**DBMS LAB PROGRAMS**

**160119733143**

**Snigdha Reddy**

**Q1:Implement hello world program using procedures**

**Programme:**

CREATE OR REPLACE PROCEDURE greetings

AS

BEGIN

dbms\_output.put\_line('Hello World!');

END;

EXECUTE greetings;

**Q2:Finding the minimum of two numbers using procedures**

**Programme:**

DECLARE

a number;

b number;

c number;

PROCEDURE findMin(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END;

BEGIN

a:= 44;

b:= 45;

findMin(a, b, c);

dbms\_output.put\_line(' Minimum of (44, 45) : ' || c);

END;

**Output:**

Statement processed.

Minimum of (44, 45) : 44

**Q3: procedure to find square of a number**

**Programme:**

DECLARE

a number;

PROCEDURE squareNum(x IN OUT number) IS

BEGIN

x := x \* x;

END;

BEGIN

a:= 5;

squareNum(a);

dbms\_output.put\_line(' Square of (5): ' || a);

END;

**Output:**

Statement processed.

Square of (5): 25

**Q4:Program to find total number of employees using functions**

**Programme:**

CREATE OR REPLACE FUNCTION totalEmployees

RETURN number IS

total number(2) := 0;

BEGIN

SELECT count(\*) into total

FROM scott.emp;

RETURN total;

END;

/

DECLARE

c number(2);

BEGIN

c := totalEmployees();

dbms\_output.put\_line('Total no. of Employees: ' || c);

END;

**Output:**

Function created.

Statement processed.

Total no. of Employees: 14

**Q5:Programe to get name and id of the employee based on empno,using exception handling**

**Programme:**

DECLARE

c\_empno scott.emp.EMPNO%type := 7499;

c\_name scott.emp.ENAME%type;

BEGIN

SELECT EMPNO,ENAME INTO c\_empno,c\_name

FROM scott.emp

WHERE EMPNO=c\_empno;

DBMS\_OUTPUT.PUT\_LINE ('Name: '|| c\_name);

DBMS\_OUTPUT.PUT\_LINE ('Empno: ' || c\_empno);

EXCEPTION

WHEN no\_data\_found THEN

dbms\_output.put\_line('No such Employee!');

WHEN others THEN

dbms\_output.put\_line('Error!');

END;

Output:

Statement processed.

Name: ALLEN

Empno: 7499

**Q6:Programme to show too many rows exception**

**Programme:**

DECLARE

temp varchar2(10);

BEGIN

-- raises an exception as SELECT

-- into trying to return too many rows

SELECT ENAME into temp from scott.emp ;

dbms\_output.put\_line(temp);

EXCEPTION

WHEN too\_many\_rows THEN

dbms\_output.put\_line('error trying to SELECT too many rows');

end;

**Output:**

Statement processed.

error trying to SELECT too many rows

**Q7:Programme to implement NO\_DATA\_FOUND exception**

**Programme:**

**DECLARE**

**temp number(22);**

**BEGIN**

**SELECT EMPNO into temp from scott.emp where ENAME='KING1';**

**dbms\_output.put\_line(temp);**

**exception**

**WHEN no\_data\_found THEN**

**dbms\_output.put\_line('ERROR');**

**dbms\_output.put\_line('there is no name as');**

**dbms\_output.put\_line('KING in SCOTT.emp table');**

**end;**

**Output:**

**Statement processed.**

**ERROR**

**there is no name as**

**KING in SCOTT.emp table**

**Q8:Programe to implement Zero Divide Error**

**Progarmme:**

**DECLARE**

**a int:=10;**

**b int:=0;**

**answer int;**

**BEGIN**

**answer:=a/b;**

**dbms\_output.put\_line('the result after division is '||answer);**

**exception**

**WHEN zero\_divide THEN**

**dbms\_output.put\_line('dividing by zero please check the values again');**

**dbms\_output.put\_line('the value of a is '||a);**

**dbms\_output.put\_line('the value of b is '||b);**

**END;**

**Output:**

**Statement processed.**

**the result after division is 2**