Witaj w pracy domowej w której udoskonalisz umiejętności związane z pracą z kursorami.

PS. Nie zapomnij przywitać się na grupie i oznaczyć pracy domowej hasztagiem #MODUŁ7.

Jeśli masz problem z którymś z zadań i chcesz bym sprawdził Twoją pracę domową to:

* umieść pracę domową na dysku google(tak, bym nie musiał pobierać pliku)
* oznacz mnie w poście i napisz z czym masz konkretnie problem

Nie ma znaczenia czy Twój plik ma rozszerzenie .docx czy .sql. Zrób jak   
Ci wygodniej :)

# **ZADANIE 1 : ZADANIA DO SAMODZIELNEGO WYKONANIA.**

Uwaga. Wszystkie zadania należy wykonywać będąc zalogowanym jako użytkownik kurs\_plsql, chyba, że wskazano inaczej.

1. Stwórz kursor jawny, który w pętli WHILE wyświetli wszystkie rekordy z tabeli REGIONS.

DECLARE

CURSOR cursor\_regions IS

SELECT \* FROM regions;

v\_regions cursor\_regions%ROWTYPE;

v\_found BOOLEAN:= TRUE;

BEGIN

OPEN cursor\_regions;

WHILE v\_found

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_regions.region\_id||' '||v\_regions.region\_name);

FETCH cursor\_regions INTO v\_regions;

v\_found:=cursor\_regions%FOUND;

END LOOP;

CLOSE cursor\_regions;

end;

1. Przepisz program z zadania 1 na program korzystający z pętli FOR kursorowej i kursora niejawnego.

BEGIN

FOR i IN (SELECT \*

FROM regions)

LOOP

DBMS\_OUTPUT.PUT\_LINE(i.region\_id||' '||i.region\_name);

END LOOP;

END;

1. Wyjaśnij czym różni się sql%rowcount od cursor\_name%rowcount

sql%rowcount - atrybut kursora niejawnego; zwraca ile łącznie rekordów polecenia DML (np. SELECT) przetworzył

cursor\_name%rowcount - atrybut kursora jawnego; zwraca ile rekordów przetworzył do tej pory

1. Sprawdź przy pomocy %rowcount czy w bazie danych istnieją departamenty, których managerowie zarabiają mniej niż 8000.

wersja1.

DECLARE

v\_rowcount NUMBER;

CURSOR c\_departments IS

SELECT d.department\_id, d.department\_name, d.manager\_id, e.salary

FROM departments d

INNER JOIN employees e

ON d.manager\_id=e.employee\_id

WHERE e.salary<8000;

BEGIN

FOR i IN c\_departments

LOOP

v\_rowcount:=c\_departments%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE ('id departamentu: '||i.department\_id||', '||

'nazwa departamentu: '||i.department\_name||', ' || 'id managera: '||i.manager\_id);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('Liczba departamentów, w których managerowie zarabiają mniej niż 8000 to: '||v\_rowcount);

END;

wersja 2

DECLARE

CURSOR curr\_emp IS SELECT department\_id

FROM employees

WHERE job\_id LIKE '%MAN' AND SALARY<8000;

v\_dep curr\_emp%ROWTYPE;

v\_rowcount NUMBER;

BEGIN

OPEN curr\_emp;

LOOP

FETCH curr\_emp INTO v\_dep;

EXIT WHEN curr\_emp%NOTFOUND;

v\_rowcount:=curr\_emp%rowcount;

END LOOP;

CLOSE curr\_emp;

IF v\_rowcount>0 THEN

DBMS\_OUTPUT.PUT\_LINE('Istnieją departamenty, gdzie menadżerowie zarabiają poniżej 8000');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Nie istnieją departamenty, gdzie menadżerowie zarabiają poniżej 8000');

END IF;

END;

1. Napisz jawny kursor z parametrem, który będzie zwracał listę pracowników z podanego w parametrze nazwy departamentu(department\_name).

–ver. 1

DECLARE

CURSOR c\_employee(in\_dep\_name VARCHAR2) IS

SELECT e.first\_name, e.last\_name, e.department\_id, e.salary

FROM employees e

INNER JOIN departments d

ON d.department\_id=e.department\_id

WHERE d.department\_name=in\_dep\_name;

v\_employee c\_employee%ROWTYPE;

BEGIN

OPEN c\_employee('Sales');

LOOP

FETCH c\_employee INTO v\_employee;

EXIT WHEN c\_employee%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (v\_employee.first\_name|| ', '||v\_employee.last\_name||

', '|| v\_employee.department\_id);

END LOOP;

CLOSE c\_employee;

END;

—ver.2

DECLARE

CURSOR c\_emp(in\_dep\_name VARCHAR2) IS

SELECT a.first\_name,a.last\_name

FROM employees a

INNER JOIN departments b ON a.department\_id=b.department\_id

WHERE b.department\_name = in\_dep\_name;

v\_emp\_rec c\_emp%ROWTYPE;

BEGIN

OPEN c\_emp('Purchasing');

LOOP

FETCH c\_emp INTO v\_emp\_rec;

EXIT WHEN c\_emp%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_emp\_rec.first\_name||', '||v\_emp\_rec.last\_name);

END LOOP;

CLOSE c\_emp;

END;

1. Rozwiń program z punktu 5 o:
   * podniesienie pensji pracowników o 20%
   * oczekiwanie na blokowanie rekordów przez maksymalnie 6 sekund
   * blokowanie wyłącznie rekordów z tabeli employees
   * wykorzystanie where current of

–v1

DECLARE

CURSOR c\_employee(in\_dep\_name VARCHAR2) IS

SELECT e.first\_name, e.last\_name, e.department\_id, e.salary

FROM employees e

INNER JOIN departments d

ON d.department\_id=e.department\_id

WHERE d.department\_name=in\_dep\_name

FOR UPDATE OF e.salary WAIT 6;

v\_employee c\_employee%ROWTYPE;

BEGIN

OPEN c\_employee('Sales');

LOOP

FETCH c\_employee INTO v\_employee;

EXIT WHEN c\_employee%NOTFOUND;

UPDATE employees

SET salary=salary\*1.2

WHERE CURRENT OF c\_employee;

END LOOP;

CLOSE c\_employee;

END;

–v.2

DECLARE

CURSOR c\_emp(in\_dep\_name VARCHAR2) IS

SELECT a.first\_name,a.last\_name,a.salary

FROM employees a

INNER JOIN departments b ON a.department\_id=b.department\_id

WHERE b.department\_name = in\_dep\_name

FOR UPDATE OF a.first\_name WAIT 6;

v\_emp\_rec c\_emp%ROWTYPE;

BEGIN

OPEN c\_emp('Purchasing');

LOOP

FETCH c\_emp INTO v\_emp\_rec;

EXIT WHEN c\_emp%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_emp\_rec.first\_name||', '||v\_emp\_rec.last\_name);

UPDATE employees

SET salary = salary\*1.2

WHERE CURRENT OF c\_emp;

END LOOP;

CLOSE c\_emp;

END;