

22.9.25

Task - 7 PL/SQL

Aim: To implement PL/SQL Procedures, Functions and loops on Number theory and business scenarios.

Procedure: PL/SQL is a combination of SQL along with procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

1.) Simple PL/SQL Program (Static Input)

DECLARE

message VARCHAR (20) = 'Booking closed';

BEGIN

dbms_output.put_line (message);

END

output: Booking closed.

2.) Conditional statement (Dynamic Input)

DECLARE

sid NUMBER (2) = 100;

BEGIN

if (sid = 10) THEN

dbms_output.put_line ('value of sid is 10');

ELSE IF (sid = 20) THEN

dbms_output.put_line ('value of sid is 20');

ELSE IF (sid = 30) THEN

dbms_output.put_line ('value of sid is 30');

ENDIF

dbms_output.put_line ('exact value of sid (sid)');

END;

output: None of the value is matching
Exact value of hid 10 = 10

3.) Nested loops Example:

```
DECLARE
    hid NUMBER(1);
    oid NUMBER(1);

BEGIN
    <= outer loop >>
    for hid (1) ... loop
        <= inner loop >>
        for oid (1) 1 ... loop
            dbms_output.put_line
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

4.) Procedure Example:

```
CREATE OR REPLACE PROCEDURE booking_status
    <id NUMBER>
IS
BEGIN
    If <id > 200 THEN
        dbms_output.put_line ('no booking available')
    ELSE
        dbms_output.put_line ('Booking open');
    ENDIF;
```

Execution :

BEGIN

Booking-status (130) ;

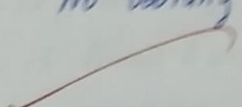
Booking-status (250) ;

END ;

Output :

Booking open

No Booking available



FOR PL/SQL Procedure for loops

Example 1: using WHILE loop with cursor

```
CREATE OR REPLACE PROCEDURE PRINT_Prime-Customer-ID
```

```
CURSOR cust_cur IS
```

```
SELECT customer_id FROM customer;
```

```
v_id NUMBER;
```

```
v_is Prime BOOLEAN;
```

```
v_i NUMBER;
```

```
BEGIN
```

```
open cust-cur;
```

```
Loop
```

```
FETCH cust-cur INTO v_id;
```

```
EXIT THEN cust-cur;
```

```
IF v_id < 2 THEN
```

```
v_i := 2;
```

```
WHILE v_i <= TRUNC(SORT(v_id)) LOOP
```

```
IF MOD(v_id, v_i) = 0 THEN
```

```
v_is_Prime := FALSE;
```

```
EXIT;
```

```
END IF;
```

```
v_i := v_i + 1;
```

```
END LOOP;
```

```
END SE cust-cur;
```

```
END;
```

The Procedure checks are customer ids in table and prints the prime ones using a loop.

Example 2: using for loop for first n prime numbers

```
CREATE OR REPLACE PROCEDURE PRINT-FIRST-n-Primes(n num)
```

```
v_num NUMBER := 2
```


V_Count NUMBER := 0

V_Is_Prime BOOLEAN;

BEGIN

WHILE V_Count = n loop

V_Is_Prime := True;

FOR i IN 2 TRUNCATE (SORT (Vnum)) LOOP

IF MOD(V_num) = 0 THEN

V_Is_Prime := FALSE;

EXIT;

END IF;

END LOOP;

IF V_Is_Prime THEN

DBMS_OUTPUT.PUT_LINE ('PRIME', '||V_num);

V_Count := V_Count + 1

END IF;

END LOOP;

END;

The Procedure prints the first n prime numbers using a for loop.

VEL TECH - CSE	
FG NO.	07
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VOICE (5)	5
	15

Result: This Procedure function, loops program using PL/SQL and loops are executed successfully.