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;
/
```

```
Sum of digits: 16.666666666666666666666666666666666666666
```

```
PL/SQL procedure successfully completed.
```

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;
/
```

```
7 is a prime number.


PL/SQL procedure successfully completed.
```

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;
```

```
Table C_AREAS created.


PL/SQL procedure successfully completed.
```

	RADIUS	AREA
1	3	28.26
2	4	50.24
3	5	78.5
4	6	113.04
5	7	153.86

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;
/
```

```
Remainder in words: Five


PL/SQL procedure successfully completed.
```

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;
```

```
Employee Name: KING
Department Number: 10
Salary: 5000


PL/SQL procedure successfully completed.
```

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.

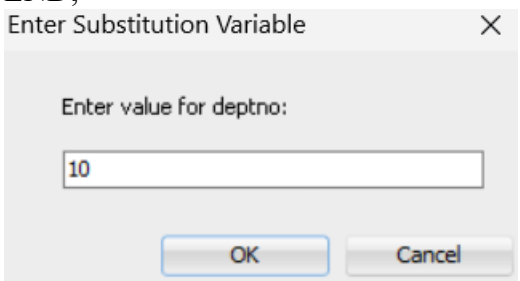
```
DECLARE
  v_cnt NUMBER(4);
BEGIN
  UPDATE emp
  SET sal = sal * 1.10
  WHERE deptno = 30;
  SELECT COUNT(*) INTO v_cnt FROM emp WHERE deptno = 30;
  DBMS_OUTPUT.PUT_LINE('Number of records updated: ' || v_cnt);
END;
```

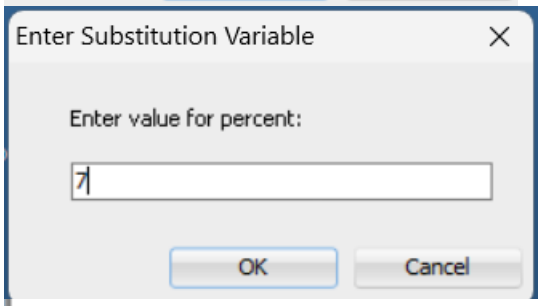
```
PL/SQL procedure successfully completed.

Number of records updated: 6


PL/SQL procedure successfully completed.
```

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;
```






```
Salaries updated successfully for department 10 by 7%


PL/SQL procedure successfully completed.
```

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;
```

```
Employee Number: 7934
Employee Name: MILLER
Job: CLERK
Manager: 7782
Hire Date: 23-JAN-82
Salary: 1430
Commission:
Department Number: 10
-------------------------
Employee Number: 7782
Employee Name: CLARK
Job: MANAGER
Manager: 7839
Hire Date: 09-JUN-81
Salary: 2695
Commission:
Department Number: 10
-------------------------
Employee Number: 7839
Employee Name: KING
Job: PRESIDENT
Manager:
Hire Date: 17-NOV-81
Salary: 5500
Commission:
Department Number: 10
-------------------------


PL/SQL procedure successfully completed.
```

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;
```

```
Employee Name: SMITH
Department Name: RESEARCH
Employee Name: ALLEN
Department Name: SALES
Employee Name: WARD
Department Name: SALES
Employee Name: JONES
Department Name: RESEARCH
Employee Name: MARTIN
Department Name: SALES
Employee Name: BLAKE
Department Name: SALES
Employee Name: CLARK
Department Name: ACCOUNTING
Employee Name: SCOTT
Department Name: RESEARCH
Employee Name: KING
Department Name: ACCOUNTING
Employee Name: TURNER
Department Name: SALES
Employee Name: ADAMS
Department Name: RESEARCH
Employee Name: JAMES
Department Name: SALES
Employee Name: FORD
Department Name: RESEARCH
Employee Name: MILLER
Department Name: ACCOUNTING


PL/SQL procedure successfully completed.
```