	35					
6	7	42					
7	7	49					
8	7	56					
9	7	63					
10	7	70					
				100 rows selected.			
1	8	8					
2	8	16		SQL> drop table my_table;			
3	8	24					
4	8	32		Table dropped.			

## Завдання

6. Сформувати запит, що виводить номер і прізвище працівника, номер і прізвище його начальника, якщо у працівника є начальник.

```
select w.empno , w.ename worker, e.empno, e.ename employer
from emp w, emp e
where w.mgr = e.empno;
```

```

SQL> select w.empno , w.ename worker, e.empno, e.ename employer
  2  from emp w, emp e
  3  where w.mgr = e.empno;

      EMPNO WORKER          EMPNO EMPLOYER
-----  -----
    7902 FORD              7566 JONES
    7788 SCOTT             7566 JONES
    7499 ALLEN             7698 BLAKE
    7521 WARD              7698 BLAKE
    7654 MARTIN            7698 BLAKE
    7844 TURNER            7698 BLAKE
    7900 JAMES              7698 BLAKE
    7934 MILLER             7782 CLARK
    7876 ADAMS              7788 SCOTT
    7566 JONES             7839 KING
    8003 DR NO              7839 KING
    7698 BLAKE             7839 KING
    7782 CLARK              7839 KING
    8000 JACKIE CHAN        7839 KING
    7369 SMITH              7902 FORD
    8001 JET LI              8000 JACKIE CHAN
    8002 BRUCE LEE           8000 JACKIE CHAN

17 rows selected.

```

7. Сформуйте запит, який повертає номера і назви відділів, в яких є співробітники.

**select d.deptno, d.dname**

**from emp e, dept d**

**where e.deptno = d.deptno**

**and e.empno is not null;**

```

SQL> select d.deptno, d.dname
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  and e.empno is not null;

      DEPTNO DNAME
-----  -----
        10 ACCOUNTING
        30 SALES
        10 ACCOUNTING
        20 RESEARCH
        30 SALES
        30 SALES
        30 SALES
        30 SALES
        20 RESEARCH
        20 RESEARCH
        20 RESEARCH
        20 RESEARCH
        10 ACCOUNTING
        60 SALES2
        60 SALES2
        60 SALES2

17 rows selected.

```

## Завдання 8

- Створіть таблицю My\_emp, що містить дані про всіх співробітників у яких зарплата більше \$ 2500.
- Переконайтесь що всі співробітники додані правильно.
- Виведіть список співробітників, які є в таблиці emp і в таблиці My\_emp.

```
select * from emp;
```

```
create table My_emp as
```

```
select * from emp
```

```
where sal > 2500;
```

```
select * from My_emp;
```

```
select e.empno, e.ename  
from emp e, My_emp me  
where e.ename = me.ename;
```

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

```
18 rows selected.
```

```
SQL> create table My_emp as  
2 select * from emp  
3 where sal > 2500;
```

```
Table created.
```

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60

```
6 rows selected.
```

```

SQL> select e.empno, e.ename
  2  from emp e, My_emp me
  3  where e.ename = me.ename;

    EMPNO ENAME
    -----
      7839 KING
      7698 BLAKE
      7566 JONES
      7902 FORD
      7788 SCOTT
      8001 JET LI

6 rows selected.

```

### Завдання 9

1. Додайте в таблиці My\_emp працівникам відділів №10 та №20 \$200 зарплати.
2. Виведіть список працівників My\_emp, їх нову зарплату та зарплату з таблиці emp.
3. \* Змініть номери відділів у 5 будь-яких (на ваш вибір) працівників из My\_emp.
4. \* Виведіть працівників, у яких змінився відділ, їх старий та новий відділ.
5. Видаліть таблицю My\_emp.

**1.**

**update My\_emp**

**set sal = sal + 200**

**where deptno = 10 or deptno = 20;**

**select \* from My\_emp;**

```

SQL> update My_emp
  2  set sal = sal + 200
  3  where deptno = 10 or deptno = 20;

4 rows updated.

SQL> select * from My_emp;

    EMPNO ENAME          JOB           MGR HIREDATE        SAL       COMM  DEPTNO
    -----
      7839 KING        PRESIDENT    (null) 17-NOV-11     5200  (null)      10
      7698 BLAKE       MANAGER     7839 01-MAY-11     2850  (null)      30
      7566 JONES       MANAGER     7839 02-APR-11     3175  (null)      20
      7902 FORD        ANALYST    7566 03-DEC-11     3200  (null)      20
      7788 SCOTT       ANALYST    7566 09-DEC-12     3200  (null)      20
      8001 JET LI      SALESMAN   8000 20-FEB-11     2600      600      60

6 rows selected.

```

2.

**select me.ename, me.sal, e.sal**

**from emp e, My\_emp me**

**where e.empno = me.empno;**

```
SQL> select me.ename, me.sal, e.sal
  2  from emp e, My_emp me
  3  where e.empno = me.empno;
```

ENAME	SAL	SAL
JONES	3175	2975
BLAKE	2850	2850
SCOTT	3200	3000
KING	5200	5000
FORD	3200	3000
JET LI	2600	2600

6 rows selected.

3.

**update My\_emp**

**set deptno = 3**

**where comm is null;**

**select \* from My\_emp;**

```
SQL> update My_emp
  2  set deptno = 3
  3  where comm is null;
```

5 rows updated.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5200	(null)	3
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	3
7566	JONES	MANAGER	7839	02-APR-11	3175	(null)	3
7902	FORD	ANALYST	7566	03-DEC-11	3200	(null)	3
7788	SCOTT	ANALYST	7566	09-DEC-12	3200	(null)	3
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60

6 rows selected.

4.

```
select me.ename, e.deptno, me.deptno  
from emp e, My_emp me  
where e.deptno != me.deptno  
and e.empno = me.empno;
```

```
SQL> select me.ename, e.deptno, me.deptno  
  2  from emp e, My_emp me  
  3  where e.deptno != me.deptno  
  4    and e.empno = me.empno;  
  
ENAME          DEPTNO      DEPTNO  
-----  -----  
JONES           20          3  
BLAKE           30          3  
SCOTT           20          3  
KING            10          3  
FORD            20          3
```

5.

```
drop table My_emp;
```

```
SQL> drop table My_emp;  
Table dropped.
```

Завдання 10

1. \* Виведіть список співробітників, які є начальниками.
2. Виведіть список підрозділів, в яких працює хоча б один начальник.
3. \* Виведіть список міст, в яких працює хоча б один начальник.

1.

```
select e.empno, e.ename employer  
from emp w, emp e  
where w.mgr = e.empno;
```

```
select distinct e.empno, e.ename employer  
from emp w, emp e  
where w.mgr = e.empno;
```

```

SQL> select e.empno, e.ename employer
  2  from emp w, emp e
  3  where w.mgr = e.empno;

          EMPNO EMPLOYER
-----
      7566 JONES
      7566 JONES
      7698 BLAKE
      7698 BLAKE
      7698 BLAKE
      7698 BLAKE
      7698 BLAKE
      7782 CLARK
      7788 SCOTT
      7839 KING
      7839 KING
      7839 KING
      7839 KING
      7902 FORD
     8000 JACKIE CHAN
     8000 JACKIE CHAN

17 rows selected.

SQL> select distinct e.empno, e.ename employer
  2  from emp w, emp e
  3  where w.mgr = e.empno;

          EMPNO EMPLOYER
-----
      7698 BLAKE
      7788 SCOTT
      7839 KING
     8000 JACKIE CHAN
      7782 CLARK
      7566 JONES
      7902 FORD

7 rows selected.

```

## 2.

**select e.deptno**

**from emp w, emp e**

**where w.mgr = e.empno;**

**select distinct e.deptno**

**from emp w, emp e**

**where w.mgr = e.empno;**

```
SQL> select e.deptno
  2  from emp w, emp e
  3  where w.mgr = e.empno;

        DEPTNO
-----
       20
       20
       30
       30
       30
       30
       30
       30
       10
       20
       10
       10
       10
       10
       20
       60
       60

17 rows selected.

SQL>
SQL> select distinct e.deptno
  2  from emp w, emp e
  3  where w.mgr = e.empno;

        DEPTNO
-----
       20
       60
       30
       10
```

### 3.

**select d.loc**

**from emp w, emp e, dept d**

**where w.mgr = e.empno**

**and e.deptno = d.deptno;**

**select distinct d.loc**

**from emp w, emp e, dept d**

**where w.mgr = e.empno**

**and e.deptno = d.deptno;**

```

SQL> select d.loc
  2  from emp w, emp e, dept d
  3  where w.mgr = e.empno
  4    and e.deptno = d.deptno;

LOC
-----
NEW_YORK
NEW_YORK
NEW_YORK
CHICAGO
CHICAGO
CHICAGO
CHICAGO
CHICAGO
DALLAS
DALLAS
DALLAS
DALLAS
NEW_YORK
NEW_YORK
HONKONG
HONKONG
NEW_YORK

17 rows selected.

SQL>
SQL> select distinct d.loc
  2  from emp w, emp e, dept d
  3  where w.mgr = e.empno
  4    and e.deptno = d.deptno;

LOC
-----
NEW_YORK
CHICAGO
DALLAS
HONKONG

```

### Завдання 11\*

- Створіть запит, який виведе всі пари імен співробітників, що працюють в одному відділі. При цьому, не включати в список комбінації імен співробітників самих з собою (тобто комбінацію типу «Іванов-Іванов») і комбінації імен співробітників, що відрізняються порядком розташування (тобто включати одну з двох комбінацій типу «ІвановПетров» і «Петров-Іванов»).
- Модифікуйте запит 12.1, так щоб пари будувалися для співробітників, які працюють в одному місті, а не в одному відділі.

1.

```
select w1.ename ||' - '|| w2.ename as "WORKER - WORKER"
from emp w1, emp w2
where w1.deptno = w2.deptno
and w1.ename != w2.ename;
```

```
SQL> select w1.ename ||' - '|| w2.ename as "WORKER - WORKER"
2  from emp w1, emp w2
3  where w1.deptno = w2.deptno
4    and w1.ename != w2.ename;
```

```
WORKER - WORKER
```

```
CLARK - KING
MILLER - KING
MARTIN - BLAKE
ALLEN - BLAKE
TURNER - BLAKE
JAMES - BLAKE
WARD - BLAKE
KING - CLARK
MILLER - CLARK
FORD - JONES
SMITH - JONES
SCOTT - JONES
ADAMS - JONES
BLAKE - MARTIN
ALLEN - MARTIN
TURNER - MARTIN
JAMES - MARTIN
WARD - MARTIN
BLAKE - ALLEN
MARTIN - ALLEN
TURNER - ALLEN
JAMES - ALLEN
WARD - ALLEN
BLAKE - TURNER
MARTIN - TURNER
ALLEN - TURNER
JAMES - TURNER
WARD - TURNER
BLAKE - JAMES
MARTIN - JAMES
ALLEN - JAMES
TURNER - JAMES
WARD - JAMES
BLAKE - WARD
MARTIN - WARD
ALLEN - WARD
TURNER - WARD
```

```
WORKER - WORKER
```

```
WORKER - WORKER
```

```
JAMES - WARD
JONES - FORD
SMITH - FORD
SCOTT - FORD
ADAMS - FORD
JONES - SMITH
FORD - SMITH
SCOTT - SMITH
ADAMS - SMITH
JONES - SCOTT
FORD - SCOTT
SMITH - SCOTT
ADAMS - SCOTT
JONES - ADAMS
FORD - ADAMS
SMITH - ADAMS
SCOTT - ADAMS
KING - MILLER
CLARK - MILLER
JET LI - JACKIE CHAN
BRUCE LEE - JACKIE CHAN
JACKIE CHAN - JET LI
BRUCE LEE - JET LI
JACKIE CHAN - BRUCE LEE
JET LI - BRUCE LEE
```

```
62 rows selected.
```

2.

```
select distinct w1.ename ||' - '|| w2.ename as "WORKER - WORKER"
from emp w1, emp w2, dept d1, dept d2
where d1.loc = d2.loc
and w1.deptno is not null
and w2.deptno is not null
and w1.ename != w2.ename;
```

```
SQL> select distinct w1.ename ||' - '|| w2.ename as "WORKER - WORKER"
  2  from emp w1, emp w2, dept d1, dept d2
  3  where d1.loc = d2.loc
  4    and w1.deptno is not null
  5    and w2.deptno is not null
  6    and w1.ename != w2.ename;
```

WORKER - WORKER

```
-----  
KING - BLAKE  
KING - JONES  
KING - JET LI  
BLAKE - KING  
BLAKE - SCOTT  
BLAKE - BRUCE LEE  
CLARK - ADAMS  
JONES - TURNER  
JONES - JAMES  
JONES - FORD  
MARTIN - MILLER  
ALLEN - CLARK  
TURNER - ADAMS  
TURNER - JET LI  
JAMES - ALLEN  
WARD - FORD  
WARD - BRUCE LEE  
SMITH - TURNER  
SMITH - JAMES  
SMITH - WARD  
SCOTT - MILLER  
ADAMS - TURNER  
ADAMS - FORD  
ADAMS - MILLER  
JACKIE CHAN - MARTIN  
JACKIE CHAN - TURNER  
JET LI - KING  
JET LI - CLARK  
JET LI - JAMES  
BRUCE LEE - TURNER  
BRUCE LEE - JAMES  
KING - TURNER  
BLAKE - FORD  
CLARK - BLAKE  
JONES - BLAKE  
JONES - CLARK  
ALLEN - JONES
```

WORKER - WORKER	WORKER - WORKER	WORKER - WORKER
ALLEN - JAMES	MARTIN - FORD	ALLEN - FORD
TURNER - BLAKE	MARTIN - JACKIE CHAN	TURNER - ALLEN
JAMES - CLARK	MARTIN - BRUCE LEE	TURNER - BRUCE LEE
JAMES - WARD	ALLEN - KING	JAMES - KING
JAMES - SMITH	TURNER - JAMES	JAMES - JET LI
JAMES - JACKIE CHAN	TURNER - MILLER	WARD - MILLER
WARD - JONES	JAMES - JONES	FORD - MARTIN
WARD - SCOTT	JAMES - MARTIN	FORD - TURNER
WARD - ADAMS	JAMES - TURNER	FORD - JAMES
FORD - CLARK	FORD - WARD	SMITH - ALLEN
FORD - ALLEN	FORD - BRUCE LEE	SMITH - JACKIE CHAN
FORD - JET LI	SCOTT - TURNER	SCOTT - MARTIN
SMITH - BLAKE	SCOTT - SMITH	MILLER - JONES
SMITH - MILLER	SCOTT - JACKIE CHAN	MILLER - ADAMS
SCOTT - CLARK	ADAMS - KING	JACKIE CHAN - JET LI
SCOTT - JAMES	ADAMS - BLAKE	JET LI - BRUCE LEE
SCOTT - JET LI	ADAMS - WARD	BRUCE LEE - KING
ADAMS - JONES	ADAMS - SCOTT	BRUCE LEE - MARTIN
ADAMS - SMITH	ADAMS - JET LI	BRUCE LEE - ALLEN
MILLER - BLAKE	MILLER - KING	BRUCE LEE - SMITH
MILLER - JAMES	MILLER - ALLEN	BRUCE LEE - SCOTT
MILLER - FORD	MILLER - TURNER	BRUCE LEE - ADAMS
JACKIE CHAN - ALLEN	MILLER - SCOTT	BRUCE LEE - JACKIE CHAN
KING - JAMES	JACKIE CHAN - JAMES	KING - MARTIN
KING - SCOTT	JACKIE CHAN - WARD	KING - FORD
KING - MILLER	JACKIE CHAN - SMITH	KING - ADAMS
BLAKE - JONES	JET LI - BLAKE	BLAKE - ADAMS
BLAKE - ALLEN	JET LI - SMITH	CLARK - ALLEN
CLARK - JONES	KING - ALLEN	JONES - MILLER
CLARK - SCOTT	BLAKE - JAMES	MARTIN - JET LI
CLARK - BRUCE LEE	BLAKE - JACKIE CHAN	ALLEN - MARTIN
JONES - SCOTT	CLARK - KING	ALLEN - TURNER
JONES - JET LI	CLARK - FORD	ALLEN - SCOTT
MARTIN - BLAKE	CLARK - SMITH	TURNER - JONES
MARTIN - JONES	JONES - KING	TURNER - MARTIN
MARTIN - ALLEN	JONES - JACKIE CHAN	TURNER - SCOTT
MARTIN - WARD	MARTIN - SMITH	JAMES - SCOTT

WORKER - WORKER	WORKER - WORKER	WORKER - WORKER
WARD - KING	JAMES - MILLER	WARD - CLARK
FORD - KING	JAMES - BRUCE LEE	WARD - MARTIN
FORD - ADAMS	WARD - ALLEN	WARD - TURNER
FORD - MILLER	WARD - JAMES	WARD - JACKIE CHAN
FORD - JACKIE CHAN	WARD - SMITH	FORD - JONES
SMITH - KING	WARD - JET LI	SMITH - MARTIN
SMITH - CLARK	FORD - BLAKE	SCOTT - KING
SMITH - FORD	FORD - SMITH	ADAMS - MARTIN
SCOTT - BLAKE	SMITH - JONES	ADAMS - ALLEN
SCOTT - JONES	SMITH - ADAMS	ADAMS - JACKIE CHAN
SCOTT - FORD	SMITH - BRUCE LEE	MILLER - CLARK
SCOTT - ADAMS	SCOTT - WARD	MILLER - SMITH
ADAMS - BRUCE LEE	ADAMS - JAMES	MILLER - BRUCE LEE
MILLER - MARTIN	MILLER - JACKIE CHAN	JACKIE CHAN - KING
JACKIE CHAN - JONES	MILLER - JET LI	JET LI - MARTIN
JACKIE CHAN - FORD	JACKIE CHAN - BLAKE	JET LI - TURNER
JACKIE CHAN - MILLER	JACKIE CHAN - CLARK	JET LI - SCOTT
JET LI - FORD	JACKIE CHAN - SCOTT	BRUCE LEE - FORD
JET LI - MILLER	JACKIE CHAN - ADAMS	BRUCE LEE - MILLER
BRUCE LEE - BLAKE	JACKIE CHAN - BRUCE LEE	KING - SMITH
BRUCE LEE - JONES	JET LI - ALLEN	KING - JACKIE CHAN
BRUCE LEE - JET LI	BRUCE LEE - CLARK	KING - BRUCE LEE
KING - CLARK	BRUCE LEE - WARD	BLAKE - CLARK
BLAKE - TURNER	KING - WARD	BLAKE - MARTIN
BLAKE - WARD	CLARK - MARTIN	BLAKE - SMITH
JONES - ALLEN	CLARK - TURNER	BLAKE - MILLER
JONES - WARD	CLARK - JAMES	BLAKE - JET LI
JONES - ADAMS	CLARK - WARD	CLARK - MILLER
MARTIN - CLARK	CLARK - JACKIE CHAN	JONES - MARTIN
MARTIN - SCOTT	CLARK - JET LI	JONES - SMITH
ALLEN - WARD	ALLEN - BLAKE	JONES - BRUCE LEE
ALLEN - ADAMS	ALLEN - SMITH	MARTIN - KING
ALLEN - MILLER	ALLEN - JACKIE CHAN	MARTIN - TURNER
ALLEN - BRUCE LEE	TURNER - CLARK	MARTIN - JAMES
TURNER - FORD	TURNER - JACKIE CHAN	MARTIN - ADAMS
TURNER - SMITH	JAMES - ADAMS	ALLEN - JET LI
JAMES - BLAKE	WARD - BLAKE	TURNER - KING

WORKER - WORKER
TURNER - WARD
JAMES - FORD
FORD - SCOTT
SMITH - SCOTT
SMITH - JET LI
SCOTT - ALLEN
SCOTT - BRUCE LEE
ADAMS - CLARK
MILLER - WARD
JET LI - JONES
JET LI - WARD
JET LI - ADAMS
JET LI - JACKIE CHAN

272 rows selected.

## 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;**

```
SQL> select ename name, loc city
  2  from emp
  3  natural join dept;
```

NAME	CITY
KING	NEW_YORK
MILLER	NEW_YORK
CLARK	NEW_YORK
ADAMS	DALLAS
SCOTT	DALLAS
SMITH	DALLAS
FORD	DALLAS
JONES	DALLAS
WARD	CHICAGO
JAMES	CHICAGO
TURNER	CHICAGO
ALLEN	CHICAGO
MARTIN	CHICAGO
BLAKE	CHICAGO
JACKIE CHAN	HONKONG
JET LI	HONKONG
BRUCE LEE	HONKONG

17 rows selected.

