

### **EXPERIMENT: 7**

Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISEAPPLICATION ERROR. 37

**A. WHILE LOOP:** A **WHILE LOOP** statement in PL/SQL programming language repeatedly executes a target statement as long as a given condition is true.

#### **Syntax:**

WHILE condition LOOP

    sequence\_of\_statements

END LOOP;

PL/SQL Code: A PL/SQL Program to find sum of ODD number upto given number using While loop

SQL> ed 7a

```
set serveroutput on;
declare
    inval number;
    endval number;
    s number default 0;
begin
    inval:=1;
    endval:=&endval;
    while inval<endval loop
        s:=s+inval;
        inval:=inval+2;
    end loop;
    dbms_output.put_line('sum of odd numbers between 1 and '||endval||' is '|| s);
end;
/
```

SQL> @7a

Enter value for endval: 100

old 7:      endval:=&endval;

new 7:      endval:=100;

sum of odd numbers between 1 and 100 is 2500

PL/SQL procedure successfully completed.

B, **FOR Loop:** A **FOR LOOP** is a repetition control structure that allows us to efficiently write a loop

that needs to execute a specific number of times.

38

**Syntax**

FOR counter IN initial\_value .. final\_value LOOP

sequence\_of\_statements;

END LOOP;

**PL/SQL CODE:** A PL/SQL code to print multiplication table using for loop

SQL> ed 7b

```
set serveroutput on;
DECLARE
  VAR1 NUMBER;
  VAR2 NUMBER;
BEGIN
  dbms_output.put_line('Enter number to print multiplication table');
  VAR1:=&VAR1;
  FOR VAR2 IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(VAR1||'X'||VAR2||'='||VAR1*VAR2);
  END LOOP;
END;
/
```

SQL> @7b

Enter value for var1: 2

old 6: VAR1:=&VAR1;

new 6: VAR1:=2;

Enter number to print multiplication table

2X1=2

2X2=4

2X3=6

2X4=8

2X5=10

2X6=12

2X7=14

2X8=16

2X9=18

2X10=20

PL/SQL procedure successfully completed.

**C. NESTED LOOP:** PL/SQL allows using one loop inside another loop. It may be either basic, while or for loop.

**Syntax:**

```
WHILE condition1 LOOP
    sequence_of_statements1
    WHILE condition2 LOOP
        sequence_of_statements2
    END LOOP;
END LOOP;
```

**PL/SQL CODE:** A PL/SQL program to print n prime number using nested loop.

```
SQL> ed 7c
```

```
DECLARE
    i number(3);
    j number(3);
BEGIN
    i := 2;
    LOOP
        j := 2;
        LOOP
            exit WHEN ((mod(i, j) = 0) or (j = i));
            j := j + 1;
        END LOOP;
        IF (j = i) THEN
            dbms_output.put_line(i || ' is prime');
        END IF;
        i := i + 1;
        exit WHEN i = 50;
    END LOOP;
END;
```

```
/
```

SQL> @7c

---

2 is prime  
3 is prime  
5 is prime  
7 is prime  
11 is prime  
13 is prime  
17 is prime  
19 is prime  
23 is prime  
29 is prime  
31 is prime  
37 is prime  
41 is prime  
43 is prime  
47 is prime

40

---

PL/SQL procedure successfully completed.

---

**EXPERIMENT: 8**

Programs development using creation of procedures, passing parameters IN and OUT of 41  
PROCEDURES.

---

**SQL> create table enquiry (enqno1 number(3), fname varchar2(30));**

Table created.

**SQL> insert into enquiry values (111,'sai');**

1 row created.

**SQL> insert into enquiry values (112,'sindhu');**

1 row created.

**PL/SQL CODE to create procedure**

SQL> ed findname

```
create procedure findname(enquiryno1 IN number,fname1 OUT varchar2) is
fname2 varchar2(30);
begin
select fname into fname2 from enquiry where enqno1=enquiryno1;
fname1:=fname2;
exception when no_data_found then
raise_application_error(-20100,'The given number is not present');
end;
/
```

SQL> @findname

Procedure created.

---

---

## PL/SQL Code for calling procedure in program

SQL> ed pro8

42

```
set serveroutput on;
declare
enqno2 number(5);
fname2 varchar2(30);
begin
enqno2:=&enqno2;
findname(enqno2,fname2);
dbms_output.put_line('Person name of equiry id '||enqno2||' is '||fname2);
end;
/
```

SQL> @pro8

Enter value for enqno2: 114

old 5: enqno2:=&enqno2;

new 5: enqno2:=114;

declare

\*

ERROR at line 1:

ORA-20100: The given number is not present

ORA-06512: at "SYSTEM.FINDNAME", line 7

ORA-06512: at line 6

SQL> @pro8

Enter value for enqno2: 112

old 5: enqno2:=&enqno2;

new 5: enqno2:=112;

Person name of equiry id 112 is sindhu

PL/SQL procedure successfully completed.