Пороскун Олена Олегівна ПМ-81

Лабораторна робота 4

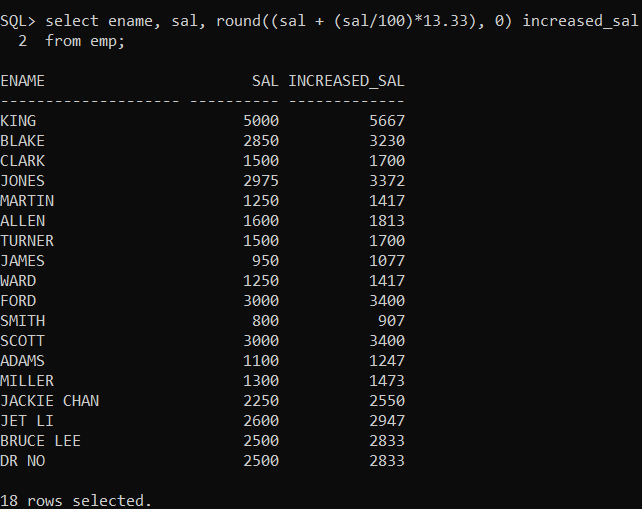


Робота з числами

1. Відділу кадрів потрібен звіт, який містить ім'я співробітника, поточну зарплату і зарплату збільшену 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;**

****

2. Порівняйте результати, які повертають запити:

1. SELECT TRUNC(55.52,2) FROM DUAL;

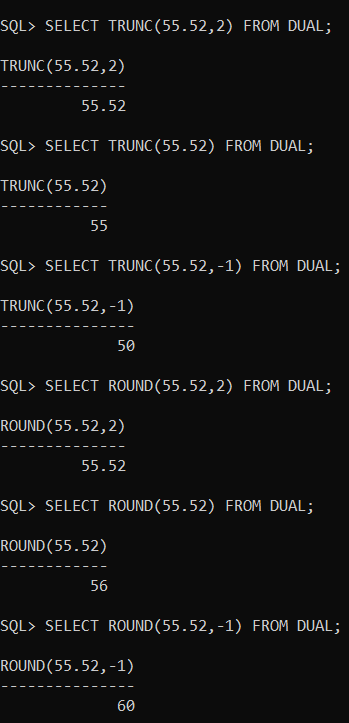
2. SELECT TRUNC(55.52) FROM DUAL;

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;

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:

1. сотих;
2. цілого числа;
3. 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», «С» або «J». Дайте стовпцю з довжиною імені псевдонім.

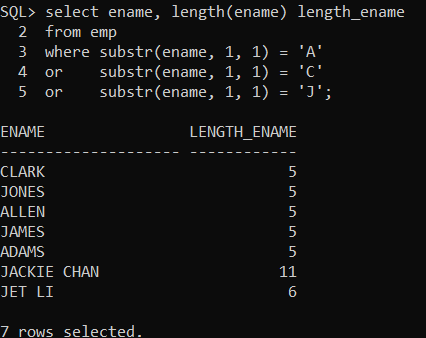
**select ename, length(ename) length\_ename**

**from emp**

**where substr(ename, 1, 1) = 'A'**

**or substr(ename, 1, 1) = 'C'**

**or substr(ename, 1, 1) = 'J';**

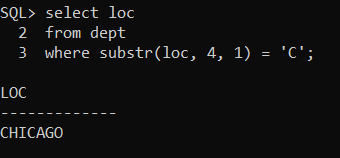
****

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

**select loc**

**from dept**

**where substr(loc, 4, 1) = 'C';**

****

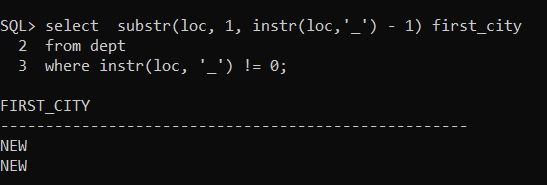
5. **\*** Виведіть перші слова в назвах міст, якщо назва міста складається з 2-х слів. (Назви зберігаються в стовбці loc таблиці Dept)