2. Виконайте завдання 2 тричі: використовуючи

NATURAL JOIN, JOIN ... ON i JOIN ... USING:

Відділу кадрів потрібен звіт, який 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';**

```

SQL> select ename name, sal salary, deptno
  2  from emp
  3  natural join dept
  4  where loc = 'DALLAS';

```

NAME	SALARY	DEPTNO
ADAMS	1100	20
SCOTT	3000	20
SMITH	800	20
FORD	3000	20
JONES	2975	20

**select ename name, sal salary, e.deptno  
from emp e  
join dept d on (d.deptno=e.deptno)  
where loc = 'DALLAS';**

```

SQL> select ename name, sal salary, e.deptno
  2  from emp e
  3  join dept d on (d.deptno=e.deptno)
  4  where loc = 'DALLAS';

```

NAME	SALARY	DEPTNO
ADAMS	1100	20
SCOTT	3000	20
SMITH	800	20
FORD	3000	20
JONES	2975	20

**select ename name, sal salary, deptno  
from emp e  
join dept using (deptno)  
where loc = 'DALLAS';**

```

SQL> select ename name, sal salary, deptno
  2  from emp e
  3  join dept using (deptno)
  4  where loc = 'DALLAS';

```

NAME	SALARY	DEPTNO
ADAMS	1100	20
SCOTT	3000	20
SMITH	800	20
FORD	3000	20
JONES	2975	20

## 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);
```

```
SQL> select worker.empno, worker.ename worker, manager.empno, manager.ename manager
  2  from emp worker
  3 join emp manager on (worker.mgr = manager.empno);

EMPNO WORKER          EMPNO MANAGER
-----  -----
 7902 FORD            7566 JONES
 7788 SCOTT           7566 JONES
 7499 ALLEN           7698 BLAKE
 7521 WARD            7698 BLAKE
 7654 MARTIN          7698 BLAKE
 7844 TURNER          7698 BLAKE
 7900 JAMES            7698 BLAKE
 7934 MILLER          7782 CLARK
 7876 ADAMS           7788 SCOTT
 7566 JONES           7839 KING
 8003 DR NO            7839 KING
 7698 BLAKE           7839 KING
 7782 CLARK           7839 KING
 8000 JACKIE CHAN      7839 KING
 7369 SMITH           7902 FORD
 8001 JET LI            8000 JACKIE CHAN
 8002 BRUCE LEE         8000 JACKIE CHAN

17 rows selected.
```

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%';
```

```
SQL> select ename name
  2  from emp e
  3 join dept using (deptno)
  4 where loc like '%O%';

NAME
-----
KING
MILLER
CLARK
WARD
JAMES
TURNER
ALLEN
MARTIN
BLAKE
JACKIE CHAN
JET LI
BRUCE LEE

12 rows selected.
```

## 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);
```

```
SQL> select worker.empno, worker.ename worker, manager.empno, manager.ename manager  
2  from emp manager  
3  right outer join emp worker  
4  on (worker.mgr = manager.empno);  
  
EMPNO WORKER          EMPNO MANAGER  
-----  
7698 BLAKE            7839 KING  
7782 CLARK            7839 KING  
7566 JONES            7839 KING  
8000 JACKIE CHAN      7839 KING  
8003 DR NO             7839 KING  
7654 MARTIN           7698 BLAKE  
7499 ALLEN            7698 BLAKE  
7844 TURNER           7698 BLAKE  
7900 JAMES             7698 BLAKE  
7521 WARD              7698 BLAKE  
7934 MILLER           7782 CLARK  
7902 FORD              7566 JONES  
7788 SCOTT             7566 JONES  
7369 SMITH             7902 FORD  
7876 ADAMS             7788 SCOTT  
8001 JET LI             8000 JACKIE CHAN  
8002 BRUCE LEE         8000 JACKIE CHAN  
7839 KING              (null)    (null)  
  
18 rows selected.
```

```
select worker.empno, worker.ename worker, manager.empno, manager.ename manager  
from emp worker  
left outer join emp manager  
on (worker.mgr = manager.empno);
```

```

SQL> select worker.empno, worker.ename worker, manager.empno, manager.ename manager
  2  from emp worker
  3  left outer join emp manager
  4  on (worker.mgr = manager.empno);

```

EMPNO WORKER	EMPNO MANAGER
7698 BLAKE	7839 KING
7782 CLARK	7839 KING
7566 JONES	7839 KING
8000 JACKIE CHAN	7839 KING
8003 DR NO	7839 KING
7654 MARTIN	7698 BLAKE
7499 ALLEN	7698 BLAKE
7844 TURNER	7698 BLAKE
7900 JAMES	7698 BLAKE
7521 WARD	7698 BLAKE
7934 MILLER	7782 CLARK
7902 FORD	7566 JONES
7788 SCOTT	7566 JONES
7369 SMITH	7902 FORD
7876 ADAMS	7788 SCOTT
8001 JET LI	8000 JACKIE CHAN
8002 BRUCE LEE	8000 JACKIE CHAN
7839 KING	(null)

18 rows selected.

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;

```

```

SQL> select d.deptno, dname
  2  from emp e
  3  right outer join dept d
  4  on (d.deptno = e.deptno)
  5  where e.deptno is null;

DEPTNO DNAME
----- -----
      50 RESEARCH2
     100 SALES3
      40 OPERATIONS

```

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;

```

```

SQL> select manager.empno, manager.ename
  2  from emp manager
  3  left outer join emp worker
  4  on (worker.mgr = manager.empno)
  5  where worker.empno is null;

    EMPNO ENAME
    -----
    7369 SMITH
    7499 ALLEN
    7521 WARD
    7654 MARTIN
    7844 TURNER
    7876 ADAMS
    7900 JAMES
    7934 MILLER
    8001 JET LI
    8002 BRUCE LEE
    8003 DR NO

11 rows selected.

```

## 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)

A	B	A*B
1	1	1
2	1	2
...		
10	10	100

```
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;
```

```
SQL> create table my_table(num integer default 1);  
Table created.  
  
SQL> insert into my_table values (1);  
1 row created.  
  
SQL> insert into my_table values (2);  
1 row created.  
  
SQL> insert into my_table values (3);  
1 row created.  
  
SQL> insert into my_table values (4);  
1 row created.  
  
SQL> insert into my_table values (5);  
1 row created.  
  
SQL> insert into my_table values (6);  
1 row created.  
  
SQL> insert into my_table values (7);  
1 row created.  
  
SQL> insert into my_table values (8);  
1 row created.  
  
SQL> insert into my_table values (9);  
1 row created.  
  
SQL> insert into my_table values (10);  
1 row created.
```

```
SQL> select * from my_table;
```

NUM
1
2
3
4
5
6
7
8
9
10

```
10 rows selected.
```

```
SQL> select tb.num A, ta.num B, tb.num*ta.num as "A*B"  
2  from my_table ta  
3  cross join my_table tb;
```

A	B	A*B
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	1	8
9	1	9
10	1	10
1	2	2
2	2	4
3	2	6
4	2	8
5	2	10
6	2	12
7	2	14
8	2	16
9	2	18
10	2	20
1	3	3
2	3	6

			1	7	7
			2	7	14
			3	7	21
10	2	20	4	7	28
1	3	3	5	7	35
2	3	6	6	7	42
3	3	9	7	7	49
4	3	12	8	7	56
5	3	15	9	7	63
6	3	18	10	7	70
7	3	21	1	8	8
8	3	24	2	8	16
9	3	27	3	8	24
10	3	30	4	8	32
1	4	4	A	B	A*B
2	4	8			
3	4	12			
4	4	16	5	8	40
5	4	20	6	8	48
6	4	24	7	8	56
7	4	28	8	8	64
			9	8	72
A	B	A*B	10	8	80
			1	9	9
8	4	32	2	9	18
9	4	36	3	9	27
10	4	40	4	9	36
1	5	5	5	9	45
2	5	10	6	9	54
3	5	15	7	9	63
4	5	20	8	9	72
5	5	25	9	9	81
6	5	30	10	9	90
7	5	35	1	10	10
8	5	40	2	10	20
9	5	45	3	10	30
10	5	50	4	10	40
1	6	6	5	10	50
2	6	12	6	10	60
3	6	18	7	10	70
4	6	24	8	10	80
5	6	30	9	10	90
6	6	36	10	10	100
7	6	42			
8	6	48			
9	6	54			
10	6	60			
1	7	7			
2	7	14			

100 rows selected.

SQL> drop table my\_table;

Table dropped.

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;
```

```
SQL> select worker.ename worker  
      2  from emp worker  
      3  join emp manager on (worker.mgr = manager.empno)  
      4  join dept dw on (worker.deptno = dw.deptno)  
      5  join dept dm on (manager.deptno = dm.deptno)  
      6  where dw.deptno != dm.deptno;  
  
WORKER  
-----  
JONES  
BLAKE  
JACKIE CHAN
```

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;
```

```
SQL> select worker.empno, worker.ename worker, dw.loc, manager.empno, manager.ename manager, dm.loc  
      2  from emp worker  
      3  join emp manager on (worker.mgr = manager.empno)  
      4  join dept dw on (worker.deptno = dw.deptno)  
      5  join dept dm on (manager.deptno = dm.deptno)  
      6  where dw.deptno != dm.deptno;  
  
EMPNO WORKER          LOC        EMPNO MANAGER    LOC  
-----  -----  
7566 JONES           DALLAS      7839 KING     NEW_YORK  
7698 BLAKE           CHICAGO    7839 KING     NEW_YORK  
8000 JACKIE CHAN    HONKONG    7839 KING     NEW_YORK
```

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;
```

```
SQL> select manager.ename manager  
  2  from emp manager  
  3  join emp worker on (worker.mgr = manager.empno)  
  4  join dept dw on (worker.deptno = dw.deptno)  
  5  join dept dm on (manager.deptno = dm.deptno)  
  6  where dw.deptno != dm.deptno;  
  
MANAGER  
-----  
KING  
KING  
KING
```

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;
```

```
SQL> select worker.empno, worker.ename worker, dw.loc, manager.empno, manager.ename manager, dm.loc  
  2  from emp manager  
  3  join emp worker on (worker.mgr = manager.empno)  
  4  join dept dw on (worker.deptno = dw.deptno)  
  5  join dept dm on (manager.deptno = dm.deptno)  
  6  where dw.deptno != dm.deptno;  
  
EMPNO WORKER          LOC      EMPNO MANAGER      LOC  
-----  
7566 JONES           DALLAS    7839 KING        NEW_YORK  
7698 BLAKE           CHICAGO   7839 KING        NEW_YORK  
8000 JACKIE CHAN     HONKONG   7839 KING        NEW_YORK
```

## Комплексне завдання 1. «Межі»

1. Створіть таблицю Job\_Salegrade:

Посада	Мінімальна Зарплата	Максимальна Зарплата
Manager	1000	2000
Salesman	1000	5000
Clerk	500	1200
Analyst	2000	5000

```
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;
```

```
SQL> create table Job_Salegrade(job varchar(20) not null,
  2                               min_salary NUMBER(4) not null,
  3                               max_salary NUMBER(4) not null);

Table created.

SQL> insert into Job_Salegrade values('MANAGER', 1000, 2000);

1 row created.

SQL> insert into Job_Salegrade values('SALESMAN', 1000, 5000);

1 row created.

SQL> insert into Job_Salegrade values('CLERK', 500, 1200);

1 row created.

SQL> insert into Job_Salegrade values('ANALYST', 2000, 5000);

1 row created.

SQL> select * from Job_Salegrade;

JOB          MIN_SALARY  MAX_SALARY
-----  -----  -----
MANAGER          1000      2000
SALESMAN         1000      5000
CLERK            500       1200
ANALYST          2000      5000
```

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;
```

```
SQL> select ename name, e.job, sal salary, max_salary  
2  from emp e  
3  join Job_Salegrade j on (j.job = e.job)  
4  where sal > max_salary;
```

NAME	JOB	SALARY	MAX_SALARY
BLAKE	MANAGER	2850	2000
JONES	MANAGER	2975	2000
MILLER	CLERK	1300	1200

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;
```

```
SQL> select ename name, e.job, sal salary, min_salary, max_salary  
2  from emp e  
3  join Job_Salegrade j on (j.job = e.job)  
4  where sal not between min_salary and max_salary;
```

NAME	JOB	SALARY	MIN_SALARY	MAX_SALARY
BLAKE	MANAGER	2850	1000	2000
JONES	MANAGER	2975	1000	2000
MILLER	CLERK	1300	500	1200

4. Створіть представлення (view) empl\_and\_salegrade на основі з'єднання таблиць Job\_Salegrade 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;
```

```

SQL> create view empl_and_salegrade
2 as select empno, ename, e.job, mgr, hiredate, sal, min_salary, max_salary, comm, deptno
3 from emp e
4 join Job_Salegrade j on (j.job = e.job);

```

View created.

```
SQL> select * from empl_and_salegrade;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	MIN_SALARY	MAX_SALARY	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	1000	2000	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	1000	2000	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	1000	2000	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1000	5000	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	1000	5000	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	1000	5000	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	500	1200	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	1000	5000	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	2000	5000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	500	1200	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	2000	5000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	500	1200	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	500	1200	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1000	5000	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	1000	5000	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	1000	5000	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	2000	5000	(null)	(null)

17 rows selected.

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

A) 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;**

```

SQL> select loc
  2  from emp e
  3  join dept d on (d.deptno = e.deptno)
  4  join Job_Salegrade j on (j.job = e.job)
  5  where sal not between min_salary and max_salary;

LOC
-----
CHICAGO
DALLAS
NEW_YORK

```

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);**

```

SQL> select ename name, sal salary, min_salary, max_salary, e.deptno, loc
  2  from emp e
  3  join dept d on (d.deptno = e.deptno)
  4  join Job_Salegrade j on (j.job = e.job);

```

NAME	SALARY	MIN_SALARY	MAX_SALARY	DEPTNO	LOC
CLARK	1500	1000	2000	10	NEW_YORK
JONES	2975	1000	2000	20	DALLAS
BLAKE	2850	1000	2000	30	CHICAGO
WARD	1250	1000	5000	30	CHICAGO
TURNER	1500	1000	5000	30	CHICAGO
ALLEN	1600	1000	5000	30	CHICAGO
MARTIN	1250	1000	5000	30	CHICAGO
JACKIE CHAN	2250	1000	5000	60	HONKONG
JET LI	2600	1000	5000	60	HONKONG
BRUCE LEE	2500	1000	5000	60	HONKONG
MILLER	1300	500	1200	10	NEW_YORK
ADAMS	1100	500	1200	20	DALLAS
SMITH	800	500	1200	20	DALLAS
JAMES	950	500	1200	30	CHICAGO
SCOTT	3000	2000	5000	20	DALLAS
FORD	3000	2000	5000	20	DALLAS

16 rows selected.

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;

```

```

SQL> select loc
  2  from empl_and_salegrade es
  3  join dept d on (d.deptno = es.deptno)
  4  where sal not between min_salary and max_salary;

LOC
-----
CHICAGO
DALLAS
NEW_YORK

```

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);

```

```

SQL> select ename name, sal salary, min_salary, max_salary, es.deptno, loc
  2  from empl_and_salegrade es
  3  join dept d on (d.deptno = es.deptno);

```

NAME	SALARY	MIN_SALARY	MAX_SALARY	DEPTNO	LOC
CLARK	1500	1000	2000	10	NEW_YORK
JONES	2975	1000	2000	20	DALLAS
BLAKE	2850	1000	2000	30	CHICAGO
WARD	1250	1000	5000	30	CHICAGO
TURNER	1500	1000	5000	30	CHICAGO
ALLEN	1600	1000	5000	30	CHICAGO
MARTIN	1250	1000	5000	30	CHICAGO
JACKIE CHAN	2250	1000	5000	60	HONKONG
JET LI	2600	1000	5000	60	HONKONG
BRUCE LEE	2500	1000	5000	60	HONKONG
MILLER	1300	500	1200	10	NEW_YORK
ADAMS	1100	500	1200	20	DALLAS
SMITH	800	500	1200	20	DALLAS
JAMES	950	500	1200	30	CHICAGO
SCOTT	3000	2000	5000	20	DALLAS
FORD	3000	2000	5000	20	DALLAS

16 rows selected.

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);**

```

SQL> select loc
  2  from empl_and_salegrade es1
  3  join dept d on (d.deptno = es1.deptno)
  4  where es1.sal between es1.min_salary and es1.max_salary
  5  and es1.deptno not in
  6          (select es2.deptno
  7           from empl_and_salegrade es2
  8           where es2.sal not between es2.min_salary and es2.max_salary);

```

**LOC**

-----  
HONKONG  
HONKONG  
HONKONG

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);
```

```
SQL> select es1.ename name, es1.sal salary, es1.min_salary, es1.max_salary, es1.deptno, loc
  2  from empl_and_salegrade es1
  3  join dept d on (d.deptno = es1.deptno)
  4  where es1.sal between es1.min_salary and es1.max_salary
  5  and es1.deptno not in
  6      (select es2.deptno
  7       from empl_and_salegrade es2
  8       where es2.sal not between es2.min_salary and es2.max_salary);

NAME          SALARY MIN_SALARY MAX_SALARY    DEPTNO LOC
-----        -----  -----  -----      -----
JACKIE CHAN      2250      1000      5000      60 HONKONG
JET LI           2600      1000      5000      60 HONKONG
BRUCE LEE         2500      1000      5000      60 HONKONG
```

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

```
drop view empl_and_salegrade;
```

```
drop table Job_Salegrade;
```

```
SQL> drop view empl_and_salegrade;
View dropped.

SQL> drop table Job_Salegrade;
Table dropped.
```

## Комплексне завдання 2. «Відпустки»

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

Empno	Начало	Окончание
7839	2013-01-07	2013-01-15
7839	2013-01-24	2013-02-17
7698	2013-02-15	2013-02-24
7934	2013-02-16	2013-02-19

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;
```

```
SQL> select * from vacation;  
  
EMPNO START_DATE END_DATE  
-----  
7839 07-JAN-13 15-JAN-13  
7839 24-JAN-13 17-FEB-13  
7698 05-FEB-13 24-FEB-13  
7934 16-FEB-13 19-FEB-13  
7839 07-JAN-13 08-JAN-13  
7782 03-JAN-15 16-MAR-15  
7566 07-FEB-15 08-MAR-15  
8000 19-JAN-15 05-APR-15  
7900 01-MAR-15 20-MAR-15  
7844 25-FEB-15 16-MAR-15  
7844 07-MAR-13 14-APR-13  
8002 09-JAN-14 10-JAN-14  
8002 01-SEP-14 09-SEP-14  
7902 07-JUN-13 09-JUN-13  
8001 17-JAN-13 27-JAN-13  
  
15 rows selected.
```

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;
```

```
SQL> select mn.empno, mn.ename manager  
2  from emp wr  
3  left outer join emp mn on (wr.mgr = mn.empno)  
4  left outer join vacation vcwr on (wr.empno = vcwr.empno)  
5  left outer join vacation vcmn on (mn.empno = vcmn.empno)  
6  where sysdate between vcmn.start_date and vcmn.end_date;  
  
no rows selected
```

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);
```

EMPNO	WORKER	MGR	START_DATE	END_DATE	EMPNO	MANAGER	START_DATE	END_DATE
7566	JONES	7839	07-FEB-15	08-MAR-15	7839	KING	07-JAN-13	15-JAN-13
7698	BLAKE	7839	05-FEB-13	24-FEB-13	7839	KING	07-JAN-13	15-JAN-13
7782	CLARK	7839	03-JAN-15	16-MAR-15	7839	KING	07-JAN-13	15-JAN-13
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	7839	KING	07-JAN-13	15-JAN-13
8003	DR NO	7839	(null)	(null)	7839	KING	07-JAN-13	15-JAN-13
7566	JONES	7839	07-FEB-15	08-MAR-15	7839	KING	24-JAN-13	17-FEB-13
7698	BLAKE	7839	05-FEB-13	24-FEB-13	7839	KING	24-JAN-13	17-FEB-13
7782	CLARK	7839	03-JAN-15	16-MAR-15	7839	KING	24-JAN-13	17-FEB-13
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	7839	KING	24-JAN-13	17-FEB-13
8003	DR NO	7839	(null)	(null)	7839	KING	24-JAN-13	17-FEB-13
7499	ALLEN	7698	(null)	(null)	7698	BLAKE	05-FEB-13	24-FEB-13
7521	WARD	7698	(null)	(null)	7698	BLAKE	05-FEB-13	24-FEB-13
7654	MARTIN	7698	(null)	(null)	7698	BLAKE	05-FEB-13	24-FEB-13
7844	TURNER	7698	07-MAR-13	14-APR-13	7698	BLAKE	05-FEB-13	24-FEB-13
7844	TURNER	7698	25-FEB-15	16-MAR-15	7698	BLAKE	05-FEB-13	24-FEB-13
7900	JAMES	7698	01-MAR-15	20-MAR-15	7698	BLAKE	05-FEB-13	24-FEB-13
7566	JONES	7839	07-FEB-15	08-MAR-15	7839	KING	07-JAN-13	08-JAN-13
7698	BLAKE	7839	05-FEB-13	24-FEB-13	7839	KING	07-JAN-13	08-JAN-13
7782	CLARK	7839	03-JAN-15	16-MAR-15	7839	KING	07-JAN-13	08-JAN-13
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	7839	KING	07-JAN-13	08-JAN-13
8003	DR NO	7839	(null)	(null)	7839	KING	07-JAN-13	08-JAN-13
7934	MILLER	7782	16-FEB-13	19-FEB-13	7782	CLARK	03-JAN-15	16-MAR-15
7788	SCOTT	7566	(null)	(null)	7566	JONES	07-FEB-15	08-MAR-15
7902	FORD	7566	07-JUN-13	09-JUN-13	7566	JONES	07-FEB-15	08-MAR-15
8001	JET LI	8000	17-JAN-13	27-JAN-13	8000	JACKIE CHAN	19-JAN-15	05-APR-15
8002	BRUCE LEE	8000	09-JAN-14	10-JAN-14	8000	JACKIE CHAN	19-JAN-15	05-APR-15
8002	BRUCE LEE	8000	01-SEP-14	09-SEP-14	8000	JACKIE CHAN	19-JAN-15	05-APR-15
7369	SMITH	7902	(null)	(null)	7902	FORD	07-JUN-13	09-JUN-13
7839	KING	(null)	24-JAN-13	17-FEB-13	(null)	(null)	(null)	(null)
7839	KING	(null)	07-JAN-13	15-JAN-13	(null)	(null)	(null)	(null)
7839	KING	(null)	07-JAN-13	08-JAN-13	(null)	(null)	(null)	(null)
7876	ADAMS	7788	(null)	(null)	7788	SCOTT	(null)	(null)

32 rows selected.

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;
```

```

SQL> select dm.loc
  2  from emp wr
  3  left outer join emp mn on (wr.mgr = mn.empno)
  4  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  5  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  6  join dept dw on (wr.deptno = dw.deptno)
  7  join dept dm on (mn.deptno = dm.deptno)
  8  where sysdate between vcmn.start_date and vcmn.end_date;

no rows selected

