**CSE2007 DBMS LAB**

**SLOT: L39+L40**

**NAME - AMAN SAHU**

**REG. NO – 22BCE7224**

**EXPERIMENT NO.-9**

1.Write a PL/SQL program to find the sum of digits in a given number.

DECLARE

num NUMBER := 12345;

total\_sum NUMBER := 0;

digit NUMBER;

BEGIN

WHILE num > 0 LOOP

digit := MOD(num, 10);

total\_sum := total\_sum + digit;

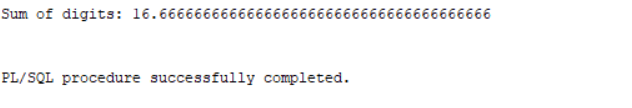
num := num / 10;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Sum of digits: ' || total\_sum);

END;

/



2. Write a PL / SQL program to check whether the given number is prime or not.

DECLARE

num NUMBER := 7;

isPrime BOOLEAN := TRUE;

i NUMBER := 2;

BEGIN

IF num <= 1 THEN

isPrime := FALSE;

ELSE

WHILE i <= SQRT(num) LOOP

IF MOD(num, i) = 0 THEN

isPrime := FALSE;

EXIT;

END IF;

i := i + 1;

END LOOP;

END IF;

IF isPrime THEN

DBMS\_OUTPUT.PUT\_LINE(num || ' is a prime number.');

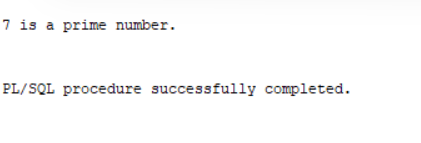
ELSE

DBMS\_OUTPUT.PUT\_LINE(num || ' is not a prime number.');

END IF;

END;

/



3. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying

from 3 to 7. Store the radius and the corresponding values of calculated area in an empty

table named areas, consisting of two columns radius and area.

CREATE TABLE c\_areas (

radius NUMBER,

area NUMBER

);

DECLARE

radius\_val NUMBER;

area\_val NUMBER;

BEGIN

FOR radius\_val IN 3..7 LOOP

area\_val := 3.14 \* radius\_val \* radius\_val;

INSERT INTO c\_areas VALUES (radius\_val, area\_val);

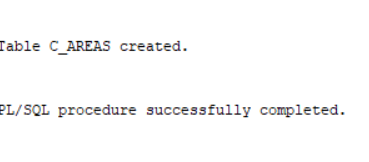
END LOOP;

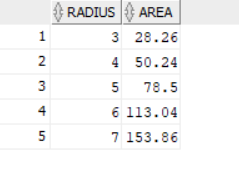
COMMIT;

END;

/

SELECT \* From c\_areas;





4. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is

less than or equal to 10. Else display an error message. Otherwise Display the remainder

in words.

DECLARE

v\_number NUMBER := 11;

v\_divisor NUMBER := 6;

v\_remainder NUMBER;

v\_remainder\_word VARCHAR2(20);

BEGIN

IF v\_divisor > 10 THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Divisor should be less than or equal to 10');

ELSE

v\_remainder := MOD(v\_number, v\_divisor);

CASE v\_remainder

WHEN 0 THEN v\_remainder\_word := 'Zero';

WHEN 1 THEN v\_remainder\_word := 'One';

WHEN 2 THEN v\_remainder\_word := 'Two';

WHEN 3 THEN v\_remainder\_word := 'Three';

WHEN 4 THEN v\_remainder\_word := 'Four';

WHEN 5 THEN v\_remainder\_word := 'Five';

WHEN 6 THEN v\_remainder\_word := 'Six';

WHEN 7 THEN v\_remainder\_word := 'Seven';

WHEN 8 THEN v\_remainder\_word := 'Eight';

WHEN 9 THEN v\_remainder\_word := 'Nine';

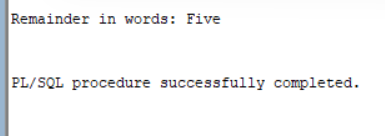
END CASE;

DBMS\_OUTPUT.PUT\_LINE('Remainder in words: ' || v\_remainder\_word);

END IF;

END;

/



5. Write a PL/SQL block that will display the name, dept no, salary of highest paid

employees.

set serveroutput on

DECLARE

v\_ename emp.ename%TYPE;

v\_sal emp.sal%TYPE;

v\_deptno emp.deptno%TYPE;

v\_maxsal emp.sal%TYPE;

BEGIN

SELECT MAX(sal) INTO v\_maxsal FROM emp;

FOR cur IN (SELECT ename, sal, deptno FROM emp WHERE sal = v\_maxsal)

LOOP

v\_ename := cur.ename;

v\_sal := cur.sal;

v\_deptno := cur.deptno;

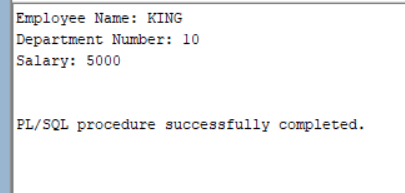
DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || v\_ename);

DBMS\_OUTPUT.PUT\_LINE('Department Number: ' || v\_deptno);

DBMS\_OUTPUT.PUT\_LINE('Salary: ' || v\_sal);

END LOOP;

END;



6. Write a PL/SQL block to update the salaries of employees of department number is 30

by 10 percent and display the number of records is updated.