* 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;**

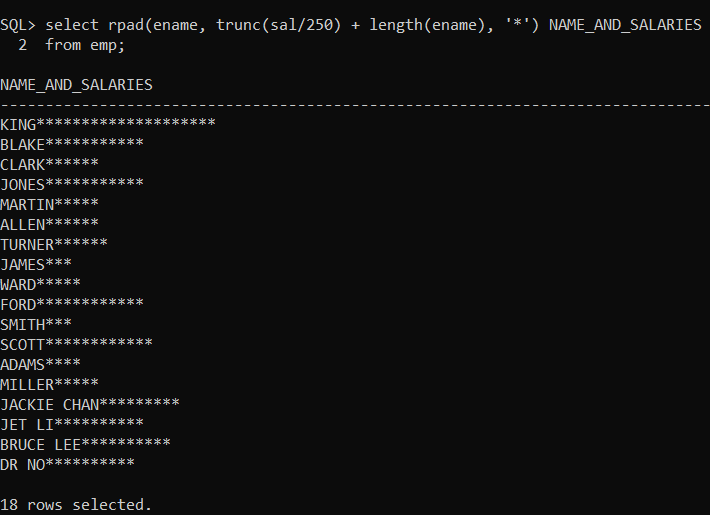
****

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;**

****

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).

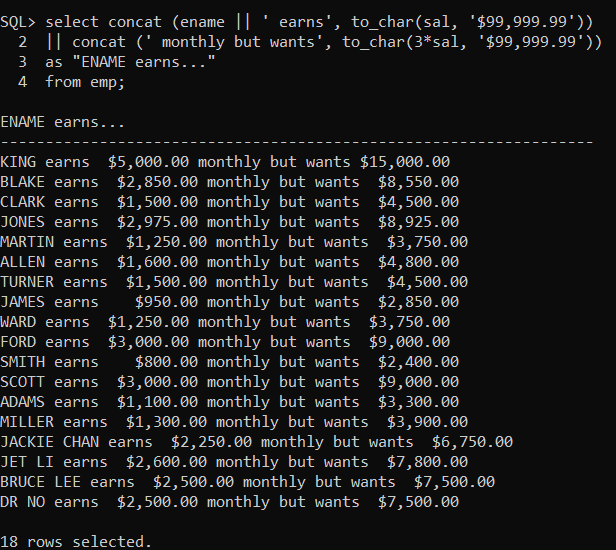
Для об'єднання рядків скористайтеся функцією.

**select concat (ename || ' earns', to\_char(sal, '$99,999.99'))**

**|| concat (' monthly but wants', to\_char(3\*sal, '$99,999.99'))**

**as "ENAME earns..."**

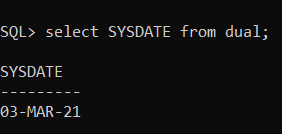
**from emp;**



Робота з датами

8. Напишіть запит, який поверне поточну дату.

**select SYSDATE from dual;**

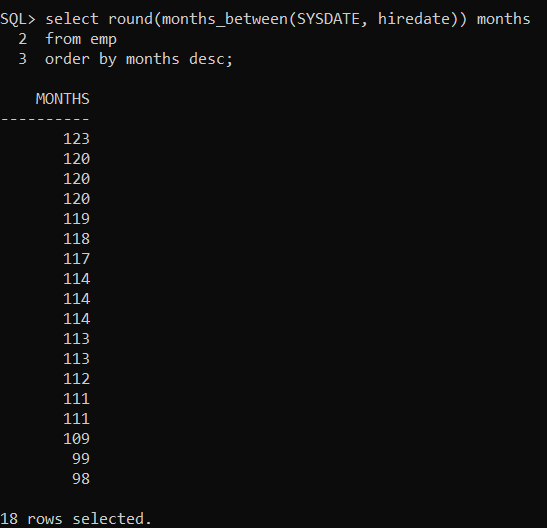


9. \* Сформувати запит, що виводить в порядку спадання кількість місяців, що минули з моменту найму кожного працівника. Кількість місяців округлити до найближчого цілого.