```

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);

```

```

SQL> select wr.empno, wr.ename worker, wr.mgr, vcwr.start_date, vcwr.end_date, dw.loc,
  2      mn.empno, mn.ename manager, vcmn.start_date, vcmn.end_date , dm.loc
  3  from emp wr
  4  left outer join emp mn on (wr.mgr = mn.empno)
  5  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  6  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  7  join dept dw on (wr.deptno = dw.deptno)
  8  join dept dm on (mn.deptno = dm.deptno);

```

EMPNO	WORKER	MGR	START_DATE	END_DATE	LOC	EMPNO	MANAGER	START_DATE	END_DATE	LOC
7698	BLAKE	7839	05-FEB-13	24-FEB-13	CHICAGO	7839	KING	07-JAN-13	15-JAN-13	NEW_YORK
7782	CLARK	7839	03-JAN-15	16-MAR-15	NEW_YORK	7839	KING	07-JAN-13	15-JAN-13	NEW_YORK
7566	JONES	7839	07-FEB-15	08-MAR-15	DALLAS	7839	KING	07-JAN-13	15-JAN-13	NEW_YORK
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	HONKONG	7839	KING	07-JAN-13	15-JAN-13	NEW_YORK
7698	BLAKE	7839	05-FEB-13	24-FEB-13	CHICAGO	7839	KING	24-JAN-13	17-FEB-13	NEW_YORK
7782	CLARK	7839	03-JAN-15	16-MAR-15	NEW_YORK	7839	KING	24-JAN-13	17-FEB-13	NEW_YORK
7566	JONES	7839	07-FEB-15	08-MAR-15	DALLAS	7839	KING	24-JAN-13	17-FEB-13	NEW_YORK
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	HONKONG	7839	KING	24-JAN-13	17-FEB-13	NEW_YORK
7900	JAMES	7698	01-MAR-15	20-MAR-15	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7844	TURNER	7698	25-FEB-15	16-MAR-15	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7844	TURNER	7698	07-MAR-13	14-APR-13	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7499	ALLEN	7698	(null)	(null)	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7521	WARD	7698	(null)	(null)	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7654	MARTIN	7698	(null)	(null)	CHICAGO	7698	BLAKE	05-FEB-13	24-FEB-13	CHICAGO
7698	BLAKE	7839	05-FEB-13	24-FEB-13	CHICAGO	7839	KING	07-JAN-13	08-JAN-13	NEW_YORK
7782	CLARK	7839	03-JAN-15	16-MAR-15	NEW_YORK	7839	KING	07-JAN-13	08-JAN-13	NEW_YORK
7566	JONES	7839	07-FEB-15	08-MAR-15	DALLAS	7839	KING	07-JAN-13	08-JAN-13	NEW_YORK
8000	JACKIE CHAN	7839	19-JAN-15	05-APR-15	HONKONG	7839	KING	07-JAN-13	08-JAN-13	NEW_YORK
7934	MILLER	7782	16-FEB-13	19-FEB-13	NEW_YORK	7782	CLARK	03-JAN-15	16-MAR-15	NEW_YORK
7902	FORD	7566	07-JUN-13	09-JUN-13	DALLAS	7566	JONES	07-FEB-15	08-MAR-15	DALLAS
7788	SCOTT	7566	(null)	(null)	DALLAS	7566	JONES	07-FEB-15	08-MAR-15	DALLAS
8002	BRUCE LEE	8000	09-JAN-14	10-JAN-14	HONKONG	8000	JACKIE CHAN	19-JAN-15	05-APR-15	HONKONG
8002	BRUCE LEE	8000	01-SEP-14	09-SEP-14	HONKONG	8000	JACKIE CHAN	19-JAN-15	05-APR-15	HONKONG
8001	JET LI	8000	17-JAN-13	27-JAN-13	HONKONG	8000	JACKIE CHAN	19-JAN-15	05-APR-15	HONKONG
7369	SMITH	7902	(null)	(null)	DALLAS	7902	FORD	07-JUN-13	09-JUN-13	DALLAS
7876	ADAMS	7788	(null)	(null)	DALLAS	7788	SCOTT	(null)	(null)	DALLAS

26 rows selected.

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;
```

```
SQL> select wr.empno, wr.ename worker
  2  from emp wr
  3  left outer join emp mn on (wr.mgr = mn.empno)
  4  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  5  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  6  where sysdate between vcmn.start_date and vcmn.end_date;

no rows selected
```

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;
```

```
SQL> select mn.empno, mn.ename manager
  2  from emp wr
  3  left outer join emp mn on (wr.mgr = mn.empno)
  4  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  5  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  6  where sysdate between vcwr.start_date and vcwr.end_date;

no rows selected
```

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;
```

```

SQL> select wr.empno, wr.ename worker
  2  from emp wr
  3  left outer join emp mn on (wr.mgr = mn.empno)
  4  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  5  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  6  where sysdate > vcwr.start_date
  7  and extract(day from sysdate) > extract(day from vcwr.end_date)
  8  and extract(month from sysdate) = extract(month from vcwr.end_date)
  9  and extract(year from sysdate) = extract(year from vcwr.end_date)
10  and wr.comm is not null;

no rows selected

```

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;

```

```

SQL> select wr.empno, wr.ename worker, wr.comm, sysdate, extract(month from sysdate) mon_sys,
  2  vcwr.start_date, extract(month from vcwr.start_date) month_start,
  3  vcwr.end_date, extract(month from vcwr.end_date) month_end
  4  from emp wr
  5  left outer join emp mn on (wr.mgr = mn.empno)
  6  left outer join vacation vcwr on (wr.empno = vcwr.empno)
  7  left outer join vacation vcmn on (mn.empno = vcmn.empno)
  8  where wr.comm is not null;

      EMPNO WORKER          COMM SYSDATE      MON_SYS START_DATE MONTH_START END_DATE   MONTH_END
-----  -----  -----  -----  -----  -----  -----  -----  -----  -----
    8000 JACKIE CHAN        1700 09-APR-21       4 19-JAN-15      1 05-APR-15      4
    8000 JACKIE CHAN        1700 09-APR-21       4 19-JAN-15      1 05-APR-15      4
    7499 ALLEN             300 09-APR-21        4 (null) (null) (null) (null)
    7521 WARD              500 09-APR-21        4 (null) (null) (null) (null)
    7654 MARTIN            1400 09-APR-21       4 (null) (null) (null) (null)
    7844 TURNER            0 09-APR-21         4 07-MAR-13      3 14-APR-13      4
    7844 TURNER            0 09-APR-21         4 25-FEB-15      2 16-MAR-15      3
    8000 JACKIE CHAN        1700 09-APR-21       4 19-JAN-15      1 05-APR-15      4
    8001 JET LI             600 09-APR-21       4 17-JAN-13      1 27-JAN-13      1

9 rows selected.

```

7. Видаліть таблицю Vocation(vacation).

**drop table vacation;**

```

SQL> drop table vacation;

Table dropped.

```

## Лабораторна робота 9

## Операції над множинами

1. Створіть звіт, який містить номери відділів, в яких немає клерків.

```
select e1.deptno
```

```
from emp e1
```

```
where e1.deptno not in
```

```
(select e2.deptno
  from emp e2
  where e2.job = 'CLERK')
```

```
intersect
```

```
select ee.deptno
```

```
from emp ee;
```

```
SQL> select e1.deptno
  2  from emp e1
  3  where e1.deptno not in
  4                                (select e2.deptno
  5                                from emp e2
  6                                where e2.job = 'CLERK')
  7  intersect
  8  select ee.deptno
  9  from emp ee;
```

DEPTNO
60

Перевірка(відділи з клерками):

```
select deptno
```

```
from emp
```

```
where job = 'CLERK';
```

```
SQL> select deptno
  2  from emp
  3  where job = 'CLERK';
```

DEPTNO
30
20
20
10

2. Створіть звіт, який містить номери відділів, в яких немає співробітників.

```
select d.deptno  
from dept d  
minus  
select e.deptno  
from emp e;
```

```
SQL> select d.deptno  
  2  from dept d  
  3  minus  
  4  select e.deptno  
  5  from emp e;
```

DEPTNO
40
50
100

3. Створіть звіт, який містить номер відділу, посади, які є у відділах № 10, 30, і 20. Номери відділів повинні бути в зазначеному порядку (10, 30, 20)

```
select e1.job , e1.deptno  
from emp e1  
where e1.deptno = 10  
union all  
select e2.job, e2.deptno  
from emp e2  
where e2.deptno = 30  
union all  
select e3.job, e3.deptno  
from emp e3  
where e3.deptno = 20;
```

```
SQL> select e1.job , e1.deptno  
  2  from emp e1  
  3  where e1.deptno = 10  
  4  union all  
  5  select e2.job, e2.deptno  
  6  from emp e2  
  7  where e2.deptno = 30  
  8  union all  
  9  select e3.job, e3.deptno  
 10  from emp e3  
 11  where e3.deptno = 20;
```

JOB	DEPTNO
PRESIDENT	10
MANAGER	10
CLERK	10
MANAGER	30
SALESMAN	30
SALESMAN	30
SALESMAN	30
CLERK	30
SALESMAN	30
MANAGER	20
ANALYST	20
CLERK	20
ANALYST	20
CLERK	20

14 rows selected.

Перевірка:

```
select e1.ename, e1.job , e1.deptno  
from emp e1  
where e1.deptno = 10  
union all  
select e2.ename, e2.job, e2.deptno  
from emp e2  
where e2.deptno = 30  
union all  
select e3.ename, e3.job, e3.deptno  
from emp e3  
where e3.deptno = 20;
```

```
SQL> select e1.ename, e1.job , e1.deptno  
  2  from emp e1  
  3  where e1.deptno = 10  
  4  union all  
  5  select e2.ename, e2.job, e2.deptno  
  6  from emp e2  
  7  where e2.deptno = 30  
  8  union all  
  9  select e3.ename, e3.job, e3.deptno  
 10 from emp e3  
 11 where e3.deptno = 20;
```

ENAME	JOB	DEPTNO
KING	PRESIDENT	10
CLARK	MANAGER	10
MILLER	CLERK	10
BLAKE	MANAGER	30
MARTIN	SALESMAN	30
ALLEN	SALESMAN	30
TURNER	SALESMAN	30
JAMES	CLERK	30
WARD	SALESMAN	30
JONES	MANAGER	20
FORD	ANALYST	20
SMITH	CLERK	20
SCOTT	ANALYST	20
ADAMS	CLERK	20

14 rows selected.

4. Створіть звіт, який містить імена і номери всіх співробітників відділу №10 і всіх співробітників у яких зарплата вище \$ 1500. Відсортуйте дані за спаданням імен співробітників

```
select e1.empno, e1.ename  
from emp e1  
where e1.deptno = 10  
union  
select e2.empno, e2.ename  
from emp e2  
where e2.sal > 1500  
order by ename desc;
```

```
SQL> select e1.empno, e1.ename  
  2  from emp e1  
  3  where e1.deptno = 10  
  4  union  
  5  select e2.empno, e2.ename  
  6  from emp e2  
  7  where e2.sal > 1500  
  8  order by ename desc;
```

EMPNO	ENAME
7788	SCOTT
7934	MILLER
7839	KING
7566	JONES
8001	JET LI
8000	JACKIE CHAN
7902	FORD
8003	DR NO
7782	CLARK
8002	BRUCE LEE
7698	BLAKE
7499	ALLEN

12 rows selected.

Перевірка:

```
select e1.empno, e1.ename, e1.deptno, e1.sal
from emp e1
where e1.deptno = 10
union
select e2.empno, e2.ename, e2.deptno, e2.sal
from emp e2
where e2.sal > 1500
order by ename desc;
```

```
SQL> select e1.empno, e1.ename, e1.deptno, e1.sal
2  from emp e1
3  where e1.deptno = 10
4  union
5  select e2.empno, e2.ename, e2.deptno, e2.sal
6  from emp e2
7  where e2.sal > 1500
8  order by ename desc;
```

EMPNO	ENAME	DEPTNO	SAL
7788	SCOTT	20	3000
7934	MILLER	10	1300
7839	KING	10	5000
7566	JONES	20	2975
8001	JET LI	60	2600
8000	JACKIE CHAN	60	2250
7902	FORD	20	3000
8003	DR NO	(null)	2500
7782	CLARK	10	1500
8002	BRUCE LEE	60	2500
7698	BLAKE	30	2850
7499	ALLEN	30	1600

12 rows selected.

## 5. Складні вибірки

- Створіть представлення Emp\_USA, яке містить всіх співробітників, що працюють в США (Нью Йорку, Далласі, Бостоні, Чикаго)

```
create view Emp_USA
as select e.ename, e.deptno, d.loc
from emp e, dept d
where e.deptno = d.deptno
and d.loc in ('NEW_YORK', 'DALLAS', 'BOSTON', 'CHICAGO');
select * from Emp_USA;
```

```

SQL> create view Emp_USA
  2  as select e.ename, e.deptno, d.loc
  3  from emp e, dept d
  4  where e.deptno = d.deptno
  5  and d.loc in ('NEW_YORK', 'DALLAS', 'BOSTON', 'CHICAGO');

View created.

SQL> select * from Emp_USA;

ENAME          DEPTNO LOC
-----          -----
KING           10    NEW_YORK
MILLER         10    NEW_YORK
CLARK          10    NEW_YORK
ADAMS          20    DALLAS
SCOTT          20    DALLAS
SMITH          20    DALLAS
FORD           20    DALLAS
JONES          20    DALLAS
WARD           30    CHICAGO
JAMES          30    CHICAGO
TURNER         30    CHICAGO
ALLEN          30    CHICAGO
MARTIN         30    CHICAGO
BLAKE          30    CHICAGO

14 rows selected.

```

- Створіть представлення Emp\_SALES яке містить всіх співробітників, що працюють у відділах та містять в назві «SALE»

```

create view Emp_SALES
as select e.ename, e.deptno, d.dname, d.loc
from emp e, dept d
where e.deptno = d.deptno
and d.dname like 'SALE%';

select * from Emp_SALES;

```

```

SQL> create view Emp_SALES
  2  as select e.ename, e.deptno, d.dname, d.loc
  3  from emp e, dept d
  4  where e.deptno = d.deptno
  5  and d.dname like 'SALE%';

View created.

SQL> select * from Emp_SALES;

ENAME          DEPTNO DNAME        LOC
-----          -----
BLAKE          30    SALES       CHICAGO
MARTIN         30    SALES       CHICAGO
ALLEN          30    SALES       CHICAGO
TURNER         30    SALES       CHICAGO
JAMES          30    SALES       CHICAGO
WARD           30    SALES       CHICAGO
JACKIE CHAN   60    SALES2     HONKONG
JET LI          60    SALES2     HONKONG
BRUCE LEE      60    SALES2     HONKONG

9 rows selected.

```

➤ Складіть звіти, які містять:

1. Список, в якому не повторюються імена співробітників , що працюють в Emp\_USA або Emp\_SALES

```
select ename  
from Emp_USA  
union  
select ename  
from Emp_SALES;
```

```
SQL> select ename  
  2  from Emp_USA  
  3  union  
  4  select ename  
  5  from Emp_SALES;  
  
ENAME  
-----  
ADAMS  
ALLEN  
BLAKE  
BRUCE LEE  
CLARK  
FORD  
JACKIE CHAN  
JAMES  
JET LI  
JONES  
KING  
MARTIN  
MILLER  
SCOTT  
SMITH  
TURNER  
WARD  
  
17 rows selected.
```

2. Перелік всіх співробітників, які працюють в Emp\_USA або Emp\_SALES

```
select ename  
from Emp_USA  
union all  
select ename  
from Emp_SALES;
```

```
SQL> select ename  
  2  from Emp_USA  
  3  union all  
  4  select ename  
  5  from Emp_SALES;  
  
ENAME  
-----  
KING  
BLAKE  
BLAKE  
CLARK  
JONES  
MARTIN  
MARTIN  
ALLEN  
ALLEN  
TURNER  
TURNER  
JAMES  
JAMES  
WARD  
WARD  
FORD  
SMITH  
SCOTT  
ADAMS  
MILLER  
JACKIE CHAN  
JET LI  
BRUCE LEE  
  
23 rows selected.
```

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

```
select ename  
from Emp_USA  
intersect  
select ename  
from Emp_SALES;
```

```
SQL> select ename  
  2  from Emp_USA  
  3  intersect  
  4  select ename  
  5  from Emp_SALES;  
  
ENAME  
-----  
ALLEN  
BLAKE  
JAMES  
MARTIN  
TURNER  
WARD  
  
6 rows selected.
```

4. Перелік співробітників які працюють в Emp\_SALES але не в Emp\_USA.

```
select ename  
from Emp_SALES  
minus  
select ename  
from Emp_USA;
```

```
SQL> select ename  
  2  from Emp_SALES  
  3  minus  
  4  select ename  
  5  from Emp_USA;  
  
ENAME  
-----  
BRUCE LEE  
JACKIE CHAN  
JET LI
```

➤ Видаліть представлення.

```
drop view Emp_USA;  
drop view Emp_SALES;
```

```
SQL> drop view Emp_USA;  
View dropped.  
  
SQL> drop view Emp_SALES;  
View dropped.
```

## 6. Складні вибірки

➤ Складіть звіт, який відповідає всім наступним вимогам:

- ❖ Звіт містить ім'я співробітника і номер відділу для всіх співробітників, незалежно від того чи належать вони до якогось відділу (без Set-операцій);
- ❖ Звіт містить назив і номер відділу для всіх відділів, незалежно від того чи є у відділі хоча б один співробітник (без Set-операцій);
- ❖ Звіт повинен містити 2 стовпчика (ім'я співробітника і номер відділу або назва відділу і номер відділу) і формуватися одним запитом.

**select ename, deptno**

**from emp**

**where deptno in**

**(select e.deptno**

**from emp e**

**union**

**select d.deptno**

**from dept d)**

**or deptno is null;**

```
SQL> select ename, deptno
  2  from emp
  3  where deptno in
  4          (select e.deptno
  5           from emp e
  6           union
  7           select d.deptno
  8            from dept d)
  9  or deptno is null;
```

ENAME	DEPTNO
KING	10
BLAKE	30
CLARK	10
JONES	20
MARTIN	30
ALLEN	30
TURNER	30
JAMES	30
WARD	30
FORD	20
SMITH	20
SCOTT	20
ADAMS	20
MILLER	10
JACKIE CHAN	60
JET LI	60
BRUCE LEE	60
DR NO	(null)

18 rows selected.

```
SQL> select e.deptno
  2  from emp e
  3  union
  4  select d.deptno
  5  from dept d;
```

DEPTNO
10
20
30
40
50
60
100
(null)

8 rows selected.

Перевірка:

```
select e.deptno
  from emp e
  union
  select d.deptno
  from dept d;
```

## 7 \* Складні вибірки

Створіть запит, якій виведе імена, номер відділу і зарплатою 2-х співробітників з відділу №20, які йдуть на 2-му та 3-му місці за рівнем зарплати у цьому відділі.

Зверніть, увагу, що найбільшу зарплату (3000) в цьому відділі отримують дві людини. Отже вивести потрібно Adams і Joins.

Підказка:

Для того, щоб обмежити кількість стрічок що виводяться в підзапитах, можна використати умову: *where rownum<=3*.

```
select ename, deptno, sal
from emp
where deptno = 20
and sal >= any
    (select sal
     from emp
     where deptno = 20)
and rownum <= 4
minus
select ename, deptno, sal
from emp
where deptno = 20
and sal >= any
    (select sal
     from emp
     where deptno = 20)
and rownum <= 2;
```

```
SQL> select ename, deptno, sal
  2  from emp
  3  where deptno = 20
  4  and sal >= any
  5          (select sal
  6           from emp
  7           where deptno = 20)
  8  and rownum <= 4
  9  minus
10  select ename, deptno, sal
11  from emp
12  where deptno = 20
13  and sal >= any
14          (select sal
15           from emp
16           where deptno = 20)
17  and rownum <= 2;
```

ENAME	DEPTNO	SAL
ADAMS	20	1100
JONES	20	2975

Перевірка:

```
select ename, deptno, sal
from emp
where deptno = 20
and sal >= any
    (select sal
     from emp
     where deptno = 20);
```

```
SQL> select ename, deptno, sal
  2  from emp
  3  where deptno = 20
  4  and sal >= any
  5          (select sal
  6           from emp
  7           where deptno = 20);
```

ENAME	DEPTNO	SAL
FORD	20	3000
SCOTT	20	3000
JONES	20	2975
ADAMS	20	1100
SMITH	20	800

## 8\* Складні вибірки

Створіть запит, який виведе імена, номер відділу і зарплату двох співробітників з відділу №10 та двох з відділу №20 з найбільшими зарплатами.

Підказки:

- 1) Для того, щоб обмежити кількість рядків, що виводяться у підзапитах, можна скористатися умовою: where rownum<=2
- 2) Запити до співробітників відділів 10 і 20 можна об'єднати використовуючи Union
- 3) Для того, щоб мати можливість використовувати сортування в запитах всередині команди Union, їх потрібно обернути в Select.

```
select ename, deptno, sal
from
  (select *
   from emp
   where deptno = 10
   order by sal desc)
where rownum <= 2
union
select ename, deptno, sal
from
  (select *
   from emp
   where deptno = 20
   order by sal desc)
where rownum <= 2
order by deptno, sal desc;
```

```
SQL> select ename, deptno, sal
  2  from
  3    (select *
  4     from emp
  5     where deptno = 10
  6     order by sal desc)
  7  where rownum <= 2
  8  union
  9  select ename, deptno, sal
10  from
11    (select *
12     from emp
13     where deptno = 20
14     order by sal desc)
15  where rownum <= 2
16  order by deptno, sal desc;

ENAME          DEPTNO      SAL
-----  -----
KING            10        5000
CLARK           10        1500
FORD             20        3000
SCOTT            20        3000
```

Перевірка:

```
select ename, deptno, sal
from
  (select *
   from emp
   where deptno = 10
   order by sal desc)
union
```

```
select ename, deptno, sal
from
  (select *
   from emp
   where deptno = 20
   order by sal desc)
order by deptno, sal desc;
```

```

SQL> select ename, deptno, sal
  2  from
  3      (select *
  4       from emp
  5       where deptno = 10
  6       order by sal desc)
  7  union
  8  select ename, deptno, sal
  9  from
 10     (select *
 11      from emp
 12      where deptno = 20
 13      order by sal desc)
 14 order by deptno, sal desc;

ENAME          DEPTNO    SAL
-----  -----
KING            10      5000
CLARK           10      1500
MILLER          10      1300
FORD             20      3000
SCOTT            20      3000
JONES            20      2975
ADAMS            20      1100
SMITH            20       800

8 rows selected.

```

## 9\* Складні вибірки

У компанії вирішили влаштувати обмін досвідом між співробітниками. Для цього кожен день один із співробітників робить доповідь. Співробітники виступають в алфавітному порядку. Після того, як всі співробітники зробили доповідь, доповідачем стає перший співробітник. Створіть запит який повертає ім'я доповідача, якщо відомо ім'я попереднього доповідача.

```

select rpad('&my_ename', 20, ' ') as speaker, ename as next_speaker
from
  (select ename
   from
     (select ename
      from emp
      where ename > '&my_ename'
      order by ename)
   union all
   select ename
   from
     (select ename
      from emp
      order by ename)
   where rownum < 2)
where rownum < 2;

```

```

SQL> select rpad('&my_ename', 20, ' ') as speaker, ename as next_speaker
  2  from
  3      (select ename
  4       from
  5           (select ename
  6            from emp
  7            where ename > '&my_ename'
  8            order by ename)
  9      union all
 10      select ename
 11       from
 12          (select ename
 13            from emp
 14            order by ename)
 15      where rownum < 2)
 16 where rownum < 2;
Enter value for my_ename: KING
old   1: select rpad('&my_ename', 20, ' ') as speaker, ename as next_speaker
new   1: select rpad('KING', 20, ' ') as speaker, ename as next_speaker
Enter value for my_ename: KING
old   7:             where ename > '&my_ename'
new   7:             where ename > 'KING'

SPEAKER          NEXT_SPEAKER
-----
KING              MARTIN

```

Перевірка:

```

SQL> select ename
  2  from emp
  3  order by ename;

ENAME
-----
ADAMS
ALLEN
BLAKE
BRUCE LEE
CLARK
DR NO
FORD
JACKIE CHAN
JAMES
JET LI
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD

18 rows selected.

```

## Лабораторна робота 10

1. Підрахуйте:

1.1. Скільки в фірмі відділів?