DECLARE

v\_cnt NUMBER(4);

BEGIN

UPDATE emp

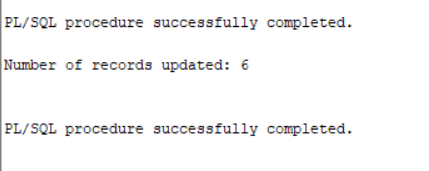
SET sal = sal \* 1.10

WHERE deptno = 30;

SELECT COUNT(\*) INTO v\_cnt FROM emp WHERE deptno = 30;

DBMS\_OUTPUT.PUT\_LINE('Number of records updated: ' || v\_cnt);

END;



7. Write a PL/SQL block to update the salaries of employees by ‘K’ percent of specific

department mentioned by user input. Hint: Parameterized Cursors.

DECLARE

-- Declare variables

v\_deptno emp.DEPTNO%TYPE;

v\_percent NUMBER(5,2); -- Percentage increase/decrease

BEGIN

-- Accept user input for department number and percentage

v\_deptno := &deptno; -- User input for department number

v\_percent := &percent; -- User input for percentage

-- Open cursor to fetch employees in the specified department

FOR emp\_rec IN (SELECT \* FROM emp WHERE DEPTNO = v\_deptno) LOOP

-- Update salary with the specified percentage

UPDATE emp

SET SAL = SAL \* (1 + v\_percent / 100)

WHERE EMPNO = emp\_rec.EMPNO;

END LOOP;

-- Commit the changes

COMMIT;

-- Display success message

DBMS\_OUTPUT.PUT\_LINE('Salaries updated successfully for department ' || v\_deptno || ' by ' || v\_percent || '%');

EXCEPTION

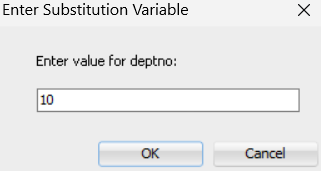
WHEN OTHERS THEN

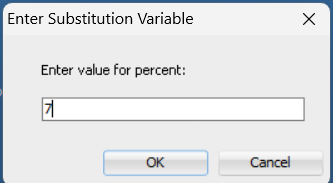
-- Display error message if any exception occurs

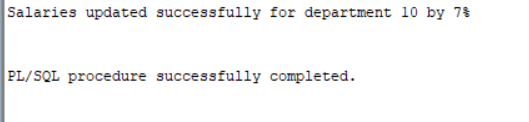
DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

ROLLBACK; -- Rollback changes if an error occurs

END;







8. Write a PL/SQL block to display the employee records of specific department using

Cursor for loop.

DECLARE

DEPT\_ID NUMBER := 10; -- You can change this to the department number you want

CURSOR emp\_cur IS SELECT \* FROM emp WHERE DEPTNO = DEPT\_ID;

BEGIN

FOR emp\_rec IN emp\_cur LOOP

DBMS\_OUTPUT.PUT\_LINE('Employee Number: ' || emp\_rec.EMPNO);

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || emp\_rec.ENAME);

DBMS\_OUTPUT.PUT\_LINE('Job: ' || emp\_rec.JOB);

DBMS\_OUTPUT.PUT\_LINE('Manager: ' || emp\_rec.MGR);

DBMS\_OUTPUT.PUT\_LINE('Hire Date: ' || emp\_rec.HIREDATE);

DBMS\_OUTPUT.PUT\_LINE('Salary: ' || emp\_rec.SAL);

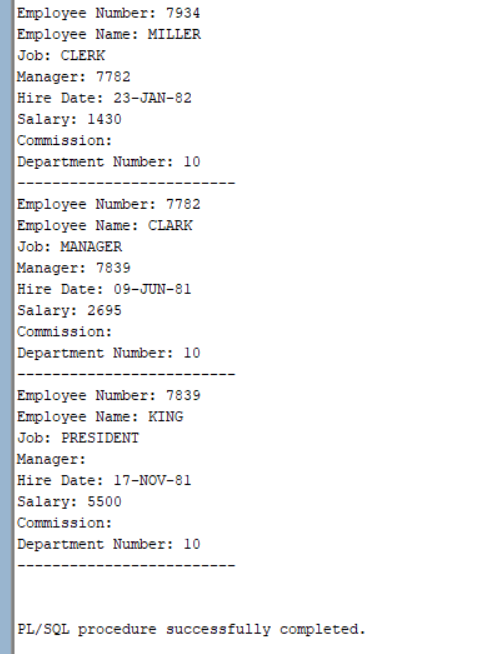
DBMS\_OUTPUT.PUT\_LINE('Commission: ' || emp\_rec.COMM);

DBMS\_OUTPUT.PUT\_LINE('Department Number: ' || emp\_rec.DEPTNO);

DBMS\_OUTPUT.PUT\_LINE('-------------------------');

END LOOP;

END;



9. Write a PL/SQL block to display all employees and their department names using

Cursors.

DECLARE

CURSOR cur\_emp IS

SELECT e.ename, d.dname

FROM emp e

JOIN dept d ON e.deptno = d.deptno;

v\_ename emp.ename%TYPE;

v\_dname dept.dname%TYPE;

BEGIN

FOR cur IN cur\_emp

LOOP

v\_ename := cur.ename;

v\_dname := cur.dname;

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || v\_ename);

DBMS\_OUTPUT.PUT\_LINE('Department Name: ' || v\_dname);

END LOOP;

END;