**select round(months\_between(SYSDATE, hiredate)) months**

**from emp**

**order by months desc;**

****

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.

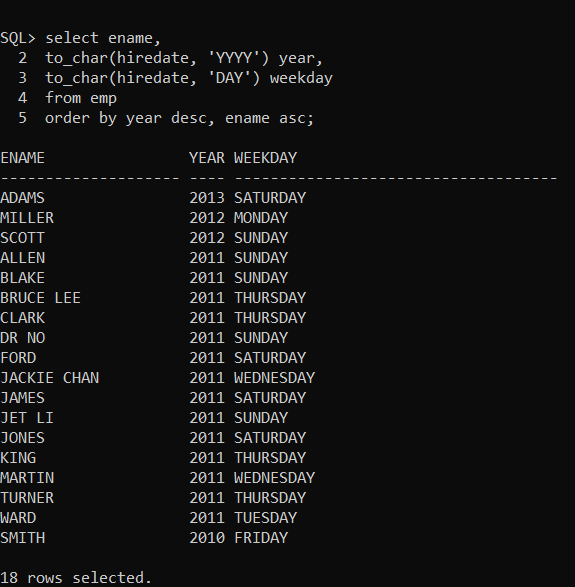
**select ename,**

**to\_char(hiredate, 'YYYY') year,**

**to\_char(hiredate, 'DAY') weekday**

**from emp**

**order by years desc, ename asc;**

****

Значення Null

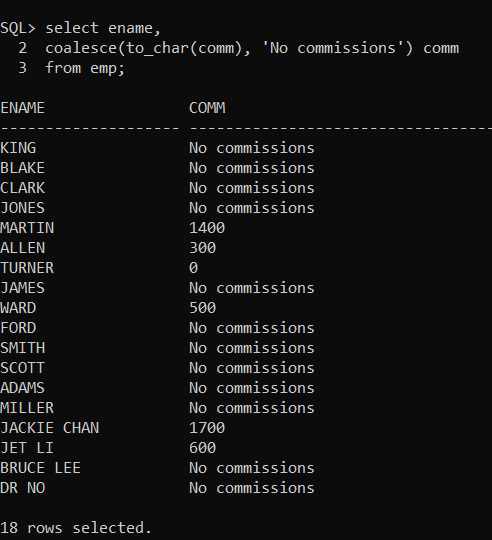
11. Сформувати запит, що виводить прізвище службовця і його комісійні. У разі відсутності комісійних в стовпці має бути виведено "No commissions".

**select ename,**

**coalesce(to\_char(comm),**

**'No commissions') comm**

**from emp;**

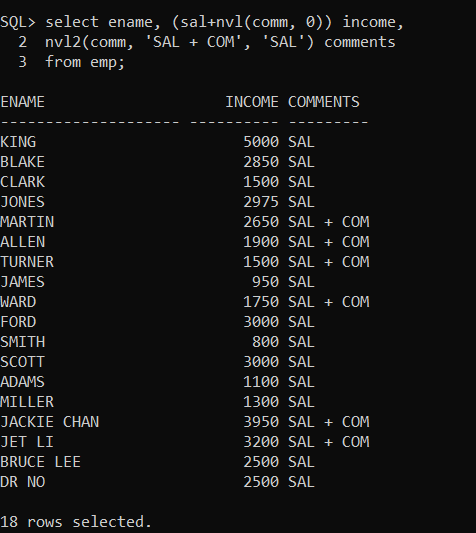


12. Створіть звіт, який містить ім'я співробітника, заробіток (зарплата + премія) та коментар. Поле коментар містить значення «SAL + COM», якщо у співробітника є премія, або «SAL», якщо премія не встановлена.