```
select count(distinct deptno) department
from emp;
```

```
select count(deptno) department
from dept;
```

```
SQL> select count(distinct deptno) department
  2  from emp;

DEPARTMENT
-----
4

SQL> select count(deptno) department
  2  from dept;

DEPARTMENT
-----
7
```

1.2 Скільки в фірмі співробітників?

```
select count(empno) worker
from emp;
```

```
SQL> select count(empno) worker
  2  from emp;

WORKER
-----
18
```

1.3 У скількох відділах є співробітники?

```
select count(deptno) department
from emp;
```

```
SQL> select count(deptno) department
  2  from emp;

DEPARTMENT
-----
17
```

1.4 Скільки у фірми аналітиків?

```
select count(*) ANALYST
from emp
where job = 'ANALYST';
```

```
SQL> select count(*) ANALYST
  2  from emp
  3  where job = 'ANALYST';

ANALYST
-----
3
```

1.5 Скільки людей працює в Гонконзі?

```
select count(*) HONKONG
from emp e, dept d
where e.deptno = d.deptno
and loc = 'HONKONG';
```

```
SQL> select count(*) HONKONG
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  and loc = 'HONKONG';

HONKONG
-----
3
```

1.6 Скільки людей в фірмі не мають підлеглих

```
select count(*) no_employees  
from emp  
where empno not in  
      (select distinct e1.empno  
       from emp e1, emp e2  
       where e1.empno = e2.mgr);
```

Перевірка:

```
select count(distinct e1.empno) employee  
from emp e1, emp e2  
where e1.empno = e2.mgr;
```

```
SQL> select count(*) no_employees  
  2  from emp  
  3  where empno not in  
  4          (select distinct e1.empno  
  5          from emp e1, emp e2  
  6          where e1.empno = e2.mgr);  
  
NO_EMPLOYEES  
-----  
           11
```

Завдання

2. Сформувати запит, що виводить мінімальну, середню, максимальну і сумарну зарплату всіх працівників.

```
select min(sal) min_sal, avg(sal) avg_sal, max(sal) max_sal, sum(sal) sum_sal  
from emp;
```

```
SQL> select min(sal) min_sal, avg(sal) avg_sal, max(sal) max_sal, sum(sal) sum_sal  
  2  from emp;  
  
MIN_SAL     AVG_SAL     MAX_SAL     SUM_SAL  
-----  
    800  2106.94444      5000      37925
```

3. Сформувати запит, що виводить номера і імена всіх співробітників, зарплата яких вища за середню.

```
select empno, ename  
from emp  
where sal >  
      (select avg(sal)  
       from emp);
```

```
SQL> select empno, ename  
  2  from emp  
  3  where sal >  
  4          (select avg(sal)  
  5          from emp);  
  
EMPNO ENAME  
-----  
    7839 KING  
    7698 BLAKE  
    7566 JONES  
    7902 FORD  
    7788 SCOTT  
    8000 JACKIE CHAN  
    8001 JET LI  
    8002 BRUCE LEE  
    8003 DR NO  
  
9 rows selected.
```

Перевірка:

```
select avg(sal)
from emp;
```

```
select empno, ename, sal
from emp
where sal >
      (select avg(sal)
       from emp);
```

```
SQL> select avg(sal)
  2  from emp;

          AVG(SAL)
-----
2106.94444

SQL> select empno, ename, sal
  2  from emp
  3  where sal >
  4      (select avg(sal)
  5    from emp);

      EMPNO ENAME          SAL
----- 7839 KING           5000
        7698 BLAKE          2850
        7566 JONES          2975
        7902 FORD            3000
        7788 SCOTT           3000
        8000 JACKIE CHAN     2250
        8001 JET LI           2600
        8002 BRUCE LEE        2500
        8003 DR NO            2500

9 rows selected.
```

4. Сформувати запит, що виводить ім'я співробітника з найбільшою премією.

```
select ename, comm
from emp
where comm =
      (select max(comm)
       from emp);
```

```
SQL> select ename, comm
  2  from emp
  3  where comm =
  4      (select max(comm)
  5    from emp);

      ENAME          COMM
----- JACKIE CHAN     1700
```

```
SQL> select ename, comm
  2  from emp;

      ENAME          COMM
----- KING          (null)
        BLAKE         (null)
        CLARK         (null)
        JONES         (null)
        MARTIN        1400
        ALLEN         300
        TURNER        0
        JAMES         (null)
        WARD          500
        FORD          (null)
        SMITH         (null)
        SCOTT         (null)
        ADAMS         (null)
        MILLER        (null)
        JACKIE CHAN   1700
        JET LI         600
        BRUCE LEE      (null)
        DR NO          (null)

18 rows selected.
```

Перевірка:

```
select ename, comm
from emp;
```

## Завдання

5. Сформуйте запит, що виводить імена керівників і суму зарплат його підлеглих.

```
select e1.ename employee, sum(e2.sal) worker_sum_sal
from emp e1, emp e2
where e1.empno = e2.mgr
group by e1.ename;
```

```
SQL> select e1.ename employee, sum(e2.sal) worker_sum_sal
  2  from emp e1, emp e2
  3  where e1.empno = e2.mgr
  4  group by e1.ename;

EMPLOYEE          WORKER_SUM_SAL
-----              -----
JONES                  6000
FORD                   800
KING                 12075
BLAKE                  6550
JACKIE CHAN            5100
SCOTT                  1100
CLARK                  1300

7 rows selected.
```

Перевірка:

```
select e2.empno, e2.ename worker, e2.mgr, e2.sal,
       e1.empno, e1.ename employee
  from emp e1, emp e2
 where e1.empno           =           e2.mgr;
      
```

```
SQL> select e2.empno, e2.ename worker, e2.mgr, e2.sal,
  2      e1.empno, e1.ename employee
  3  from emp e1, emp e2
  4  where e1.empno = e2.mgr;

EMPNO WORKER          MGR      SAL      EMPNO EMPLOYEE
-----  -----          ---  -----  -----  -----
 7902 FORD             7566    3000    7566 JONES
 7788 SCOTT            7566    3000    7566 JONES
 7499 ALLEN            7698    1600    7698 BLAKE
 7521 WARD              7698    1250    7698 BLAKE
 7654 MARTIN            7698    1250    7698 BLAKE
 7844 TURNER            7698    1500    7698 BLAKE
 7900 JAMES              7698     950    7698 BLAKE
 7934 MILLER            7782    1300    7782 CLARK
 7876 ADAMS              7788    1100    7788 SCOTT
 7566 JONES             7839    2975    7839 KING
 8003 DR NO              7839    2500    7839 KING
 7698 BLAKE              7839    2850    7839 KING
 7782 CLARK              7839    1500    7839 KING
 8000 JACKIE CHAN        7839    2250    7839 KING
 7369 SMITH              7902     800    7902 FORD
 8001 JET LI              8000    2600    8000 JACKIE CHAN
 8002 BRUCE LEE           8000    2500    8000 JACKIE CHAN

17 rows selected.
```

6. Сформуйте запит, що виводить імена керівників і середню зарплату його підлеглих за винятком підлеглих SALESMAN-ів.

```
select e1.ename employee, avg(e2.sal) worker_avg_sal
from emp e1, emp e2
where e1.empno = e2.mgr
and e2.job not like 'SALESMAN'
group by e1.ename;
```

```
SQL> select e1.ename employee, avg(e2.sal) worker_avg_sal
  2  from emp e1, emp e2
  3  where e1.empno = e2.mgr
  4  and e2.job not like 'SALESMAN'
  5  group by e1.ename;

EMPLOYEE          WORKER_AVG_SAL
-----            -----
JONES                  3000
FORD                   800
KING                 2456.25
BLAKE                  950
SCOTT                  1100
CLARK                  1300

6 rows selected.
```

7. Сформуйте запит, що виводить імена керівників, у яких зарплата більше ніж максимальна у його підлеглих.

```
select e1.ename employee
from emp e1
where e1.sal >
      (select max(e2.sal)
       from emp e2
       where e1.empno = e2.mgr);
```

```
SQL> select e1.ename employee
  2  from emp e1
  3  where e1.sal >
  4      (select max(e2.sal)
  5       from emp e2
  6       where e1.empno = e2.mgr);

EMPLOYEE
-----
BLAKE
CLARK
SCOTT
KING
FORD
```

## Завдання

8. \* В якому відділі максимальна середня зарплата?

```
select d.deptno, dname  
from dept d  
where (select avg(sal)  
      from emp e  
     where e.deptno = d.deptno) =  
          (select max(avg(sal))  
           from emp  
          group by deptno);
```

```
SQL> select d.deptno, dname  
  2  from dept d  
  3  where (select avg(sal)  
  4    from emp e  
  5    where e.deptno = d.deptno) =  
  6          (select max(avg(sal))  
  7           from emp  
  8           group by deptno);
```

DEPTNO	DNAME
10	ACCOUNTING

Перевірка:

```
select d.deptno, dname, avg(sal)  
from emp e, dept d  
where e.deptno = d.deptno  
group by d.deptno, dname;
```

```
SQL> select d.deptno, dname, avg(sal)  
  2  from emp e, dept d  
  3  where e.deptno = d.deptno  
  4  group by d.deptno, dname;
```

DEPTNO	DNAME	AVG(SAL)
10	ACCOUNTING	2600
20	RESEARCH	2175
30	SALES	1566.66667
60	SALES2	2450

9. \* У якому місті найбільше співробітників?

```
select d.loc  
from dept d  
where (select count(empno)  
      from emp e  
     where e.deptno = d.deptno)  
          =  
          (select max(count(empno))  
           from emp  
          group by deptno, loc);
```

```

SQL> select d.loc
  2  from dept d
  3  where (select count(empno)
  4      from emp e
  5      where e.deptno = d.deptno
  6      group by loc)           =
  7                                (select max(count(empno))
  8      from emp
  9      group by deptno, loc);

LOC
-----
CHICAGO

```

Перевірка:

```

select loc, count(empno)
from emp e, dept d
where e.deptno = d.deptno
group by loc;

```

```

SQL> select loc, count(empno)
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  group by loc;

LOC          COUNT(EMPNO)
-----
NEW_YORK            3
DALLAS              5
CHICAGO             6
HONKONG             3

```

10. Виведіть назви відділів в порядку зростання сум зарплат співробітників, що працюють у них.

```

select dname
from emp e, dept d
where e.deptno = d.deptno
group by dname
order by sum(sal);

```

Перевірка:

```

select d.deptno, dname, sum(sal)
from emp e, dept d
where e.deptno = d.deptno
group by d.deptno, dname
order by sum(sal);

```

```

SQL> select dname
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  group by dname
  5  order by sum(sal);

DNAME
-----
SALES2
ACCOUNTING
SALES
RESEARCH

```

```

SQL> select d.deptno, dname, sum(sal)
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  group by d.deptno, dname
  5  order by sum(sal);

```

DEPTNO	DNAME	SUM(SAL)
60	SALES2	7350
10	ACCOUNTING	7800
30	SALES	9400
20	RESEARCH	10875

11. \* В якому відділі найбільша кількість співробітників, що отримують премії?

```

select deptno
from emp
where comm is not null
group by deptno
having count(empno) =
(select max(count(empno))
from emp
where comm is not null
group by deptno);

```

```

SQL> select deptno
  2  from emp
  3  where comm is not null
  4  group by deptno
  5  having count(empno) =
  6          (select max(count(empno))
  7           from emp
  8           where comm is not null
  9           group by deptno);

DEPTNO
-----
```

30

Перевірка:

```

select deptno, count(empno) count_worker_with_comm
from emp
where comm is not null
group by deptno;

```

```

SQL> select deptno, count(empno) count_worker_with_comm
  2  from emp
  3  where comm is not null
  4  group by deptno;

DEPTNO COUNT_WORKER_WITH_COMM
-----
```

60	2
30	4

А в якому на премії йде найбільше грошей?

```

select deptno
from emp
where comm =
(select max(comm)
from emp
where comm is not null);

```

```

SQL> select deptno
  2  from emp
  3  where comm =
  4          (select max(comm)
  5           from emp
  6           where comm is not null);

DEPTNO
-----
```

60

Перевірка:

```
select deptno, comm  
from emp  
where comm is not null;
```

```
SQL> select deptno, comm  
2  from emp  
3  where comm is not null;  
  
DEPTNO      COMM  
-----  
      30      1400  
      30      300  
      30          0  
      30      500  
      60     1700  
      60      600  
  
6 rows selected.
```

## Частина 2. Ранжування

12. Виконайте ранжування відділів за кількістю співробітників.

```
select deptno, count_worker,  
rank() over (order by count_worker) count_rank  
from  
(select deptno, count(empno) count_worker  
from emp  
group by deptno);
```

```
SQL> select deptno, count_worker,  
2  rank() over (order by count_worker) count_rank  
3  from  
4    (select deptno, count(empno) count_worker  
5      from emp  
6      group by deptno);  
  
DEPTNO COUNT_WORKER COUNT_RANK  
-----  
(null)          1          1  
      60          3          2  
      10          3          2  
      20          5          4  
      30          6          5
```

13. Створіть звіт, в якому співробітники будуть пронумеровані і впорядковані за зростанням зарплати всередині відділів.

```
select deptno, ename, sal,  
rank() over (partition by deptno order by sal) number_rank  
from emp;
```

```

SQL> select deptno, ename, sal,
  2  rank() over (partition by deptno order by sal) number_rank
  3  from emp;

  DEPTNO ENAME          SAL NUMBER_RANK
----- -----
    10 MILLER        1300      1
    10 CLARK         1500      2
    10 KING          5000      3
    20 SMITH         800       1
    20 ADAMS        1100      2
    20 JONES         2975      3
    20 SCOTT         3000      4
    20 FORD          3000      4
    30 JAMES         950       1
    30 WARD          1250      2
    30 MARTIN        1250      2
    30 TURNER        1500      4
    30 ALLEN         1600      5
    30 BLAKE         2850      6
    60 JACKIE CHAN   2250      1
    60 BRUCE LEE     2500      2
    60 JET LI         2600      3
  (null)  DR NO       2500      1

18 rows selected.

```

14. \* Розставте співробітників з Нью-Йорка по заробітку (зарплата + премія)

```

select deptno, loc, ename, income,
rank() over (order by income) number_rank
from
(select d.deptno, loc, ename, sal + nvl(comm, 0) income
from dept d, emp e
where e.deptno = d.deptno
and loc = 'NEW_YORK');

```

```

SQL> select deptno, loc, ename, income,
  2  rank() over (order by income) number_rank
  3  from
  4    (select d.deptno, loc, ename, sal + nvl(comm, 0) income
  5    from dept d, emp e
  6    where e.deptno = d.deptno
  7    and loc = 'NEW_YORK');

  DEPTNO LOC          ENAME          INCOME NUMBER_RANK
----- -----
    10 NEW_YORK      MILLER        1300      1
    10 NEW_YORK      CLARK         1500      2
    10 NEW_YORK      KING          5000      3

```

DEPTNO	LOC	ENAME	INCOME	NUMBER_RANK
10	NEW_YORK	MILLER	1300	1
10	NEW_YORK	CLARK	1500	2
10	NEW_YORK	KING	5000	3

15. \* Виведіть ім'я, відділ і зарплату для співробітників, які отримують найменшу зарплату у відділі.

```
select ename, deptno, sal min_sal,
rank() over (order by sal) number_rank
from emp
where sal =
      (select min(sal)
       from emp e
       where e.deptno = emp.deptno);
```

```
SQL> select ename, deptno, sal min_sal,
  2  rank() over (order by sal) number_rank
  3  from emp
  4  where sal =
  5      (select min(sal)
  6       from emp e
  7       where e.deptno = emp.deptno);

    ENAME        DEPTNO    MIN_SAL NUMBER_RANK
-----  -----  -----
SMITH            20      800          1
JAMES            30      950          2
MILLER           10     1300          3
JACKIE CHAN      60     2250          4
```

Зведені таблиці

16. Складіть звіт який відображає середню зарплату по кожній з посад відділу.

```
select *
from
      (select d.deptno, dname, sal, job
       from emp e, dept d
       where e.deptno = d.deptno)
pivot
      (avg(sal) for job
       in ('PRESIDENT', 'MANAGER', 'SALESMAN', 'CLERK', 'ANALYST')) pvt
order by deptno;
```

```
SQL> select *
  2  from
  3      (select d.deptno, dname, sal, job
  4       from emp e, dept d
  5       where e.deptno = d.deptno)
  6  pivot
  7  (avg(sal) for job
  8  in ('PRESIDENT', 'MANAGER', 'SALESMAN', 'CLERK', 'ANALYST')) pvt
  9  order by deptno;

    DEPTNO DNAME        'PRESIDENT'  'MANAGER'  'SALESMAN'  'CLERK'  'ANALYST'
-----  -----  -----
          10 ACCOUNTING      5000      1500 (null)      1300 (null)
          20 RESEARCH      (null)      2975 (null)      950      3000
          30 SALES        (null)      2850      1400      950 (null)
          60 SALES2        (null)      (null)      2450 (null)      (null)
```

Перевірка:

**select ename, job, sal, deptno from emp;**

ENAME	JOB	SAL	DEPTNO
KING	PRESIDENT	5000	10
BLAKE	MANAGER	2850	30
CLARK	MANAGER	1500	10
JONES	MANAGER	2975	20
MARTIN	SALESMAN	1250	30
ALLEN	SALESMAN	1600	30
TURNER	SALESMAN	1500	30
JAMES	CLERK	950	30
WARD	SALESMAN	1250	30
FORD	ANALYST	3000	20
SMITH	CLERK	800	20
SCOTT	ANALYST	3000	20
ADAMS	CLERK	1100	20
MILLER	CLERK	1300	10
JACKIE CHAN	SALESMAN	2250	60
JET LI	SALESMAN	2600	60
BRUCE LEE	SALESMAN	2500	60
DR NO	ANALYST	2500	(null)

18 rows selected.

17. Складіть звіт який містить зарплати співробітників з підсумковими рядками, які відображають суму зарплат групувати по «відділ і посаду» і по «відділі». (Стовпці звіту: відділ, посада, сумарна зарплата).

**select nvl(deptno, 0) deptno, nvl(job, '--- SUMA -->') job, sum(sal) sum\_sal  
from emp  
group by ROLLUP(deptno, job);**

DEPTNO	JOB	SUM_SAL
0	ANALYST	2500
0	--- SUMA -->	2500
10	CLERK	1300
10	MANAGER	1500
10	PRESIDENT	5000
10	--- SUMA -->	7800
20	CLERK	1900
20	ANALYST	6000
20	MANAGER	2975
20	--- SUMA -->	10875
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600
30	--- SUMA -->	9400
60	SALESMAN	7350
60	--- SUMA -->	7350
0	--- SUMA -->	37925

17 rows selected.

18. Складіть звіт який містить зарплати співробітників з підсумковими рядками, які відображають суми зарплат, що згруповані по «місто, відділ, посада», «відділ і посада», «відділ».

```
select d.deptno, nvl(loc, '- - - >') loc, nvl(job, '- - - >') job, sum(sal) sum_sal
from emp e, dept d
where e.deptno = d.deptno
group by
grouping sets ((loc, d.deptno, job), (d.deptno, job), (d.deptno));
```

```
SQL> select d.deptno, nvl(loc, '- - - >') loc, nvl(job, '- - - >') job, sum(sal) sum_sal
2  from emp e, dept d
3  where e.deptno = d.deptno
4  group by
5  grouping sets ((loc, d.deptno, job), (d.deptno, job), (d.deptno));

DEPTNO LOC          JOB           SUM_SAL
----- --          --          -----
      10 NEW_YORK    CLERK        1300
      10 - - -       > CLERK      1300
      10 NEW_YORK    MANAGER     1500
      10 - - -       > MANAGER   1500
      10 NEW_YORK    PRESIDENT  5000
      10 - - -       > PRESIDENT 5000
      10 - - -       > - - - - > 7800
      20 DALLAS      CLERK        1900
      20 - - -       > CLERK      1900
      20 DALLAS      ANALYST     6000
      20 - - -       > ANALYST   6000
      20 DALLAS      MANAGER     2975
      20 - - -       > MANAGER   2975
      20 - - -       > - - - - > 10875
      30 CHICAGO     CLERK        950
      30 - - -       > CLERK      950
      30 CHICAGO     MANAGER     2850
      30 - - -       > MANAGER   2850
      30 CHICAGO     SALESMAN   5600
      30 - - -       > SALESMAN  5600
      30 - - -       > - - - - > 9400
      60 HONKONG     SALESMAN   7350
      60 - - -       > SALESMAN  7350
      60 - - -       > - - - - > 7350

24 rows selected.
```

19. \* Складіть звіт який містить зарплати співробітників з підсумковими рядками, які відображають суми зарплат, що згруповані по «місто, відділ, посада», «відділ і посада», «місто, посада», «місто, відділ», «місто», «відділ» , «посада».

```
select d.deptno, nvl(loc, '- - - >') loc, nvl(job, '- - - >') job, sum(sal) sum_sal
from emp e, dept d
where e.deptno = d.deptno
group by
grouping sets ((loc, d.deptno, job), (d.deptno, job), (loc, job),
                (loc, d.deptno), (loc), (d.deptno), (job));
```

```

SQL> select d.deptno, nvl(loc, ' - - - >') loc, nvl(job, ' - - - >') job, sum(sal) sum_sal
  2 from emp e, dept d
  3 where e.deptno = d.deptno
  4 group by
  5 grouping sets ((loc, d.deptno, job), (d.deptno, job), (loc, job),
  6                  (loc, d.deptno), (loc), (d.deptno), (job));

```

DEPTNO	LOC	JOB	SUM_SAL
10	NEW_YORK	CLERK	1300
20	DALLAS	CLERK	1900
30	CHICAGO	CLERK	950
20	DALLAS	ANALYST	6000
10	NEW_YORK	MANAGER	1500
20	DALLAS	MANAGER	2975
30	CHICAGO	MANAGER	2850
30	CHICAGO	SALESMAN	5600
60	HONKONG	SALESMAN	7350
10	NEW_YORK	PRESIDENT	5000
10	- - - >	CLERK	1300
20	- - - >	CLERK	1900
30	- - - >	CLERK	950
(null)	- - - >	CLERK	4150
(null)	- - - >	ANALYST	6000
(null)	- - - >	ANALYST	6000
10	- - - >	MANAGER	1500
20	- - - >	MANAGER	2975
30	- - - >	MANAGER	2850
(null)	- - - >	MANAGER	7325
30	- - - >	SALESMAN	5600
60	- - - >	SALESMAN	7350
(null)	- - - >	SALESMAN	12950
10	- - - >	PRESIDENT	5000
(null)	- - - >	PRESIDENT	5000
10	NEW_YORK	- - - >	7800
10	- - - >	- - - >	7800
20	DALLAS	- - - >	10875
20	- - - >	- - - >	10875
30	CHICAGO	- - - >	9400
30	- - - >	- - - >	9400
60	HONKONG	- - - >	7350
60	- - - >	- - - >	7350
(null)	DALLAS	CLERK	1900
(null)	DALLAS	ANALYST	6000
(null)	DALLAS	MANAGER	2975
(null)	DALLAS	- - - >	10875

(null)	DALLAS	- - - >	10875
DEPTNO	LOC	JOB	SUM_SAL
(null)	CHICAGO	CLERK	950
(null)	CHICAGO	MANAGER	2850
(null)	CHICAGO	SALESMAN	5600
(null)	CHICAGO	- - - >	9400
(null)	HONKONG	SALESMAN	7350
(null)	HONKONG	- - - >	7350
(null)	NEW_YORK	CLERK	1300
(null)	NEW_YORK	MANAGER	1500
(null)	NEW_YORK	PRESIDENT	5000
(null)	NEW_YORK	- - - >	7800

47 rows selected.

20. \* Складіть звіт, який містить тільки підсумкові рядки, які відображають суми зарплат, що згруповані по «місто, відділ, посада», «відділ, посада», «місто».

