Task 01: Loop, Record, Exception [2 Marks]

1. Display a message *“Avengers Assemble”* 6 times using a loop. (0.5)

Begin

For counter in 1..6 loop dbms\_output.put\_line('Avengers Assemble');

end loop;

END;

1. Using records display the name and job of employee with emp no 7536.

Your output should be *“King works as a President”.* (0.5)

DECLARE

EMP\_RECORD EMP%ROWTYPE;

BEGIN

SELECT \* INTO EMP\_RECORD FROM EMP WHERE EMPNO=7839;

DBMS\_OUTPUT.PUT\_LINE(EMP\_RECORD.ENAME || ' WORKS AS A '||EMP\_RECORD.JOB); END;

1. Implement exception handling for multiple rows and no data by displaying message as *“No data found”* or *“Multiple rows returned”* using if conditions. (1)

DECLARE id emp.empno%type; name emp.ename%type;

BEGIN

SELECT empno, ename INTO id, name FROM emp;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No Data Found');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE ('Multiple rows returned'); END;

Task 02: Cursor [5 Marks]

4. Using cursor display employee data who work as CLERKS in the form

*“SMITH is a CLERK”*. (1)

DECLARE

C\_ENAME EMP.ENAME%TYPE;

C\_JOB EMP.JOB%TYPE;

CURSOR EMP\_CURSOR IS SELECT ENAME,JOB FROM EMP WHERE JOB =

'CLERK';

BEGIN

OPEN EMP\_CURSOR;

LOOP

FETCH EMP\_CURSOR INTO C\_ENAME,C\_JOB;

EXIT WHEN EMP\_CURSOR%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (C\_ENAME||' IS A '||C\_JOB);

END LOOP;

CLOSE EMP\_CURSOR;

END;

5. Using cursor display DEPT table data. Remember that DEPT table has three columns DEPTNO, DNAME and LOC. Data must be displayed in format “Deptno: 10, Dname: ACCOUNTING, Loc: NEW YORK”. (2)

DECLARE

D\_DEPTNO DEPT.DEPTNO%TYPE;

D\_NAME DEPT.DNAME%TYPE;

D\_LOC DEPT.LOC%TYPE;

CURSOR DEPT\_CURSOR IS SELECT DEPTNO, DNAME, LOC FROM DEPT;

BEGIN

OPEN DEPT\_CURSOR;

LOOP

FETCH DEPT\_CURSOR INTO D\_DEPTNO, D\_NAME, D\_LOC;

EXIT WHEN DEPT\_CURSOR%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE ('DEPTNO: '||D\_DEPTNO||', DNAME:

'||D\_NAME||', LOC: '||D\_LOC);

END LOOP;

CLOSE DEPT\_CURSOR;

END;

6. Using cursor display SALGRADE table data. Remember that SALEGRADE table has three columns GRADE, HISAL and LOSAL. Data must be

displayed in format “1->1200->700”. (2)

DECLARE

G\_GRADE SALGRADE.GRADE%TYPE;

G\_HISAL SALGRADE.HISAL%TYPE;

G\_LOSAL SALGRADE.LOSAL%TYPE;

CURSOR SAL\_CURSOR IS SELECT GRADE, HISAL, LOSAL FROM SALGRADE;

BEGIN

OPEN SAL\_CURSOR;

LOOP

FETCH SAL\_CURSOR INTO G\_GRADE, G\_HISAL, G\_LOSAL;

EXIT WHEN SAL\_CURSOR%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(G\_GRADE||'->'||G\_HISAL||'->'||G\_LOSAL);

END LOOP;

CLOSE SAL\_CURSOR;

END;

Task 03: Procedure, Function [13 Marks]

1. Create a procedure named ADD which takes two numbers and displays their sum. (2)

DECLARE

* 1. NUMBER;
  2. NUMBER;
  3. NUMBER;

PROCEDURE ADD(A IN NUMBER, B IN NUMBER) IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE (A + B);

END;

BEGIN

A:= 23;

B:= 45;

ADD(A, B);

END;

1. The above procedure displays the sum. Alter the above procedure in a way that it returns the sum as output in a variable. Use the output variable and display it yourself. (2)