**select ename, (sal+nvl(comm, 0)) income,**

**nvl2(comm, 'SAL + COM', 'SAL') comments**

**from emp;**

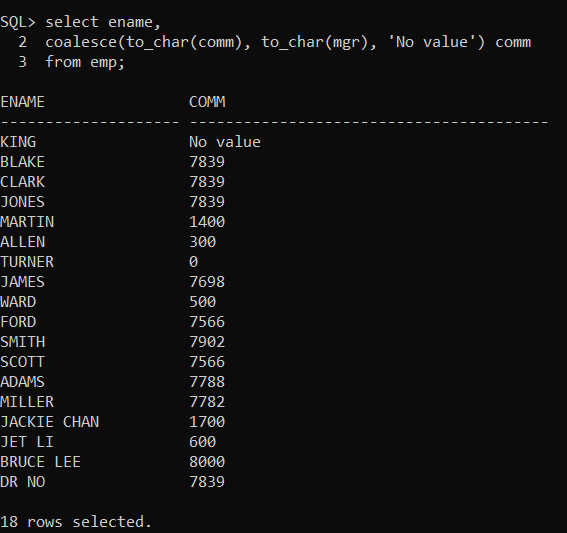
****

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;**

****

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**

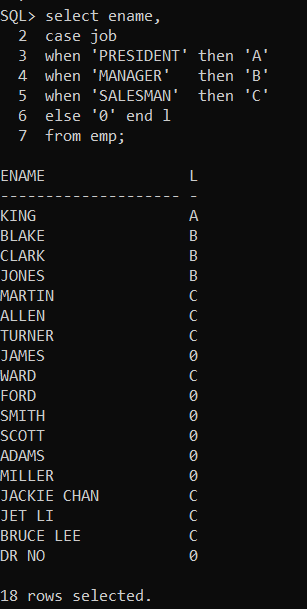
**when 'PRESIDENT' then 'A'**

**when 'MANAGER' then 'B'**

**when 'SALESMAN' then 'C'**

**else '0' end l**

**from emp;**

****

15. \* Реалізуйте попереднє завдання за допомогою DECODE.

**select ename,**

**decode (job,**

**'PRESIDENT', 'A',**

**'MANAGER', 'B',**

**'SALESMAN', 'C',**

**'0') l**

**from emp;**

****

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 може стати 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');**

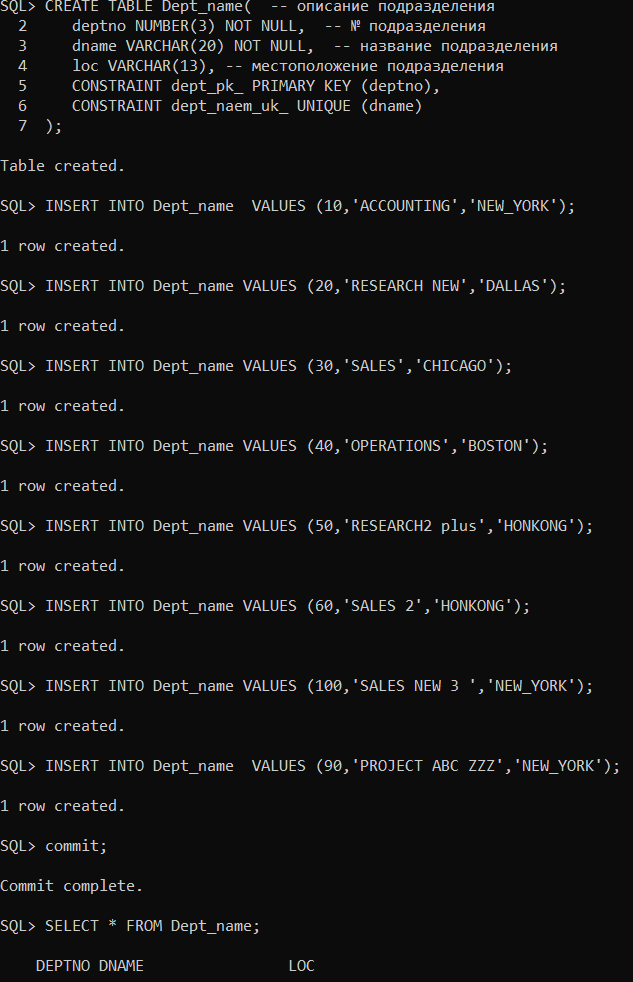
**INSERT INTO Dept\_name VALUES (100,'SALES NEW 3 ','NEW\_YORK');**