```

select sum(sal) sum_sal
from emp e, dept d
where e.deptno = d.deptno
group by
grouping sets ((loc, d.deptno, job), (d.deptno, job), (loc));

```

```

SQL> select sum(sal) sum_sal
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  group by
  5  grouping sets ((loc, d.deptno, job), (d.deptno, job), (loc));

```

SUM\_SAL

```

-----
1300
1500
5000
1900
6000
2975
950
2850
5600
7350
1300
1500
5000
1900
6000
2975
950
2850
5600
7350
7800
10875
9400
7350

```

24 rows selected.

## 21 \* «Понти дорожчі за гроші»

- У податковій є таблиця виду:

Власник	Предмети розкоші	Вартість
Олігарх Іванов	Яхта	100.000
Олігарх Іванов	Яхта	200.000
Олігарх Іванов	Дача	1.000.000
Олігарх Петров	Машина	200.000

- Види предметів розкоші - Яхта, дача, машина.
- Виведіть інформацію у вигляді:

	Кількість яхт	Вартість яхт	Кількість дач	Вартість дач	...
Олігарх Іванов	2	300.000	1	1.000.000	
...					

```

create table oligarchs_property(owner varchar(20) not null,
                                luxury_items varchar(20) not null,
                                cost int not null);

insert into oligarchs_property values('The oligarch Ivanov', 'Yacht', 100000);
insert into oligarchs_property values('The oligarch Ivanov', 'Yacht', 200000);
insert into oligarchs_property values('The oligarch Ivanov', 'Country house',
1000000);
insert into oligarchs_property values('The oligarch Petrov', 'Car', 200000);

select owner, luxury_items, to_char(cost, '9,999,999') cost
from oligarchs_property;

```

```

SQL> create table oligarchs_property(owner varchar(20) not null,
2                                     luxury_items varchar(20) not null,
3                                     cost int not null);

Table created.

SQL> insert into oligarchs_property values('The oligarch Ivanov', 'Yacht', 100000);

1 row created.

SQL> insert into oligarchs_property values('The oligarch Ivanov', 'Yacht', 200000);

1 row created.

SQL> insert into oligarchs_property values('The oligarch Ivanov', 'Country house', 1000000);

1 row created.

SQL> insert into oligarchs_property values('The oligarch Petrov', 'Car', 200000);

1 row created.

SQL> select owner, luxury_items, to_char(cost, '9,999,999') cost
  2  from oligarchs_property;

OWNER          LUXURY_ITEMS      COST
-----          -----
The oligarch Ivanov Yacht        100,000
The oligarch Ivanov Yacht        200,000
The oligarch Ivanov Country house 1,000,000
The oligarch Petrov   Car         200,000

```

Вартість предметів розкоші:

```

select *
from
(select owner, luxury_items
 from oligarchs_property)
pivot
(count(luxury_items) for luxury_items
in ('Yacht', 'Country house', 'Car')) pvt;

```

```

SQL> select *
  2  from
  3      (select owner, luxury_items
  4       from oligarchs_property)
  5 pivot
  6 (count(luxury_items) for luxury_items
  7 in ('Yacht', 'Country house', 'Car')) pvt;

```

OWNER	'Yacht'	'Country house'	'Car'
The oligarch Ivanov	2	1	0
The oligarch Petrov	0	0	1

```

select *
from
  (select owner, luxury_items, cost
   from oligarchs_property)
pivot
(sum(cost) for luxury_items
in ('Yacht', 'Country house', 'Car')) pvt;

```

```

SQL> select *
  2  from
  3      (select owner, luxury_items, cost
  4       from oligarchs_property)
  5 pivot
  6 (sum(cost) for luxury_items
  7 in ('Yacht', 'Country house', 'Car')) pvt;

```

OWNER	'Yacht'	'Country house'	'Car'
The oligarch Ivanov	300000	1000000	(null)
The oligarch Petrov	(null)	(null)	200000

**drop table oligarchs\_property;**

```

SQL> drop table oligarchs_property;
Table dropped.

```

## Лабораторна робота 11

## 1. Задания

Напишіть запит, який би повертає з таблиці emp інформацію:

- про ім'я співробітника;
- про рівні підпорядкованості (найвищий рівень - головний начальник, який нікому не підпорядковується - рівень 0);
- про шляхи підзвітності в форматі:  
/ Керівник 1 / керівник 2 / рядовий співробітник
- Звіт повинен бути відсортований за рівнем підпорядкованості.

```
select ename, level - 1 "LEVEL", SYS_CONNECT_BY_PATH(ename, '/') "PATH"
"PATH"
from emp
start with mgr is null
connect by nocycle prior empno = mgr
order by level;
```

SQL> select ename, level - 1 "LEVEL", SYS_CONNECT_BY_PATH(ename, '/') "PATH"	
2 from emp	
3 start with mgr is null	
4 connect by nocycle prior empno = mgr	
5 order by level;	
ENAME	LEVEL
-----	-----
PATH	
-----	-----
KING	0
/KING	
CLARK	1
/KING/CLARK	
JONES	1
/KING/JONES	
JACKIE CHAN	1
/KING/JACKIE CHAN	
DR NO	1
/KING/DR NO	
BLAKE	1
/KING/BLAKE	
JAMES	2
/KING/BLAKE/JAMES	
TURNER	2
/KING/BLAKE/TURNER	
MARTIN	2
/KING/BLAKE/MARTIN	

MARTIN	2
/KING/BLAKE/MARTIN	
ALLEN	2
/KING/BLAKE/ALLEN	
FORD	2
/KING/JONES/FORD	
SCOTT	2
/KING/JONES/SCOTT	
ENAME	LEVEL
-----	
PATH	
-----	
WARD	2
/KING/BLAKE/WARD	
BRUCE LEE	2
/KING/JACKIE CHAN/BRUCE LEE	
MILLER	2
/KING/CLARK/MILLER	
JET LI	2
/KING/JACKIE CHAN/JET LI	
ADAMS	3
/KING/JONES/SCOTT/ADAMS	
SMITH	3
/KING/JONES/FORD/SMITH	
18 rows selected.	

## 2. Начальники

Виберіть всі імена начальників співробітника на ім'я «JET LI»

```
select ename
from emp
where ename != 'JET LI'
start with ename = 'JET LI'
connect by nocycle prior mgr = empno;
```

SQL> select ename
2 from emp
3 where ename != 'JET LI'
4 start with ename = 'JET LI'
5 connect by nocycle prior mgr = empno;
ENAME
-----
JACKIE CHAN
KING

### 3\*. Пропуски

1. Створіть таблицю A
2. Видаліть з таблиці рядки 4 і 6.
3. Напишіть запит, який виведе віддалені числа.
4. \* Як буде виглядати запит, якщо максимальний елемент наперед не відомий
5. \* Як буде виглядати запит якщо максимальний і мінімальний елемент наперед невідомі?
6. Видаліть таблицю A.

Підказка: числову послідовність можна згенерувати запитом

A
1
2
3
4
5
6
7

Select rownum From dual Connect by level <= 7

1. Створіть таблицю A

```
create table my_A  
as (select rownum A  
     from dual  
    connect by level <= 7);  
  
select * from my_A;
```

```
SQL> create table my_A  
  2  as (select rownum A  
  3      from dual  
  4      connect by level <= 7);  
  
Table created.  
  
SQL> select * from my_A;  
  
          A  
-----  
        1  
        2  
        3  
        4  
        5  
        6  
        7  
  
7 rows selected.
```

2. Видаліть з таблиці рядки 4 і 6.

```
delete from my_A  
where A in (4, 6);  
  
select * from my_A;
```

```
SQL> delete from my_A  
  2  where A in (4, 6);  
  
2 rows deleted.  
  
SQL> select * from my_A;  
  
          A  
-----  
        1  
        2  
        3  
        5  
        7
```

**3. Напишіть запит, який виведе віддалені числа.**

```
select * from my_A  
as of timestamp(systimestamp - interval '1' minute)  
minus  
select * from my_A;
```

```
SQL> select * from my_A  
2 as of timestamp(systimestamp - interval '1' minute)  
3 minus  
4 select * from my_A;  
  
          A  
-----  
        4  
        6
```

**4. \*** Як буде виглядати запит, якщо максимальний елемент наперед не відомий

```
select rownum A from dual  
connect by level <=  
      (select max(A) from my_A)  
minus  
select * from my_A;
```

```
SQL> select rownum A from dual  
2 connect by level <=  
3           (select max(A) from my_A)  
4 minus  
5 select * from my_A;  
  
          A  
-----  
        4  
        6
```

**5. \*** Як буде виглядати запит якщо максимальний і мінімальний елемент наперед невідомі?

```
select rownum +  
      (select min(A) from my_A) - 1 A  
from dual  
connect by level <=  
      (select max(A) - min(A) from my_A)  
minus  
select * from my_A;
```

```

SQL> select rownum +
      2          (select min(A) from my_A) - 1 A
      3  from dual
      4 connect by level <=
      5          (select max(A) - min(A) from my_A)
      6 minus
      7 select * from my_A;

      A
-----
      4
      6

```

6. Видаліть таблицю А.

**drop table my\_A;**

```

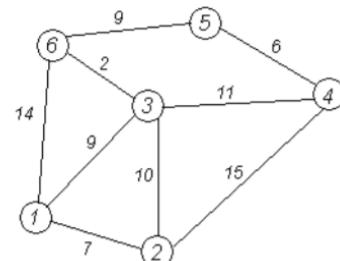
SQL> drop table my_A;
Table dropped.

```

4\*. Втекти звідси

Створіть таблицю відстаней між містами (не менше 8 рядків), наприклад: або так:

Точка А	Точка Б	Відстань
Чернігів	Вінниця	423
Чернігів	Дніпро	585
Дніпро	Чернівці	891
Чернівці	Вінниця	312



```

create table my_table(point_A varchar(13),
                      point_B varchar(13),
                      distance integer);

```

```

insert into my_table values ('Chernihiv', 'Vinnytsia', 423);
insert into my_table values ('Chernihiv', 'Dnipro', 585);
insert into my_table values ('Dnipro', 'Chernivtsi', 891);
insert into my_table values ('Chernivtsi', 'Vinnytsia', 312);
insert into my_table values ('Kyiv', 'Dnipro', 533);
insert into my_table values ('Simferopol', 'Odesa', 392);
insert into my_table values ('Sumy', 'Lviv', 896);
insert into my_table values ('Odesa', 'Sumy', 800);
insert into my_table values ('Lviv', 'Simferopol', 1173);
insert into my_table values ('Kherson', 'Kyiv', 551);
insert into my_table values ('Simferopol', 'Poltava', 635);

select * from my_table;

```

```

SQL> create table my_table(point_A varchar(13),
 2                      point_B varchar(13),
 3                      distance integer);

Table created.

SQL> insert into my_table values ('Chernihiv', 'Vinnytsia', 423);

1 row created.

SQL> insert into my_table values ('Chernihiv', 'Dnipro', 585);

1 row created.

SQL> insert into my_table values ('Dnipro', 'Chernivtsi', 891);

1 row created.

SQL> insert into my_table values ('Chernivtsi', 'Vinnytsia', 312);

1 row created.

SQL> insert into my_table values ('Kyiv', 'Dnipro', 533);

1 row created.

SQL> insert into my_table values ('Simferopol', 'Odesa', 392);

1 row created.

SQL> insert into my_table values ('Sumy', 'Lviv', 896);

1 row created.

SQL> insert into my_table values ('Odesa', 'Sumy', 800);

1 row created.

SQL> insert into my_table values ('Lviv', 'Simferopol', 1173);

1 row created.

SQL> insert into my_table values ('Kherson', 'Kyiv', 551);

1 row created.

SQL> insert into my_table values ('Simferopol', 'Poltava', 635);

1 row created.

```

POINT_A	POINT_B	DISTANCE
Chernihiv	Vinnytsia	423
Chernihiv	Dnipro	585
Dnipro	Chernivtsi	891
Chernivtsi	Vinnytsia	312
Kyiv	Dnipro	533
Simferopol	Odesa	392
Sumy	Lviv	896
Odesa	Sumy	800
Lviv	Simferopol	1173
Kherson	Kyiv	551
Simferopol	Poltava	635

11 rows selected.

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

```
select distinct point_B, level transfer
from my_table
where level = 2
and point_B != 'Sumy'
start with point_A = 'Sumy'
connect by nocycle prior point_B = point_A
order by point_B;
```

```
SQL> select distinct point_B, level transfer
  2  from my_table
  3  where level = 2
  4  and point_B != 'Sumy'
  5  start with point_A = 'Sumy'
  6  connect by nocycle prior point_B = point_A
  7  order by point_B;
```

POINT_B	TRANSFER
Simferopol	2

Перевірка:

```
select distinct point_B, level transfer
from my_table
where point_B != 'Sumy'
start with point_A = 'Sumy'
connect by nocycle prior point_B =
point_A
order by point_B;
```

```
SQL> select distinct point_B, level transfer
  2  from my_table
  3  where point_B != 'Sumy'
  4  start with point_A = 'Sumy'
  5  connect by nocycle prior point_B = point_A
  6  order by point_B;
```

POINT_B	TRANSFER
Lviv	1
Odesa	3
Poltava	3
Simferopol	2

2. \* Знайдіть маршрут з мінімальною кількістю пересадок між двома вибраними вами містами.

```
select *
from
(select level transfer,
sys_connect_by_path(point_A,'->') || '==>' || point_B full_path
from my_table
where point_B = 'Vinnytsia'
start with point_A = 'Chernihiv'
connect by nocycle prior point_B = point_A
order by transfer)
where rownum = 1;
```

```

SQL> select *
  2  from
  3    (select level transfer,
  4     sys_connect_by_path(point_A, ' -> ') || ' ==> ' || point_B full_path
  5    from my_table
  6   where point_B = 'Vinnytsia'
  7   start with point_A = 'Chernihiv'
  8   connect by nocycle prior point_B = point_A
  9   order by transfer)
10  where rownum = 1;

      TRANSFER
-----
FULL_PATH
-----
          1
-> Chernihiv ==> Vinnytsia

```

Перевірка:

```

select *
from
(select level transfer,
sys_connect_by_path(point_A,' -> ') || ' ==> ' || point_B full_path
from my_table
where point_B = 'Vinnytsia'
start with point_A = 'Chernihiv'
connect by nocycle prior point_B = point_A
order by transfer);

```

```

SQL> select *
  2  from
  3    (select level transfer,
  4     sys_connect_by_path(point_A, ' -> ') || ' ==> ' || point_B full_path
  5    from my_table
  6   where point_B = 'Vinnytsia'
  7   start with point_A = 'Chernihiv'
  8   connect by nocycle prior point_B = point_A
  9   order by transfer);

      TRANSFER
-----
FULL_PATH
-----
          1
-> Chernihiv ==> Vinnytsia

          3
-> Chernihiv -> Dnipro -> Chernivtsi ==> Vinnytsia

```

3. \* Знайдіть найкоротший маршрут між обраними вами містами. (Підказки на наступних слайдах)

<https://stackoverflow.com/questions/1267025/how-to-calculate-value-of-string-in-oracle> :

```
create or replace function new_user.calc(pi_val varchar2) return number
is
    v_return number;
begin
    execute immediate 'select'||pi_val||' from dual' INTO v_return;
    return v_return;
end;
/
```

```
SQL> create or replace function new_user.calc(pi_val varchar2) return number
  2  is
  3      v_return number;
  4  begin
  5      execute immediate 'select'||pi_val||' from dual' INTO v_return;
  6      return v_return;
  7  end;
  8 /

Function created.
```

**select new\_user.calc('2+3') from dual;**

```
SQL> select new_user.calc('2+3') from dual;

NEW_USER.CALC('2+3')
-----
5
```

```
select *
from
(select level transfer,
new_user.calc('0' || sys_connect_by_path(distance, '+')) distance,
sys_connect_by_path(point_A,'-> ') || ' ==> ' || point_B full_path
from my_table
where point_B = 'Vinnytsia'
start with point_A = 'Chernihiv'
connect by nocycle prior point_B = point_A
order by distance)
where rownum = 1;
```

```

SQL> select *
  2  from
  3      (select level transfer,
  4      new_user.calc('0' || sys_connect_by_path(distance, '+')) distance,
  5      sys_connect_by_path(point_A, ' -> ') || ' ==> ' || point_B full_path
  6      from my_table
  7      where point_B = 'Vinnytsia'
  8      start with point_A = 'Chernihiv'
  9      connect by nocycle prior point_B = point_A
 10     order by distance)
 11  where rownum = 1;

      TRANSFER    DISTANCE
      ----- -----
FULL_PATH

      1          423
-> Chernihiv ==> Vinnytsia

```

Перевірка:

```

select level transfer,
new_user.calc('0' || sys_connect_by_path(distance, '+')) distance,
sys_connect_by_path(point_A, ' -> ') || ' ==> ' || point_B full_path
from my_table
where point_B = 'Vinnytsia'
start with point_A = 'Chernihiv'
connect by nocycle prior point_B = point_A
order by distance;

```

```

SQL> select level transfer,
  2  new_user.calc('0' || sys_connect_by_path(distance, '+')) distance,
  3  sys_connect_by_path(point_A, ' -> ') || ' ==> ' || point_B full_path
  4  from my_table
  5  where point_B = 'Vinnytsia'
  6  start with point_A = 'Chernihiv'
  7  connect by nocycle prior point_B = point_A
  8  order by distance;

      TRANSFER    DISTANCE
      ----- -----
FULL_PATH

      1          423
-> Chernihiv ==> Vinnytsia

      3          1788
-> Chernihiv -> Dnipro -> Chernivtsi ==> Vinnytsia

```

Видаліть таблицю.

**drop table my\_table;**

```

SQL> drop table my_table;
Table dropped.

```

---

#### 4.3 (підказки)

Як отримати довжину маршруту?

`sys_connect_by_path` дозволяє отримати шлях від поточного рядка до кореневого; `'0' || sys_connect_by_path (length, '+')` для кожного вузла дасть нам рядок із сумою відстаней до кореня (наприклад `'0 + 1 + 1 + 2'`);

Обчислити рядок `'0 + 1 + 1 + 2'` можна декількома способами;

На думку Oracle-гуру TOM KYTE кращий спосіб:

```
SELECT XMLQUERY ('0+1+1+2' RETURNING CONTENT).getnumberval () AS x FROM dual;
```

Проблема - не працює в Oracle 10 XE через ліцензійні обмеження (в 11 XE - працює)

4.3 Короткий погляд в бік процедур, що зберігаються

Рішення з <http://stackoverflow.com/questions/1267025/how-to-calculate-value-of-string-in-oracle>

```
SQL> CREATE OR REPLACE FUNCTION calc(pi_val VARCHAR2) RETURN NUMBER IS
```

```
2 v_return NUMBER;
```

```
3 BEGIN
```

```
4 EXECUTE IMMEDIATE 'select'||pi_val||'from dual' INTO v_return;
```

```
5 RETURN v_return;
```

```
6 END;
```

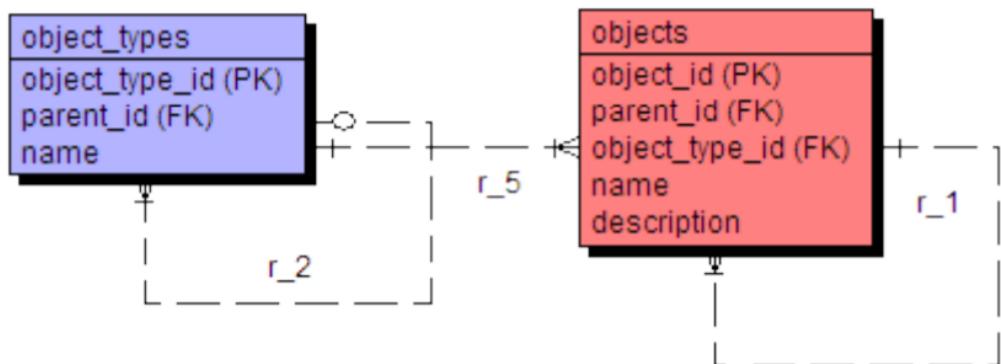
```
7 /
```

```
SQL> SELECT calc('2*6*10') FROM dual; CALC('2*6*10') ----- 120
```

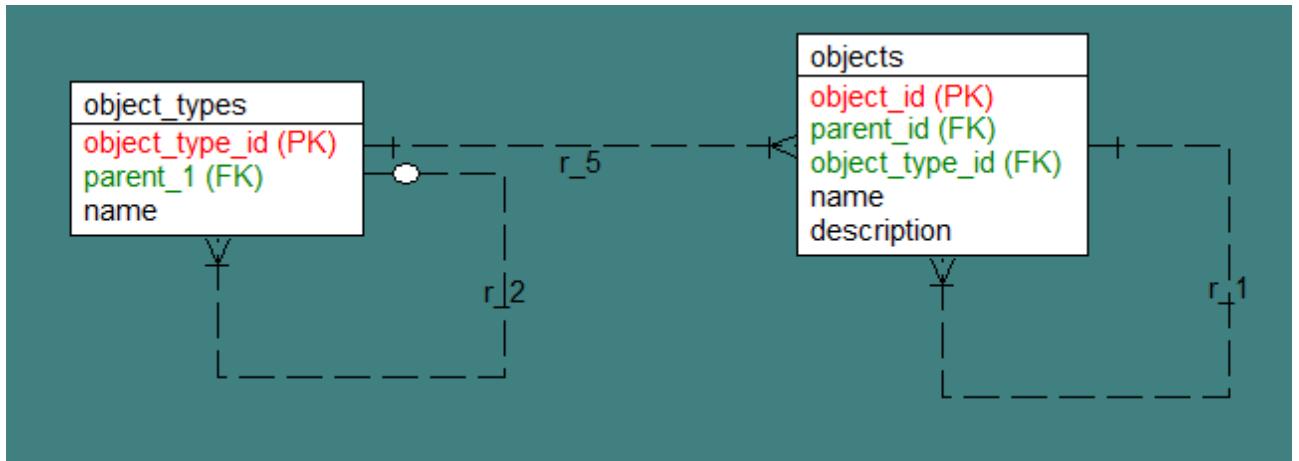
Обмеження: рішення є потенційною загрозою безпеці. Дивіться обговорення за посиланням

---

#### 5.1\*\*. Підготовчий етап



Напишіть скрипт для створення таблиць Objects і Object\_types.



