22/9/25

Task 7: PL/PQL Procedure Function and loops

Aim:
To implement PL/SQL procedures, Functions and loops

Sample PL/SQL program (static input):

```
DECLARE
    message VARCHAR 2(20):= 'Booking closed'
BEGIN
    dbms_output.put_line (message);
END;
/
```

output:
Booking closed

Conditional statement (Dynamic input):

```
DECLARE
    hid NUMBER (3):= 100;
BEGIN
    IF (hid = 10) THEN
        dbms_output.put_line('value of hid is 10');
    ELSEIF (hid = 20) THEN
        dbms_output.Put_line ('value of hid is 20');
    ELSE IF (hid = 30) THEN
        dbms.output.put_line('value of hid is 30');
    ELSE
        dbms_output.Put_line ('None of the value is matching');
    END IF;
        dbms_output.Put_line(' Exact value of hid is:' ||hid);
    END;
    /
```

Output:
None of the value is matching
Exact value of hid is 100

3. Nested Loops Example:

```
DECLARE
    hid NUMBER (1);
    oid NUMBER (1);
BEGIN
    <<outer_loop>>
    FOR oid IN 1..3 Loop
      dbms_output.put_line ('hid is:'||hid||'
                                   ' and oid is'||oid

    END LOOP inner_Loop;
    END LOOP outer_Loop;

END;
/
```

Output:
```
hid  is :1 and oid is:1
hid  is :1 and oid is:2
hid  is :1 and oid is:3
hid  is :2 and oid is:1
hid  is :2 and oid is:2
hid  is :2 and oid is:3
hid  is :3 and oid is:1
hid  is :3 and oid is:2
hid  is :3 and oid is:3
hid
```

4. Procedure Example

```
CREATE OR REPLACE PROCEDURE booking_status
                                (Cid IN NUMBER)

IS
BEGIN
    IF  C_id > 200 THEN
    dbms_output.put_line ('No booking
                              available');
```

```
ELSE
    dbms_output.put_line('Booking open');
END;
/
```

Execution:

```
BEGIN
    booking_status(100);
    booking_status(250);
END;

/
```

output:

```
Booking  open
No booking available
```

PL/SQL procedure for loop

Example 1: using WHILE Loop with cursor

Prime check using WHILE Loops for Patient IDs

```
DECLARE
    CURSOR pat_cur IS
        SELECT patient_id FROM Patient;
    p_id Patient.Patient_id %.TYPE;
    i NUMBER;
    flag NUMBER;
BEGIN
    OPEN Pat_Cur;
    FETCH Pat_Cur INTO p_id;
    WHILE Pat_Cur %.FOUND LOOP
        flag := 0:
    FOR i IN 2...P_id/2 LOOP
        IF MOD (P_id, i)= 0 THEN
            flag := 1;
            EXIT;
        END IF;
    END LOOP;
```

Example 2: using FOR LOOP for first N prime patient IDs

```
DECLARE
    n Number := 10;
    Count NUMBER := 0;
    i   NUMBER := 2;
    j   NUMBER ;
    flag  NUMBER;
BEGIN
    WHILE   COUNT  < n  Loop
      flag := 0
      FOR   j IN  2 .. i/2   LOOP
            IF   MOD (i, j)  = 0  THEN
                 flag := 1;
                 EXIT
                 END  IF;
            END  LOOP;

      IF  flag = 0  THEN
            dbms_output.Put_line ('Prime patient ID:'
                                            || i);
            COUNT := COUNT + 1;
       END IF;
       i := i + 1;
    END  LOOP;
END ;
/
```

VEL TECH	
EX No.	07
PERFORMANCE (5)	5
RESULT AND ANALY	5
VIVA VOCE (5)	5
RECORD	–
TOTAL	15
SIGN	[illegible] 22/9/14

[illegible] 22/9/25

Result:

Thus the PL/SQL functions and Loops executed successfully