**INSERT INTO Dept\_name VALUES (90,'PROJECT ABC ZZZ','NEW\_YORK');**

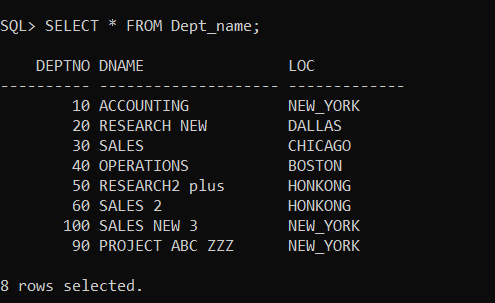
**commit;**

**SELECT \* FROM Dept\_name;**

**/**

****

*Отримали наступне:*

****

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,**

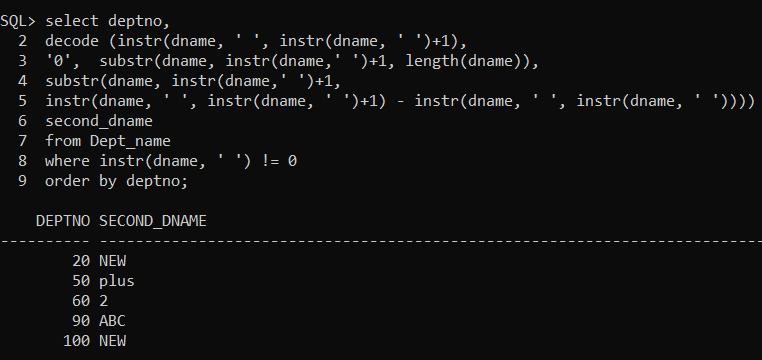
**instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' '))))**

**second\_dname**

**from Dept\_name**

**where instr(dname, ' ') != 0**

**order by deptno;**

****

*Перевірка результату:*

**select deptno, dname,**

**decode (instr(dname, ' ', instr(dname, ' ')+1),**

**'0', substr(dname, instr(dname,' ')+1, length(dname)),**

**substr(dname, instr(dname,' ')+1,**

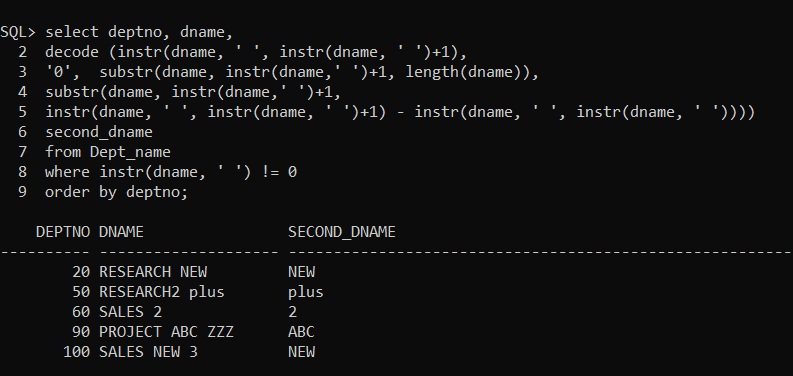
**instr(dname, ' ', instr(dname, ' ')+1) - instr(dname, ' ', instr(dname, ' '))))**

**second\_dname**

**from Dept\_name**

**where instr(dname, ' ') != 0**

**order by deptno;**

****