```

Create table object_types11 (
    "object_type_id" Number(20) NOT NULL,
    "parent_id" Number(20) default null,
    "name" Varchar2 (20),
primary key ("object_type_id")
);

```

```

Create table objects11 (
    "object_id" Number(20) NOT NULL ,
    "parent_id" Number(20) default null,
    "object_type_id" Number(20) NOT NULL,
    "name" Varchar2 (20),
    "description" Varchar2 (20),
primary key ("object_id")
);

```

```

Alter table objects11 add foreign key ("parent_id") references
objects11("object_id");
Alter table object_types11 add foreign key ("parent_id") references object_types11
("object_type_id");
Alter table objects11 add foreign key ("object_type_id") references object_types11
("object_type_id");

```

```

SQL> Create table object_types11 (
  2  "object_type_id" Number(20) NOT NULL,
  3  "parent_id" Number(20) default null,
  4  "name" Varchar2 (20),
  5 primary key ("object_type_id")
  6 );

Table created.

SQL> Create table objects11 (
  2  "object_id" Number(20) NOT NULL ,
  3  "parent_id" Number(20) default null,
  4  "object_type_id" Number(20) NOT NULL,
  5  "name" Varchar2 (20),
  6  "description" Varchar2 (20),
  7 primary key ("object_id")
  8 );

Table created.

```

```

SQL> Alter table objects11 add foreign key ("parent_id") references objects11("object_id");
Table altered.

SQL> Alter table object_types11 add foreign key ("parent_id") references object_types11 ("object_type_id");
Table altered.

SQL> Alter table objects11 add foreign key ("object_type_id") references object_types11 ("object_type_id");
Table altered.

```

**describe object\_types11;**  
**describe objects11;**

Name	Null?	Type
object_type_id	NOT NULL	NUMBER(20)
parent_id		NUMBER(20)
name		VARCHAR2(20)

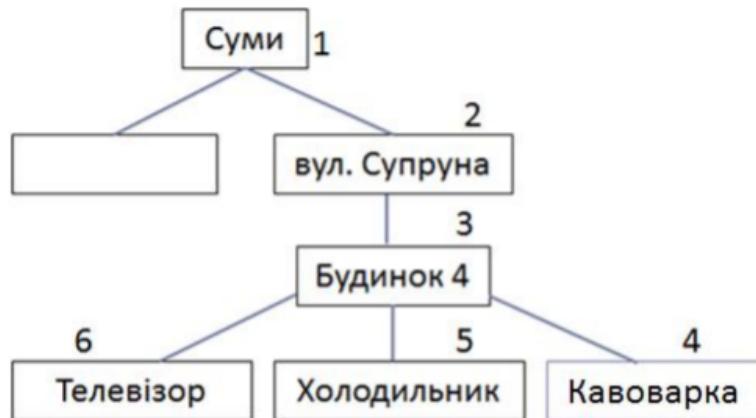
  

Name	Null?	Type
object_id	NOT NULL	NUMBER(20)
parent_id		NUMBER(20)
object_type_id	NOT NULL	NUMBER(20)
name		VARCHAR2(20)
description		VARCHAR2(20)

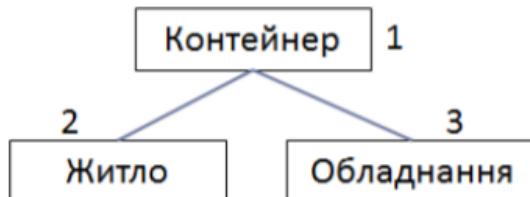
## 5.2\*\*. Предметна область

Опис предметної області: місто, вулиця, будинок, обладнання.

Дані таблиці Objects описуються деревом:



Дані таблиці Object\_types описуються деревом:



Напишіть скрипт для заповнення таблиць Objects і Object\_types.

```

insert into object_types11 values(31, default, 'CONTAINER');
insert into object_types11 values(32, 31, 'HOUSING');
insert into object_types11 values(33, 31, 'EQUIPMENT');

select * from object_types11;

```

```

SQL> insert into object_types11 values(31, default, 'CONTAINER');

1 row created.

SQL> insert into object_types11 values(32, 31, 'HOUSING');

1 row created.

SQL> insert into object_types11 values(33, 31, 'EQUIPMENT');

1 row created.

SQL> select * from object_types11;

object_type_id  parent_id name
-----  -----
      31 (null)    CONTAINER
      32          31 HOUSING
      33          31 EQUIPMENT

```

```

insert into objects11 values(101, default, 32, 'SUMY', 'HOUSING');
insert into objects11 values(100, 101, 31, null, 'CONTAINER');
insert into objects11 values(102, 101, 32, 'SUPRUN STREET', 'HOUSING');
insert into objects11 values(103, 102, 32, 'HOUSE 4', 'HOUSING');
insert into objects11 values(104, 103, 33, 'COFFEE MAKER', 'EQUIPMENT');
insert into objects11 values(105, 103, 33, 'FRIDGE', 'EQUIPMENT');
insert into objects11 values(106, 103, 33, 'TV', 'EQUIPMENT');

select * from objects11;

```

```

SQL> insert into objects11 values(101, default, 32, 'SUMY', 'HOUSING');

1 row created.

SQL> insert into objects11 values(100, 101, 31, null, 'CONTAINER');

1 row created.

SQL> insert into objects11 values(102, 101, 32, 'SUPRUN STREET', 'HOUSING');

1 row created.

SQL> insert into objects11 values(103, 102, 32, 'HOUSE 4', 'HOUSING');

1 row created.

SQL> insert into objects11 values(104, 103, 33, 'COFFEE MAKER', 'EQUIPMENT');

1 row created.

SQL> insert into objects11 values(105, 103, 33, 'FRIDGE', 'EQUIPMENT');

1 row created.

SQL> insert into objects11 values(106, 103, 33, 'TV', 'EQUIPMENT');

1 row created.

```

```
SQL> select * from objects11;
```

object_id	parent_id	object_type_id	name	description
101	(null)	32	SUMY	HOUSING
100	101	31	(null)	CONTAINER
102	101	32	SUPRUN STREET	HOUSING
103	102	32	HOUSE 4	HOUSING
104	103	33	COFFEE MAKER	EQUIPMENT
105	103	33	FRIDGE	EQUIPMENT
106	103	33	TV	EQUIPMENT

```
7 rows selected.
```

### 5.3\*\*. Завдання

1. Виведіть назви об'єктів, що належать до типу «Обладнання»;

```
select obj."name"  
from object_types11 tp, objects11 obj  
where tp."object_type_id" = obj."object_type_id"  
and tp."name" = 'EQUIPMENT';
```

```
SQL> select obj."name"  
  2  from object_types11 tp, objects11 obj  
  3  where tp."object_type_id" = obj."object_type_id"  
  4  and tp."name" = 'EQUIPMENT';  
  
name  
-----  
COFFEE MAKER  
FRIDGE  
TV
```

Перевірка:

```
select "object_id", obj."parent_id", obj."object_type_id", obj."name",  
"description", tp."object_type_id", tp."name"  
  
from object_types11 tp, objects11 obj  
  
where tp."object_type_id" = obj."object_type_id"  
  
and tp."name" = 'EQUIPMENT';
```

```
SQL> select "object_id", obj."parent_id", obj."object_type_id", obj."name", "description", tp."object_type_id", tp."name"  
  2  from object_types11 tp, objects11 obj  
  3  where tp."object_type_id" = obj."object_type_id"  
  4  and tp."name" = 'EQUIPMENT';
```

object_id	parent_id	object_type_id	name	description	object_type_id	name
104	103	33	COFFEE MAKER	EQUIPMENT	33	EQUIPMENT
105	103	33	FRIDGE	EQUIPMENT	33	EQUIPMENT
106	103	33	TV	EQUIPMENT	33	EQUIPMENT

2. Виведіть назву типу об'єкта з ім'ям «Будинок»;

```
select tp."name"
from object_types11 tp, objects11 obj
where tp."object_type_id" = obj."object_type_id"
and obj."name" like 'HOUSE%';
```

```
SQL> select tp."name"
  2  from object_types11 tp, objects11 obj
  3  where tp."object_type_id" = obj."object_type_id"
  4  and obj."name" like 'HOUSE%';

      name
-----
HOUSING
```

Перевірка:

```
select "object_id", obj."parent_id", obj."object_type_id", obj."name",
"description", tp."object_type_id", tp."name"
from object_types11 tp, objects11 obj
where tp."object_type_id" = obj."object_type_id"
and obj."name" like 'HOUSE%';
```

```
SQL> select "object_id", obj."parent_id", obj."object_type_id", obj."name", "description", tp."object_type_id", tp."name"
  2  from object_types11 tp, objects11 obj
  3  where tp."object_type_id" = obj."object_type_id"
  4  and obj."name" like 'HOUSE%';

   object_id  parent_id  object_type_id  name          description      object_type_id  name
-----  -----  -----  -----
        103       102            32  HOUSE 4    HOUSING           32  HOUSING
```

3. Виведіть назви об'єктів, які мають тип, що успадкований від типу «Контейнер»;

```
select obj."name"
from object_types11 tp, objects11 obj
where tp."object_type_id" = obj."object_type_id"
and level >= 2
start with tp."parent_id" is null
connect by nocycle prior tp."object_type_id" = tp."parent_id"
order by level;
```

```
SQL> select obj."name"
  2  from object_types11 tp, objects11 obj
  3  where tp."object_type_id" = obj."object_type_id"
  4  and level >= 2
  5  start with tp."parent_id" is null
  6  connect by nocycle prior tp."object_type_id" = tp."parent_id"
  7  order by level;

      name
-----
SUMY
HOUSE 4
FRIDGE
COFFEE MAKER
TV
SUPRUN STREET

6 rows selected.
```

Перевірка:

```
select level, "object_id", obj."parent_id", obj."object_type_id", obj."name",
"description",
       tp."object_type_id",                  tp."parent_id",                  tp."name",
SYS_CONNECT_BY_PATH(tp."name", '/') "PATH object_types"
from object_types11 tp, objects11 obj
where tp."object_type_id" = obj."object_type_id"
start with tp."parent_id" is null
connect by nocycle prior tp."object_type_id" = tp."parent_id"
order by level;
```

```
SQL> select level, "object_id", obj."parent_id", obj."object_type_id", obj."name", "description",
2          tp."object_type_id", tp."parent_id", tp."name", SYS_CONNECT_BY_PATH(tp."name", '/') "PATH object_types"
3  from object_types11 tp, objects11 obj
4 where tp."object_type_id" = obj."object_type_id"
5 start with tp."parent_id" is null
6 connect by nocycle prior tp."object_type_id" = tp."parent_id"
7 order by level;

      LEVEL    object_id   parent_id object_type_id name           description      object_type_id   parent_id name
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
PATH object_types
-----+-----+-----+-----+-----+-----+-----+-----+-----+
1      100        101      31 (null)    CONTAINER
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      103        102      32 HOUSE 4    HOUSING
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      105        103      33 FRIDGE     EQUIPMENT
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      104        103      33 COFFEE MAKER EQUIPMENT
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      106        103      33 TV         EQUIPMENT
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      101 (null)   32 SUMY      HOUSING
-----+-----+-----+-----+-----+-----+-----+-----+-----+
2      102        101      32 SUPRUN STREET HOUSING
-----+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows selected.
```

4. Вивести гілку дерева у вигляді: -> Суми -> вул. Супруна -> Будинок 4 -> Кавоварка.

```
select SYS_CONNECT_BY_PATH(obj."name", ' -> ') "PATH object name"
from objects11 obj
where obj."name" = 'COFFEE MAKER'
start with obj."parent_id" is null
connect by nocycle prior obj."object_id" = obj."parent_id"
order by level;
```

```
SQL> select SYS_CONNECT_BY_PATH(obj."name", ' -> ') "PATH object name"
2  from objects11 obj
3  where obj."name" = 'COFFEE MAKER'
4  start with obj."parent_id" is null
5  connect by nocycle prior obj."object_id" = obj."parent_id"
6  order by level;

      PATH object name
-----+-----+
-> SUMY -> SUPRUN STREET -> HOUSE 4 -> COFFEE MAKER
```

Перевірка:

```
select level, "object_id", obj."parent_id", obj."object_type_id", obj."name",
"description",

        SYS_CONNECT_BY_PATH(obj."name", ' -> ') "PATH object"

from objects11 obj

start with obj."parent_id" is null

connect by nocycle prior obj."object_id" = obj."parent_id"

order by level;
```

```
SQL> select level, "object_id", obj."parent_id", obj."object_type_id", obj."name", "description",
  2          SYS_CONNECT_BY_PATH(obj."name", ' -> ') "PATH object"
  3  from objects11 obj
  4 start with obj."parent_id" is null
  5 connect by nocycle prior obj."object_id" = obj."parent_id"
  6 order by level;
```

LEVEL	object_id	parent_id	object_type_id	name	description
					PATH object
1	101	(null)	32	SUMY	HOUSING
->	SUMY				
2	100	101	31	(null)	CONTAINER
->	SUMY	->			
2	102	101	32	SUPRUN STREET	HOUSING
->	SUMY	->			
3	103	102	32	HOUSE 4	HOUSING
->	SUMY	->			
4	104	103	33	COFFEE MAKER	EQUIPMENT
->	SUMY	->			
4	105	103	33	FRIDGE	EQUIPMENT
->	SUMY	->			
4	106	103	33	TV	EQUIPMENT
->	SUMY	->			

7 rows selected.

## Лабораторна робота 12

## Завдання

- Створіть тригер, який відстежує зміну зарплати користувача і заносить в таблицю звітності

```
-- drop table emp11;  
-- drop table emp22;
```

```
create table emp11(name char(10), salary integer);  
create table emp22(name char(10), old_salary integer, new_salary integer);
```

```
insert into emp11 values('Damon', 9800);  
insert into emp11 values('Stafan', 5400);  
insert into emp11 values('Alaric', 3700);  
insert into emp11 values('Matt', 1300);  
insert into emp11 values('Tyler', 900);  
insert into emp11 values('Jeremy', 750);
```

```
SQL> create table emp11(name char(10), salary integer);  
Table created.  
  
SQL> create table emp22(name char(10), old_salary integer, new_salary integer);  
Table created.  
  
SQL> insert into emp11 values('Damon', 9800);  
1 row created.  
  
SQL> insert into emp11 values('Stafan', 5400);  
1 row created.  
  
SQL> insert into emp11 values('Alaric', 3700);  
1 row created.  
  
SQL> insert into emp11 values('Matt', 1300);  
1 row created.  
  
SQL> insert into emp11 values('Tyler', 900);  
1 row created.  
  
SQL> insert into emp11 values('Jeremy', 750);  
1 row created.
```

```
select * from emp11;
```

```
SQL> select * from emp11;

NAME          SALARY
-----        -----
Damon          9800
Stafan         5400
Alaric         3700
Matt            1300
Tyler           900
Jeremy          750

6 rows selected.
```

```
create or replace trigger trig_task1
after update of salary on emp11
referencing old as oldRow new as newRow
for each row
when (newRow.salary != oldRow.salary)
begin
insert into emp22 values(:oldRow.name, :oldRow.salary, :newRow.salary);
end;
/
```

```
SQL> create or replace trigger trig_task1
  2  after update of salary on emp11
  3  referencing old as oldRow new as newRow
  4  for each row
  5  when (newRow.salary != oldRow.salary)
  6  begin
  7  insert into emp22 values(:oldRow.name, :oldRow.salary, :newRow.salary);
  8  end;
  9  /

```
Trigger created.
```


```

```
select * from emp22;
```

```
update emp11
```

```
set salary = salary + 200;
```

```
select * from emp11;
```

```
SQL> select * from emp22;
no rows selected

SQL> update emp11
  2  set salary = salary + 200;
  3
  4 6 rows updated.

SQL> select * from emp11;

NAME          SALARY
-----        -----
Damon          10000
Stafan         5600
Alaric         3900
Matt            1500
Tyler           1100
Jeremy          950

6 rows selected.
```

```
select * from emp22;
```

```
SQL> select * from emp22;

      NAME        OLD_SALARY  NEW_SALARY
-----  -----
Damon          9800       10000
Stafan         5400       5600
Alaric         3700       3900
Matt           1300       1500
Tyler          900        1100
Jeremy         750        950

6 rows selected.
```

```
update emp11
```

```
set salary = salary * 2
```

```
where name in ('Damon', 'Stafan', 'Alaric');
```

```
select * from emp11;
```

```
select * from emp22;
```

```
SQL> update emp11
  2  set salary = salary * 2
  3  where name in ('Damon', 'Stafan', 'Alaric');

3 rows updated.

SQL> select * from emp11;

      NAME        SALARY
-----  -----
Damon          20000
Stafan         11200
Alaric         7800
Matt           1500
Tyler          1100
Jeremy         950

6 rows selected.

SQL> select * from emp22;

      NAME        OLD_SALARY  NEW_SALARY
-----  -----
Damon          9800       10000
Stafan         5400       5600
Alaric         3700       3900
Matt           1300       1500
Tyler          900        1100
Jeremy         750        950
Damon          10000      20000
Stafan         5600       11200
Alaric         3900       7800

9 rows selected.
```

```
drop table emp11;
drop table emp22;
```

```
SQL> drop table emp11;
Table dropped.

SQL> drop table emp22;
Table dropped.
```

2. \* Створіть тригер, який відстежує зміну посади користувача і записує її в журнал. Журнал зберігає стару і нову посаду і час зміни. У разі видалення користувача він вважається звільненим, про що заноситься запис в журнал. У разі додавання користувача він вважається прийнятим на роботу, про що заноситься запис у журнал.

```
--drop table emp2;
--drop table register;

create table emp2(name char(10), job char(10));

create table register(name char(10), old_job char(10), new_job char(10),
time_of_change date);

create or replace trigger trig_task2
after insert or update or delete on emp2
referencing old as oldRow new as newRow
for each row
begin
if inserting then
insert into register values(:newRow.name, 'HIRED', :newRow.job, sysdate);
elsif updating then
insert into register values(:oldRow.name, :oldRow.job, :newRow.job, sysdate);
elsif deleting then
insert into register values(:oldRow.name, :oldRow.job, 'DISMISSED', sysdate);
end if;
end;
/

insert into emp2 values('Damon', 'MANAGER');
insert into emp2 values('Stafan', 'MANAGER');
insert into emp2 values('Alaric', 'SALESMAN');
insert into emp2 values('Matt', 'CLARK');
insert into emp2 values('Tyler', 'ANALYST');
insert into emp2 values('Jeremy', 'ANALYST');
```

```

SQL> create table emp2(name char(10), job char(10));

Table created.

SQL> create table register(name char(10), old_job char(10), new_job char(10), time_of_change date);

Table created.

SQL> create or replace trigger trig_task2
  2  after insert or update or delete on emp2
  3  referencing old as oldRow new as newRow
  4  for each row
  5  begin
  6    if inserting then
  7      insert into register values(:newRow.name,      'HIRED', :newRow.job, sysdate);
  8    elsif updating then
  9      insert into register values(:oldRow.name, :oldRow.job, :newRow.job, sysdate);
 10   elsif deleting then
 11     insert into register values(:oldRow.name, :oldRow.job, 'DISMISSED', sysdate);
 12   end if;
 13 end;
 14 /

```

Trigger created.

```

SQL> insert into emp2 values('Damon',  'MANAGER');

1 row created.

SQL> insert into emp2 values('Stafan', 'MANAGER');

1 row created.

SQL> insert into emp2 values('Alaric', 'SALESMAN');

1 row created.

SQL> insert into emp2 values('Matt',   'CLARK');

1 row created.

SQL> insert into emp2 values('Tyler',  'ANALYST');

1 row created.

SQL> insert into emp2 values('Jeremy', 'ANALYST');

1 row created.

```

**select \* from emp2;**

```

SQL> select * from emp2;

      NAME      JOB
      ----- -----
Damon      MANAGER
Stafan     MANAGER
Alaric     SALESMAN
Matt       CLARK
Tyler      ANALYST
Jeremy     ANALYST

6 rows selected.

```

```
select name, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
from register;
```

```
SQL> select name, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
  2  from register;

NAME      OLD_JOB    NEW_JOB     TIME
-----  -----  -----  -----
Damon      HIRED      MANAGER    21:13:06 PM
Stafan     HIRED      MANAGER    21:13:06 PM
Alaric     HIRED      SALESMAN   21:13:06 PM
Matt       HIRED      CLARK      21:13:06 PM
Tyler      HIRED      ANALYST   21:13:06 PM
Jeremy    HIRED      ANALYST   21:13:08 PM

6 rows selected.
```

```
update emp2
set job = 'PRESIDENT'
where name = 'Damon';
```

```
update emp2
set job = 'SALESMAN'
where name = 'Jeremy';
```

```
update emp2
set job = 'MANAGER'
where name = 'Alaric';
```

```
SQL> update emp2
  2  set job = 'PRESIDENT'
  3  where name = 'Damon';

1 row updated.

SQL> update emp2
  2  set job = 'SALESMAN'
  3  where name = 'Jeremy';

1 row updated.

SQL> update emp2
  2  set job = 'MANAGER'
  3  where name = 'Alaric';

1 row updated.
```

```
select * from emp2;
```

```
select name, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
from register;
```

```

SQL> select * from emp2;

NAME      JOB
-----
Damon    PRESIDENT
Stafan   MANAGER
Alaric   MANAGER
Matt     CLARK
Tyler    ANALYST
Jeremy   SALESMAN

6 rows selected.