DECLARE

* 1. NUMBER;
  2. NUMBER;
  3. NUMBER;

PROCEDURE ADD(A IN NUMBER, B IN NUMBER, C OUT NUMBER) IS

BEGIN

C := A + B;

END;

BEGIN

A:= 23;

B:= 45;

ADD(A, B, C);

DBMS\_OUTPUT.PUT\_LINE(C);

END;

1. Create a PL/SQL procedure SALARY\_SHEET that receives *empno* as a parameter and prepares the salary sheet of an employee as per the following rules:
   * Medical Allowance: 10% of the salary
   * House Rent: 25% of the salary
   * Conveyance Allowance: 15% of the salary
   * Tax Deductions: 12% of the salary

The procedure should output the employee’s name, job, computed allowances, tax deductions, and net salary(salary after adding all allowances and subtracting Tax) all values in different variables.

*You need to make variables to hold all these values* (5)

DECLARE

NO EMP.EMPNO%TYPE;

NAME EMP.ENAME%TYPE;

J EMP.JOB%TYPE;

SALARY EMP.SAL%TYPE;

MA NUMBER;

HR NUMBER;

CA NUMBER; TD NUMBER;

PROCEDURE SALARY\_SHEET(NO IN EMP.EMPNO%TYPE, NAME OUT

EMP.ENAME%TYPE, J OUT EMP.JOB%TYPE, SALARY OUT EMP.SAL%TYPE,

MA OUT NUMBER, HR OUT NUMBER, CA OUT NUMBER, TD OUT

NUMBER) IS

BEGIN

SELECT ENAME, JOB, SAL INTO NAME, J, SALARY FROM EMP WHERE

EMPNO = NO;

MA:= (SALARY\*0.1);

HR:= (SALARY\*0.25);

CA:= (SALARY\*0.15); TD:= (SALARY\*0.12);

END;

BEGIN

NO:= 7839;

SALARY\_SHEET(NO, NAME, J, SALARY, MA, HR, CA, TD);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE NAME: '||NAME);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE JOB: '||J);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE SALARY: '||SALARY);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE MEDICAL ALLOWANCE: '||MA);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE HOUSE RENT: '||HR);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE CONVEYNCE ALLOWANCE: '||CA);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE TAX DEDUCTION: '||TD);

SALARY:= (SALARY+(MA+HR+CA))-TD;

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE NET SALARY: '||SALARY);

END;

10. Create a PL/SQL procedure FIND\_MAX () that displays the highest salary and the name of the employee receiving the highest salary. (2)

DECLARE

NAME EMP.ENAME%TYPE;

SALARY EMP.SAL%TYPE;

PROCEDURE FIND\_MAX(NAME OUT EMP.ENAME%TYPE, SALARY OUT

EMP.SAL%TYPE) IS

BEGIN

SELECT ENAME, SAL INTO NAME, SALARY FROM EMP WHERE SAL = (SELECT

MAX(SAL) FROM EMP);

DBMS\_OUTPUT.PUT\_LINE (NAME||' EARNS '||SALARY);

END;

BEGIN

FIND\_MAX(NAME, SALARY);

END;

1. Create a function named ANNUAL\_SALARY that takes salary from table EMP and returns it after multiplying salary with 12. After creating this function, test it on the EMP table. (1)

CREATE OR REPLACE FUNCTION ANNUAL\_SAL(SAL EMP.SAL%TYPE)

RETURN NUMBER IS

BEGIN

RETURN SAL\*12;

END;

SELECT ENAME,SAL,ANNUAL\_SAL(SAL) FROM EMP;

1. Create a function named NAME\_CHANGER that takes ename from table emp and returns it after changing the ‘E’ in all names with ‘I’. After creating this function, test it on the EMP table. (1)

CREATE OR REPLACE FUNCTION NAME\_CHANGER(NAME

EMP.ENAME%TYPE)

RETURN CHAR IS

BEGIN

RETURN REPLACE(NAME, 'E','I');

END;

SELECT NAME\_CHANGER(ENAME) FROM EMP;