SQL> select name, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
  2  from register;

NAME      OLD_JOB    NEW_JOB    TIME
-----
Damon    HIRED      MANAGER    21:13:06 PM
Stafan   HIRED      MANAGER    21:13:06 PM
Alaric   HIRED      SALESMAN   21:13:06 PM
Matt     HIRED      CLARK     21:13:06 PM
Tyler    HIRED      ANALYST   21:13:06 PM
Jeremy   HIRED      ANALYST   21:13:08 PM
Damon    MANAGER    PRESIDENT  21:16:19 PM
Jeremy   ANALYST   SALESMAN   21:16:31 PM
Alaric   SALESMAN   MANAGER    21:16:37 PM

9 rows selected.

```

**delete from emp2**

**where job in ('CLARK', 'ANALYST');**

**delete from emp2**

**where job in ('PRESIDENT');**

**insert into emp2 values('Klaus', 'PRESIDENT');**

```

SQL> delete from emp2
  2  where job in ('CLARK', 'ANALYST');

2 rows deleted.

SQL> delete from emp2
  2  where job in ('PRESIDENT');

1 row deleted.

SQL> insert into emp2 values('Klaus', 'PRESIDENT');

1 row created.

```

**select \* from emp2;**

**select name, old\_job, new\_job, to\_char(time\_of\_change, 'HH24:MI:SS AM') as time**

**from register;**

```

SQL> select * from emp2;

NAME      JOB
-----
Stafan    MANAGER
Alaric    MANAGER
Jeremy   SALESMAN
Klaus     PRESIDENT

SQL> select name, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
  2  from register;

NAME      OLD_JOB    NEW_JOB    TIME
-----
Damon    HIRED      MANAGER    21:13:06 PM
Stafan    HIRED      MANAGER    21:13:06 PM
Alaric    HIRED      SALESMAN  21:13:06 PM
Matt     HIRED      CLARK     21:13:06 PM
Tyler    HIRED      ANALYST   21:13:06 PM
Jeremy   HIRED      ANALYST   21:13:08 PM
Damon    MANAGER    PRESIDENT  21:16:19 PM
Jeremy   ANALYST   SALESMAN  21:16:31 PM
Alaric    SALESMAN  MANAGER    21:16:37 PM
Matt     CLARK      DISMISSED  21:18:44 PM
Tyler    ANALYST   DISMISSED  21:18:44 PM
Damon    PRESIDENT  DISMISSED  21:18:46 PM
Klaus    HIRED      PRESIDENT  21:19:01 PM

13 rows selected.

```

**drop table emp2;**

```
SQL> drop table emp2;
```

Table dropped.

**drop table register;**

```
SQL> drop table register;
```

Table dropped.

## Завдання

3. \*Візьміть з тексту лекції приклад тригера реалізації каскадного видалення. Реалізуйте триггер, що видаляє всіх співробітників, які працюють у відділі, при видаленні відділу.

```
--drop table emp3;
--drop table dept3;
```

```

CREATE TABLE dept3( -- описание подразделения
  deptno NUMBER(3) NOT NULL, -- № подразделения
  dname VARCHAR(14) NOT NULL, -- название подразделения
  loc VARCHAR(13), -- местоположение подразделения
  CONSTRAINT dept_pk3 PRIMARY KEY (deptno),
  CONSTRAINT dept_naem_uk3 UNIQUE (dname)
);
```

```
CREATE TABLE emp3 ( -- описание сотрудников подразделений
    empno NUMBER(4) NOT NULL, -- № сотрудника
    ename VARCHAR(20), -- имя сотрудника
    job VARCHAR(9), -- должность сотрудника
    sal NUMBER(7, 2), -- з/п сотрудника
    deptno NUMBER(2), -- № подразделения, где работает сотрудник
    CONSTRAINT emp_pk3 PRIMARY KEY (empno),
    CONSTRAINT emp_deptno_fk3 FOREIGN KEY (deptno) REFERENCES
    dept3 (deptno)
);

commit;
```

```
SQL> CREATE TABLE dept3( -- описание подразделения
    2      deptno NUMBER(3) NOT NULL, -- № подразделения
    3      dname VARCHAR(14) NOT NULL, -- название подразделения
    4      loc VARCHAR(13), -- местоположение подразделения
    5      CONSTRAINT dept_pk3 PRIMARY KEY (deptno),
    6      CONSTRAINT dept_naem_uk3 UNIQUE (dname)
    7  );
```

Table created.

```
SQL> CREATE TABLE emp3 ( -- описание сотрудников подразделений
    2      empno NUMBER(4) NOT NULL, -- № сотрудника
    3      ename VARCHAR(20), -- имя сотрудника
    4      job VARCHAR(9), -- должность сотрудника
    5      sal NUMBER(7, 2), -- з/п сотрудника
    6      deptno NUMBER(2), -- № подразделения, где работает сотрудник
    7      CONSTRAINT emp_pk3 PRIMARY KEY (empno),
    8      CONSTRAINT emp_deptno_fk3 FOREIGN KEY (deptno) REFERENCES dept3 (deptno)
    9  );
```

Table created.

```
SQL> commit;
```

```
Commit complete.
```

```
INSERT INTO dept3 VALUES (10,'ACCOUNTING','NEW_YORK');

INSERT INTO dept3 VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO dept3 VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO dept3 VALUES (40,'OPERATIONS', 'BOSTON');

INSERT INTO dept3 VALUES (50, 'RESEARCH2', 'HONKONG');
```

```
SQL> INSERT INTO dept3 VALUES (10,'ACCOUNTING','NEW_YORK');

1 row created.

SQL> INSERT INTO dept3 VALUES (20,  'RESEARCH',  'DALLAS');

1 row created.

SQL> INSERT INTO dept3 VALUES (30,      'SALES', 'CHICAGO');

1 row created.

SQL> INSERT INTO dept3 VALUES (40, 'OPERATIONS', 'BOSTON');

1 row created.

SQL> INSERT INTO dept3 VALUES (50, 'RESEARCH2', 'HONKONG');

1 row created.
```

**select \* from dept3;**

```
SQL> select * from dept3;

  DEPTNO DNAME        LOC
----- -----
    10 ACCOUNTING    NEW_YORK
    20 RESEARCH      DALLAS
    30 SALES         CHICAGO
    40 OPERATIONS    BOSTON
    50 RESEARCH2     HONKONG
```

**INSERT INTO emp3 VALUES (7839, 'KING','PRESIDENT', 5000, 10);**  
**INSERT INTO emp3 VALUES (7698, 'BLAKE', 'MANAGER', 2850, 20);**  
**INSERT INTO emp3 VALUES (7782, 'CLARK', 'MANAGER', 1500, 30);**  
**INSERT INTO emp3 VALUES (7566, 'JONES', 'SALESMAN', 1250, 40);**  
**INSERT INTO emp3 VALUES (7654,'MARTIN', 'SALESMAN', 2250, 50);**  
**INSERT INTO emp3 VALUES (7499, 'ALLEN', 'SALESMAN', 1600, 10);**  
**INSERT INTO emp3 VALUES (7844,'TURNER', 'SALESMAN', 1500, 20);**  
**select \* from emp3;**

```

SQL> INSERT INTO emp3 VALUES (7839, 'KING','PRESIDENT', 5000, 10);
1 row created.

SQL> INSERT INTO emp3 VALUES (7698, 'BLAKE', 'MANAGER', 2850, 20);
1 row created.

SQL> INSERT INTO emp3 VALUES (7782, 'CLARK', 'MANAGER', 1500, 30);
1 row created.

SQL> INSERT INTO emp3 VALUES (7566, 'JONES', 'SALESMAN', 1250, 40);
1 row created.

SQL> INSERT INTO emp3 VALUES (7654,'MARTIN', 'SALESMAN', 2250, 50);
1 row created.

SQL> INSERT INTO emp3 VALUES (7499, 'ALLEN', 'SALESMAN', 1600, 10);
1 row created.

SQL> INSERT INTO emp3 VALUES (7844,'TURNER', 'SALESMAN', 1500, 20);
1 row created.

SQL> select * from emp3;

      EMPNO ENAME          JOB         SAL   DEPTNO
----- -----
    7839 KING        PRESIDENT     5000       10
    7698 BLAKE       MANAGER      2850       20
    7782 CLARK       MANAGER      1500       30
    7566 JONES       SALESMAN     1250       40
    7654 MARTIN     SALESMAN     2250       50
    7499 ALLEN       SALESMAN     1600       10
    7844 TURNER      SALESMAN     1500       20

7 rows selected.

```

**commit;**

```

create or replace trigger trig_task3
after delete on dept3
referencing old as oldRow new as newRow
for each row
begin
delete from emp3
where emp3.deptno = :oldRow.deptno;
end;
/

```

```
delete from dept3
where loc in ('NEW_YORK', 'CHICAGO');
```

```
select * from dept3;
```

```
select * from emp3;
```

```
delete from dept3
where loc in ('BOSTON');
```

```
SQL> commit;

Commit complete.

SQL> create or replace trigger trig_task3
  2  after delete on dept3
  3  referencing old as oldRow new as newRow
  4  for each row
  5  begin
  6    delete from emp3
  7    where emp3.deptno = :oldRow.deptno;
  8  end;
 9  /

Trigger created.

SQL> delete from dept3
  2  where loc in ('NEW_YORK', 'CHICAGO');

2 rows deleted.

SQL> select * from dept3;

  DEPTNO DNAME          LOC
----- -----
        20 RESEARCH        DALLAS
        40 OPERATIONS      BOSTON
        50 RESEARCH2       HONKONG

SQL> select * from emp3;

  EMPNO ENAME          JOB            SAL  DEPTNO
----- -----
      7698 BLAKE         MANAGER       2850      20
      7566 JONES         SALESMAN     1250      40
      7654 MARTIN        SALESMAN     2250      50
      7844 TURNER        SALESMAN     1500      20

SQL> delete from dept3
  2  where loc in ('BOSTON');

1 row deleted.
```

```
select * from dept3;
```

```
select * from emp3;
```

```

SQL> select * from dept3;

  DEPTNO DNAME          LOC
----- -----
      20 RESEARCH        DALLAS
      50 RESEARCH2       HONKONG

SQL> select * from emp3;

  EMPNO ENAME           JOB            SAL  DEPTNO
----- -----
    7698 BLAKE           MANAGER        2850   20
    7654 MARTIN          SALESMAN       2250   50
    7844 TURNER          SALESMAN       1500   20

```

4. \* Перевірте, що при цьому спрацьовує тригер із завдання 2.

```

select * from dept3;
select * from emp3;

```

```

SQL> select * from dept3;

  DEPTNO DNAME          LOC
----- -----
      20 RESEARCH        DALLAS
      50 RESEARCH2       HONKONG

SQL> select * from emp3;

  EMPNO ENAME           JOB            SAL  DEPTNO
----- -----
    7698 BLAKE           MANAGER        2850   20
    7654 MARTIN          SALESMAN       2250   50
    7844 TURNER          SALESMAN       1500   20

```

```
--drop table register2;
```

```
create table register2(ename char(10), old_job char(10), new_job char(10),
time_of_change date);
```

```

create or replace trigger trig_task2_4
after insert or update or delete on emp3
referencing old as oldRow new as newRow
for each row
begin
if inserting then
  insert into register2 values(:newRow.ename, 'HIRED', :newRow.job, sysdate);
elsif updating then
  insert into register2 values(:oldRow.ename, :oldRow.job, :newRow.job,
sysdate);
elsif deleting then

```

```

insert into register2 values(:oldRow.ename, :oldRow.job, 'DISMISSED',
sysdate);
end if;
end;
/

delete from dept3
where loc in ('DALLAS');

select * from dept3;

select * from emp3;

select ename, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as
time
from register2;

```

```

SQL> create table register2(ename char(10), old_job char(10), new_job char(10), time_of_change date);

Table created.

SQL> create or replace trigger trig_task2_4
  2  after insert or update or delete on emp3
  3  referencing old as oldRow new as newRow
  4  for each row
  5  begin
  6    if inserting then
  7      insert into register2 values(:newRow.ename,      'HIRED', :newRow.job, sysdate);
  8    elsif updating then
  9      insert into register2 values(:oldRow.ename, :oldRow.job, :newRow.job, sysdate);
 10   elsif deleting then
 11     insert into register2 values(:oldRow.ename, :oldRow.job, 'DISMISSED', sysdate);
 12   end if;
 13 end;
 14 /


Trigger created.

SQL> delete from dept3
  2  where loc in ('DALLAS');

1 row deleted.

SQL> select * from dept3;

  DEPTNO DNAME          LOC
----- -----
      50 RESEARCH2      HONKONG

SQL> select * from emp3;

  EMPNO ENAME           JOB            SAL  DEPTNO
----- -----
    7654 MARTIN        SALESMAN       2250      50

SQL> select ename, old_job, new_job, to_char(time_of_change, 'HH24:MI:SS AM') as time
  2  from register2;

  ENAME    OLD_JOB    NEW_JOB    TIME
----- -----
  BLAKE     MANAGER   DISMISSED  23:40:10 PM
  TURNER    SALESMAN  DISMISSED  23:40:10 PM

```

```
drop table emp3;
drop table dept3;
drop table register2;
```

```
SQL> drop table emp3;
Table dropped.

SQL> drop table dept3;
Table dropped.

SQL> drop table register2;
Table dropped.
```

## Лабораторна робота 13

## 1. Призначення прав

1. Створіть таблицю My\_emp на основі таблиці Emp.

```
create table My_emp as
```

```
select * from emp;
```

```
select * from My_emp;
```

```
SQL> create table My_emp as
  2  select * from emp;
```

Table created.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

18 rows selected.

2. Перевірте, якими правами ви володієте, і до яких об'єктів у вас є доступ.

```
show user;
```

```
select * from USER_SYS_PRIVS;
```

```
select * from USER_TAB_PRIVS_MADE;
```

```
SQL> show user;
USER is "STUDENT_PM81_09"
SQL> select * from USER_SYS_PRIVS;

USERNAME                               PRIVILEGE          ADM COM INH
STUDENT_PM81_09                         UNLIMITED TABLESPACE    NO NO NO

SQL> select * from USER_TAB_PRIVS_MADE;

GRANTEE                                TABLE_NAME          PRIVILEGE          GRA HIE COM
GRANTOR                                 TYPE               INH
PUBLIC                                  PUBLIC             --                            --
STUDENT_PM81_09                          STUDENT_PM81_09      INHERIT PRIVILEGES   NO NO NO
STUDENT_PM81_09                          STUDENT_PM81_09      INHERIT PRIVILEGES   NO NO NO
USER                                    NO
```

3. Створіть користувача New\_User.

```
create user New_User11  
identified by New12345;  
  
grant create session to New_User11;
```

```
SQL> create user New_User11  
  2  identified by New12345;  
  
User created.  
  
SQL> grant create session to New_User11;  
  
Grant succeeded.
```

4. Передайте користувачу New\_User права на читання з таблиці My\_emp.

```
grant select  
on student_pm81_09.My_emp  
to New_User11;
```

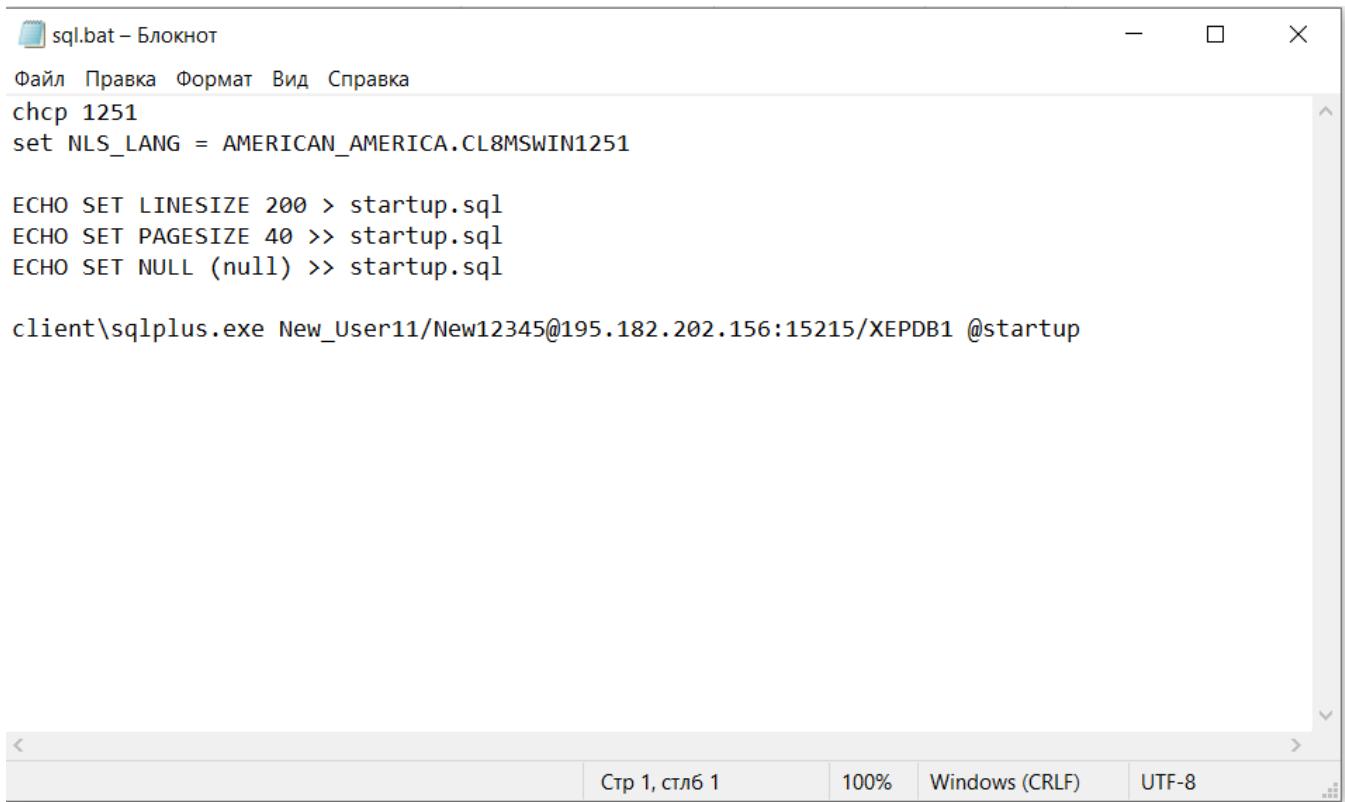
```
SQL> grant select  
  2  on student_pm81_09.My_emp  
  3  to New_User11;  
  
Grant succeeded.
```

5. Від імені користувача New\_User спробуйте читати і видаляти з таблиці My\_emp.

client\sqlplus.exe New\_User11/New12345@195.182.202.156:15215/XEPDB1 @startup

замість

client\sqlplus.exe student\_pm81\_09/student@195.182.202.156:15215/XEPDB1 @startup



```
sql.bat – Блокнот  
Файл Правка Формат Вид Справка  
chcp 1251  
set NLS_LANG = AMERICAN_AMERICA.CL8MSWIN1251  
  
ECHO SET LINESIZE 200 > startup.sql  
ECHO SET PAGESIZE 40 >> startup.sql  
ECHO SET NULL (null) >> startup.sql  
  
client\sqlplus.exe New_User11/New12345@195.182.202.156:15215/XEPDB1 @startup
```

Стр 1, стлб 1    100%    Windows (CRLF)    UTF-8

```
show user;
```

```
select * from student_pm81_09.My_emp;
```

```
delete
```

```
from student_pm81_09.My_emp
```

```
where empno = 8000;
```

```
SQL> show user;  
USER is "NEW_USER11"  
SQL> select * from student_pm81_09.My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

```
18 rows selected.
```

```
SQL> delete  
  2  from student_pm81_09.My_emp  
  3  where empno = 8000;  
from student_pm81_09.My_emp  
      *
```

```
ERROR at line 2:
```

```
ORA-01031: insufficient privileges
```

## 2. Делегування привілеїв

### 1. Створіть користувача New\_User2.

```
show user;
```

```
create user New_User22  
identified by New54321;
```

```
grant create session to New_User22;
```

```
SQL> show user;  
USER is "STUDENT_PM81_09"  
SQL> create user New_User22  
  2  identified by New54321;  
  
User created.
```

```
SQL> grant create session to New_User22;  
  
Grant succeeded.
```

2. Надайте користувачу New\_User права на видалення з таблиці My\_emp з можливістю передачі прав.

```
grant delete  
on student_pm81_09.My_emp  
to New_User11  
with grant option;
```

```
SQL> grant delete  
  2  on student_pm81_09.My_emp  
  3  to New_User11  
  4  with grant option;  
  
Grant succeeded.
```

3. Зайдіть від імені користувача New\_User і передайте права на видалення користувачу New\_User2.

```
show user;  
  
grant delete  
on student_pm81_09.My_emp  
to New_User22;
```

```
SQL> show user;  
USER is "NEW_USER11"  
SQL> grant delete  
  2  on student_pm81_09.My_emp  
  3  to New_User22;  
  
Grant succeeded.
```

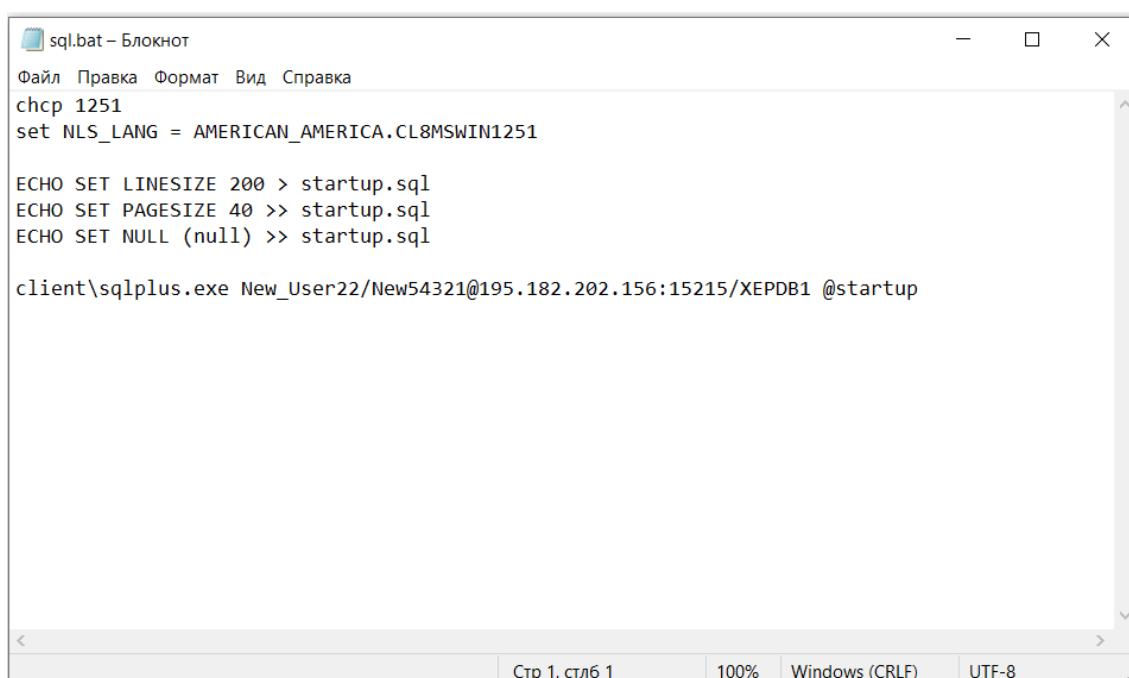
4. Від імені користувача Student відкличте права на видалення у користувача New\_User.

```
show user;  
  
revoke delete  
on student_pm81_09.My_emp  
from New_User11;
```

```
SQL> show user;  
USER is "STUDENT_PM81_09"  
SQL> revoke delete  
  2  on student_pm81_09.My_emp  
  3  from New_User11;  
  
Revoke succeeded.
```

5. Перегляньте права користувача New\_User2.

```
client\sqlplus.exe New_User22/New54321@195.182.202.156:15215/XEPDB1 @startup
```



```
sql.bat – Блокнот  
Файл Правка Формат Вид Справка  
chcp 1251  
set NLS_LANG = AMERICAN_AMERICA.CL8MSWIN1251  
  
ECHO SET LINESIZE 200 > startup.sql  
ECHO SET PAGESIZE 40 >> startup.sql  
ECHO SET NULL (null) >> startup.sql  
  
client\sqlplus.exe New_User22/New54321@195.182.202.156:15215/XEPDB1 @startup
```

```
show user;
```

```
select * from USER_SYS_PRIVS;
```

```
select * from USER_TAB_PRIVS_MADE;
```

USERNAME	PRIVILEGE	ADM	COM	INH
NEW_USER22	CREATE SESSION	NO	NO	NO
SQL> select * from USER_TAB_PRIVS_MADE;				
GRANTEE				
TABLE_NAME				
GRANTOR	PRIVILEGE	GRA	HIE	COM
TYPE	INH			
PUBLIC				
NEW_USER22				
USER	INHERIT PRIVILEGES	NO	NO	NO
NO				

### 3. Ролі

➤ Від імені Student створіть роль ROLE1 і ROLE2 з будь-якими привілеями.

```
create role ROLE1;
```

```
create role ROLE2;
```

```
SQL> create role ROLE1;
```

```
Role created.
```

```
SQL> create role ROLE2;
```

```
Role created.
```

```
grant select, insert
```

```
on My_emp
```

```
to ROLE1;
```

```
SQL> grant select, insert  
2  on My_emp  
3  to ROLE1;
```

```
Grant succeeded.
```

```
grant select, delete
```

```
on My_emp
```

```
to ROLE2;
```

```
SQL> grant select, delete  
2  on My_emp  
3  to ROLE2;
```

```
Grant succeeded.
```

➤ Спробуйте призначити користувачу

New\_User роль ROLE1, а користувачу

New\_User2 ролі ROLE1 і ROLE2.

```
grant ROLE1 to New_User11, New_User22;
```

```
grant ROLE2 to New_User22;
```

```
SQL> grant ROLE1 to New_User11, New_User22;  
Grant succeeded.  
  
SQL> grant ROLE2 to New_User22;  
Grant succeeded.
```

➤ Відкличте у користувача New\_User2 роль ROLE2.

**revoke ROLE2 from New\_User2;**

```
SQL> revoke ROLE2 from New_User2;
Revoke succeeded.
```

➤ Спробуйте призначити користувачу New\_User роль ROLE1.

**grant ROLE1 to New\_User1;**

```
SQL> grant ROLE1 to New_User1;
Grant succeeded.
```

➤ Спробуйте призначити ролі ROLE1 роль ROLE2.

**grant ROLE2 to ROLE1;**

```
SQL> grant ROLE2 to ROLE1;
Grant succeeded.
```

➤ Спробуйте призначити ролі ROLE1 роль ROLE1.

**grant ROLE1 to ROLE1;**

```
SQL> grant ROLE1 to ROLE1;
grant ROLE1 to ROLE1
*
ERROR at line 1:
ORA-01934: circular role grant detected
```

#### 4. Зачищення території

➤ Видаліть таблицю My\_emp.

➤ Видаліть ролі ROLE1 і ROLE2.

➤ Видаліть користувачів New\_User і New\_User2.

**drop table My\_emp;**

```
SQL> drop table My_emp;
```

```
Table dropped.
```

**drop role ROLE1;**

```
SQL> drop role ROLE1;
```

```
Role dropped.
```

**drop user New\_User1;**

```
SQL> drop user New_User1;
```

```
User dropped.
```

**drop user New\_User2;**

```
SQL> drop user New_User2;
User dropped.
```

```
Role dropped.
```

## 5. Транзакції

1. Створіть таблицю My\_emp із таблиці Emp, скопіюйте в неї всі дані з таблиці Emp.

**create table My\_emp as**

**select \* from emp;**

**select \* from My\_emp;**

```
SQL> create table My_emp as
  2  select * from emp;

Table created.

SQL> select * from My_emp;

  EMPNO ENAME          JOB           MGR HIREDATE        SAL      COMM  DEPTNO
----- -----
    7839 KING            PRESIDENT   (null) 17-NOV-11     5000  (null)    10
    7698 BLAKE           MANAGER    7839 01-MAY-11    2850  (null)    30
    7782 CLARK           MANAGER    7839 09-JUN-11    1500  (null)    10
    7566 JONES           MANAGER    7839 02-APR-11    2975  (null)    20
    7654 MARTIN          SALESMAN   7698 28-SEP-11    1250   1400    30
    7499 ALLEN           SALESMAN   7698 20-FEB-11    1600    300    30
    7844 TURNER          SALESMAN   7698 08-SEP-11    1500     0    30
    7900 JAMES            CLERK     7698 03-DEC-11    950  (null)    30
    7521 WARD             SALESMAN   7698 22-FEB-11    1250    500    30
    7902 FORD              ANALYST   7566 03-DEC-11    3000  (null)    20
    7369 SMITH             CLERK     7902 17-DEC-10    800  (null)    20
    7788 SCOTT             ANALYST   7566 09-DEC-12    3000  (null)    20
    7876 ADAMS             CLERK     7788 12-JAN-13    1100  (null)    20
    7934 MILLER            CLERK     7782 23-JAN-12    1300  (null)    10
    8000 JACKIE CHAN       SALESMAN   7839 28-SEP-11    2250   1700    60
    8001 JET LI             SALESMAN   8000 20-FEB-11    2600    600    60
    8002 BRUCE LEE          SALESMAN   8000 08-SEP-11    2500  (null)    60
    8003 DR NO              ANALYST   7839 11-SEP-11    2500  (null)  (null)

18 rows selected.
```

2. Видаліть з таблиці 2 будь-які рядки.

**delete from My\_emp**

**where ename in ('JET LI', 'BRUCE LEE');**

**select \* from My\_emp;**

```
SQL> delete from My_emp
  2 where ename in ('JET LI', 'BRUCE LEE');

2 rows deleted.

SQL> select * from My_emp;

  EMPNO ENAME          JOB           MGR HIREDATE        SAL      COMM  DEPTNO
----- -----
    7839 KING            PRESIDENT   (null) 17-NOV-11     5000  (null)    10
    7698 BLAKE           MANAGER    7839 01-MAY-11    2850  (null)    30
    7782 CLARK           MANAGER    7839 09-JUN-11    1500  (null)    10
    7566 JONES           MANAGER    7839 02-APR-11    2975  (null)    20
    7654 MARTIN          SALESMAN   7698 28-SEP-11    1250   1400    30
    7499 ALLEN           SALESMAN   7698 20-FEB-11    1600    300    30
    7844 TURNER          SALESMAN   7698 08-SEP-11    1500     0    30
    7900 JAMES            CLERK     7698 03-DEC-11    950  (null)    30
    7521 WARD             SALESMAN   7698 22-FEB-11    1250    500    30
    7902 FORD              ANALYST   7566 03-DEC-11    3000  (null)    20
    7369 SMITH             CLERK     7902 17-DEC-10    800  (null)    20
    7788 SCOTT             ANALYST   7566 09-DEC-12    3000  (null)    20
    7876 ADAMS             CLERK     7788 12-JAN-13    1100  (null)    20
    7934 MILLER            CLERK     7782 23-JAN-12    1300  (null)    10
    8000 JACKIE CHAN       SALESMAN   7839 28-SEP-11    2250   1700    60
    8003 DR NO              ANALYST   7839 11-SEP-11    2500  (null)  (null)

16 rows selected.
```

3.

Відкотіться до початкового стану і перевірте чи повернулися зміни.

**ROLLBACK;**

**select \* from My\_emp;**

```
SQL> ROLLBACK;
Rollback complete.

SQL> select * from My_emp;

  EMPNO ENAME          JOB           MGR HIREDATE      SAL     COMM  DEPTNO
----- -----
    7839 KING            PRESIDENT   (null) 17-NOV-11   5000  (null)      10
    7698 BLAKE           MANAGER    7839 01-MAY-11   2850  (null)      30
    7782 CLARK           MANAGER    7839 09-JUN-11   1500  (null)      10
    7566 JONES           MANAGER    7839 02-APR-11   2975  (null)      20
    7654 MARTIN          SALESMAN   7698 28-SEP-11   1250    1400      30
    7499 ALLEN           SALESMAN   7698 20-FEB-11   1600     300      30
    7844 TURNER          SALESMAN   7698 08-SEP-11   1500      0      30
    7900 JAMES            CLERK     7698 03-DEC-11   950   (null)      30
    7521 WARD             SALESMAN   7698 22-FEB-11   1250     500      30
    7902 FORD             ANALYST   7566 03-DEC-11   3000  (null)      20
    7369 SMITH            CLERK     7902 17-DEC-10   800   (null)      20
    7788 SCOTT            ANALYST   7566 09-DEC-12   3000  (null)      20
    7876 ADAMS            CLERK     7788 12-JAN-13   1100  (null)      20
    7934 MILLER           CLERK     7782 23-JAN-12   1300  (null)      10
    8000 JACKIE CHAN      SALESMAN   7839 28-SEP-11   2250    1700      60
    8001 JET LI            SALESMAN   8000 20-FEB-11   2600     600      60
    8002 BRUCE LEE         SALESMAN   8000 08-SEP-11   2500  (null)      60
    8003 DR NO             ANALYST   7839 11-SEP-11   2500  (null)  (null)

18 rows selected.
```

4. Створіть мітку «START\_POINT».

**SAVEPOINT START\_POINT;**

```
SQL> SAVEPOINT START_POINT;
```

```
Savepoint created.
```

5. Додайте в таблицю 1 рядок.

```
INSERT INTO My_emp VALUES (1001,'Damon','ANALYST', 7839,
TO_DATE('2012-10-19', 'YYYY-MM-DD'), 2000, 0, 10);
```

```
SQL> INSERT INTO My_emp VALUES (1001,'Damon','ANALYST', 7839, TO_DATE('2012-10-19', 'YYYY-MM-DD'), 2000, 0, 10);
1 row created.
```

```
select * from My_emp;
```

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)
1001	Damon	ANALYST	7839	19-OCT-12	2000	0	10

```
19 rows selected.
```

6. Додайте в таблицю індекс по стовпцю mgr.

```
CREATE INDEX My_emp_index  
on My_emp(mgr);
```

```
SQL> CREATE INDEX My_emp_index  
2  on My_emp(mgr);  
  
Index created.
```

7. Чи можливий відкат до мітки START\_POINT? перевірте.

```
ROLLBACK TO START_POINT;
```

```
SQL> ROLLBACK TO START_POINT;  
ROLLBACK TO START_POINT  
*  
ERROR at line 1:  
ORA-01086: savepoint 'START_POINT' never established in this session or is invalid
```

6. Завдання. Відкат до мітки

1. Створіть мітку «START\_POINT1».

```
SAVEPOINT START_POINT1;
```

```
select * from My_emp;
```

```
SQL> SAVEPOINT START_POINT1;
```

Savepoint created.

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)
1001	Damon	ANALYST	7839	19-OCT-12	2000	0	10

19 rows selected.

2. Додайте до таблиці 1 рядок.

```
INSERT INTO My_emp VALUES (1002,'Alaric','MANAGER', 7499,  
TO_DATE('2012-06-24', 'YYYY-MM-DD'), 3000, 0, 20);
```

```
SQL> INSERT INTO My_emp VALUES (1002,'Alaric','MANAGER', 7499, TO_DATE('2012-06-24', 'YYYY-MM-DD'), 3000, 0, 20);  
1 row created.
```

```
select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)
1001	Damon	ANALYST	7839	19-OCT-12	2000	0	10
1002	Alaric	MANAGER	7499	24-JUN-12	3000	0	20

20 rows selected.

3. Створіть мітку «START\_POINT2».

**SAVEPOINT START\_POINT2;**

```
SQL> SAVEPOINT START_POINT2;
```

```
Savepoint created.
```

4. Додайте до таблиці 1 рядок.

```
INSERT INTO My_emp VALUES (1003,'Matt','CLERK', TO_DATE('2013-08-12', 'YYYY-MM-DD'), 780, 0, 20);
```

```
select * from My_emp;
```

```
SQL> INSERT INTO My_emp VALUES (1003,'Matt','CLERK', 1002, TO_DATE('2013-08-12', 'YYYY-MM-DD'), 780, 0, 20);
```

```
1 row created.
```

```
SQL> select * from My_emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)
1001	Damon	ANALYST	7839	19-OCT-12	2000	0	10
1002	Alaric	MANAGER	7499	24-JUN-12	3000	0	20
1003	Matt	CLERK	1002	12-AUG-13	780	0	20

```
21 rows selected.
```

5. Створіть мітку «START\_POINT3».

**SAVEPOINT START\_POINT3;**

```
SQL> SAVEPOINT START_POINT3;
```

```
Savepoint created.
```

6. Видаліть з таблиці 3 рядки.

```
delete from My_emp  
where empno in (1001, 1002, 1003);  
select * from My_emp;
```

```

SQL> delete from My_emp
2 where empno in (1001, 1002, 1003);

3 rows deleted.

```

```

SQL> select * from My_emp;

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)

18 rows selected.

7. Зробіть відкат до мітки START\_POINT2. Перевірте.

**ROLLBACK TO START\_POINT2;**

```

SQL> ROLLBACK TO START_POINT2;

Rollback complete.

```

**select \* from My\_emp;**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-NOV-11	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-MAY-11	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-11	1500	(null)	10
7566	JONES	MANAGER	7839	02-APR-11	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-11	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-11	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-11	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-11	950	(null)	30
7521	WARD	SALESMAN	7698	22-FEB-11	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-11	3000	(null)	20
7369	SMITH	CLERK	7902	17-DEC-10	800	(null)	20
7788	SCOTT	ANALYST	7566	09-DEC-12	3000	(null)	20
7876	ADAMS	CLERK	7788	12-JAN-13	1100	(null)	20
7934	MILLER	CLERK	7782	23-JAN-12	1300	(null)	10
8000	JACKIE CHAN	SALESMAN	7839	28-SEP-11	2250	1700	60
8001	JET LI	SALESMAN	8000	20-FEB-11	2600	600	60
8002	BRUCE LEE	SALESMAN	8000	08-SEP-11	2500	(null)	60
8003	DR NO	ANALYST	7839	11-SEP-11	2500	(null)	(null)
1001	Damon	ANALYST	7839	19-OCT-12	2000	0	10
1002	Alaric	MANAGER	7499	24-JUN-12	3000	0	20

20 rows selected.

8. Зробіть відкат до мітки START\_POINT1. Перевірте.

**ROLLBACK TO START\_POINT1;**

```
SQL> ROLLBACK TO START_POINT1;  
Rollback complete.
```

**select \* from My\_emp;**

```
SQL> select * from My_emp;  
  
EMPNO ENAME      JOB          MGR HIREDATE    SAL   COMM     DEPTNO  
----- --          --          --  --          --  --        --  
7839  KING        PRESIDENT  (null) 17-NOV-11  5000 (null) 10  
7698  BLAKE       MANAGER   7839  01-MAY-11  2850 (null) 30  
7782  CLARK       MANAGER   7839  09-JUN-11  1500 (null) 10  
7566  JONES       MANAGER   7839  02-APR-11  2975 (null) 20  
7654  MARTIN     SALESMAN  7698  28-SEP-11  1250  1400 30  
7499  ALLEN       SALESMAN  7698  20-FEB-11  1600   300 30  
7844  TURNER     SALESMAN  7698  08-SEP-11  1500     0 30  
7900  JAMES        CLERK    7698  03-DEC-11  950  (null) 30  
7521  WARD        SALESMAN  7698  22-FEB-11  1250   500 30  
7902  FORD         ANALYST  7566  03-DEC-11  3000 (null) 20  
7369  SMITH       CLERK    7902  17-DEC-10  800  (null) 20  
7788  SCOTT       ANALYST  7566  09-DEC-12  3000 (null) 20  
7876  ADAMS       CLERK    7788  12-JAN-13  1100 (null) 20  
7934  MILLER     CLERK    7782  23-JAN-12  1300 (null) 10  
8000  JACKIE CHAN SALESMAN  7839  28-SEP-11  2250  1700 60  
8001  JET LI      SALESMAN  8000  20-FEB-11  2600   600 60  
8002  BRUCE LEE   SALESMAN  8000  08-SEP-11  2500 (null) 60  
8003  DR NO       ANALYST  7839  11-SEP-11  2500 (null) (null) 10  
1001  Damon       ANALYST  7839  19-OCT-12  2000     0 10  
  
19 rows selected.
```

9. Видаліть з таблиці 3 рядки.

```
delete from My_emp  
where empno in (1001, 8001, 8002);  
select * from My_emp;
```

```
SQL> delete from My_emp  
  2 where empno in (1001, 8001, 8002);  
3 rows deleted.  
  
SQL> select * from My_emp;  
  
EMPNO ENAME      JOB          MGR HIREDATE    SAL   COMM     DEPTNO  
----- --          --          --  --          --  --        --  
7839  KING        PRESIDENT  (null) 17-NOV-11  5000 (null) 10  
7698  BLAKE       MANAGER   7839  01-MAY-11  2850 (null) 30  
7782  CLARK       MANAGER   7839  09-JUN-11  1500 (null) 10  
7566  JONES       MANAGER   7839  02-APR-11  2975 (null) 20  
7654  MARTIN     SALESMAN  7698  28-SEP-11  1250  1400 30  
7499  ALLEN       SALESMAN  7698  20-FEB-11  1600   300 30  
7844  TURNER     SALESMAN  7698  08-SEP-11  1500     0 30  
7900  JAMES        CLERK    7698  03-DEC-11  950  (null) 30  
7521  WARD        SALESMAN  7698  22-FEB-11  1250   500 30  
7902  FORD         ANALYST  7566  03-DEC-11  3000 (null) 20  
7369  SMITH       CLERK    7902  17-DEC-10  800  (null) 20  
7788  SCOTT       ANALYST  7566  09-DEC-12  3000 (null) 20  
7876  ADAMS       CLERK    7788  12-JAN-13  1100 (null) 20  
7934  MILLER     CLERK    7782  23-JAN-12  1300 (null) 10  
8000  JACKIE CHAN SALESMAN  7839  28-SEP-11  2250  1700 60  
8003  DR NO       ANALYST  7839  11-SEP-11  2500 (null) (null) 10  
  
16 rows selected.
```

10.  
Ств  
оріт  
ь  
міт

POINT4;

ку «START\_POINT4».

**SAVEPOINT START\_POINT4;**

11. Додайте до таблиці 1 рядок.

```
INSERT INTO My_emp VALUES (1004,'Tyler','CLERK', 8003, TO_DATE('2013-09-27', 'YYYY-MM-DD'), 640, 0, 20);
```

```
SQL> INSERT INTO My_emp VALUES (1004,'Tyler','CLERK', 8003, TO_DATE('2013-09-27', 'YYYY-MM-DD'), 640, 0, 20);
1 row created.
```

**select \* from My\_emp;**

```
SQL> select * from My_emp;
-----+-----+-----+-----+-----+-----+-----+
EMPNO|ENAME |JOB   |MGR   |HIREDATE|SAL    |COMM   |DEPTNO|
-----+-----+-----+-----+-----+-----+-----+
7839 |KING  |PRESIDENT|(null) |17-NOV-11|5000  |(null)  |10
7698 |BLAKE |MANAGER  |7839  |01-MAY-11|2850  |(null)  |30
7782 |CLARK |MANAGER  |7839  |09-JUN-11|1500  |(null)  |10
7566 |JONES |MANAGER  |7839  |02-APR-11|2975  |(null)  |20
7654 |MARTIN|SALESMAN |7698  |28-SEP-11|1250   |1400   |30
7499 |ALLEN |SALESMAN |7698  |20-FEB-11|1600   |300    |30
7844 |TURNER|SALESMAN |7698  |08-SEP-11|1500   |0      |30
7900 |JAMES  |CLERK   |7698  |03-DEC-11|950    |(null)  |30
7521 |WARD   |SALESMAN |7698  |22-FEB-11|1250   |500    |30
7902 |FORD   |ANALYST |7566  |03-DEC-11|3000  |(null)  |20
7369 |SMITH  |CLERK   |7902  |17-DEC-10|800   |(null)  |20
7788 |SCOTT  |ANALYST |7566  |09-DEC-12|3000  |(null)  |20
7876 |ADAMS  |CLERK   |7788  |12-JAN-13|1100  |(null)  |20
7934 |MILLER |CLERK   |7782  |23-JAN-12|1300  |(null)  |10
8000 |JACKIE CHAN|SALESMAN |7839  |28-SEP-11|2250  |1700   |60
8003 |DR NO   |ANALYST |7839  |11-SEP-11|2500  |(null)  |(null)
1004 |Tyler  |CLERK   |8003  |27-SEP-13|640   |0      |20
-----+-----+-----+-----+-----+-----+-----+
17 rows selected.
```

12. Застосуйте зміни.

**commit;**

```
SQL> commit;
Commit complete.
```

13. Чи можливий відкат до мітки START\_POINT4? Перевірте.

**ROLLBACK TO START\_POINT4;**

```
SQL> ROLLBACK TO START_POINT4;
ROLLBACK TO START_POINT4
*
ERROR at line 1:
ORA-01086: savepoint 'START_POINT4' never established in this session or is invalid
```