

EXPERIMENT-8

Aim: To Practice PL/SQL Command.

Software Used: MySQL

Code: BASICS: Syntax, Comments, Variable Attributes, Conditionals: IF-THEN-ELSE, CASE, LOOPS-For, While.

Syntax:

```
DECLARE
```

```
message varchar2(20):= 'Hello, World!';
```

```
BEGIN
```

```
dbms_output.put_line(message);
```

```
END;
```

```
Hello World!
```

```
PL/SQL procedure successfully completed.
```

Comments:

```
DECLARE
```

```
-- variable declaration
```

```
message varchar2(20):= 'Hello, World!';
```

```
BEGIN
```

```
/*
```

```
* PL/SQL executable statement(s)
```

```
*/
```

```
dbms_output.put_line(message);
```

```
END;
```

```
/
```

```
Hello World!
```

```
PL/SQL procedure successfully completed.
```

Example:

```
DECLARE

a integer := 30;

b integer := 40;

c integer;

f real;

BEGIN

c := a + b;

dbms_output.put_line('Value of c: ' || c);

f := 100.0/3.0;

dbms_output.put_line('Value of f: ' || f);

END;
```

```
Value of c:70
```

```
Value of f:33.333333333333333333
```

```
PL/SQL procedure successfully completed.
```

Variable Attributes:

% TYPE

```
DECLARE

SALARY EMP.SAL % TYPE;

ECODE EMP.empno % TYPE;

BEGIN

Ecode :=&Ecode;

Select SAL into SALARY from EMP where EMPNO = ECODE;

dbms_output.put_line('Salary of ' || ECODE || 'is = || salary');

END;
```

Enter value for ecode:7499
Salary of 7499 is=1600

PL/SQL procedure successfully completed.

%ROWTYPE

DECLARE

EMPLOYEE EMP. % ROW TYPE;

BEGIN

EMPLOYEE.EMPNO := 2092;

5 EMPLOYEE.ENAME := 'Sanju';

Insert into EMP where (EMPNO, ENAME) Values (employee.empno, employee.ename);

dbms_output.put_line('Row Inserted');

END;

Row Inserted

PL/SQL procedure successfully completed.

Conditionals

1) IF -THEN-ELSE

DECLARE

a number(3) := 500;

BEGIN

-- check the boolean condition using if statement

IF(a < 20) THEN

-- if condition is true then print the following

dbms_output.put_line('a is less than 20 ');

ELSE

dbms_output.put_line('a is not less than 20 ');

END IF;

dbms_output.put_line('value of a is : ' || a);

END;

```
a is not less than 20
value of a is:500
PL/SQL procedure successfully completed.
```

2) CASE

```
DECLARE
```

```
grade char(1) := 'A';
```

```
BEGIN
```

```
CASE grade
```

```
when 'A' then dbms_output.put_line('Excellent');
```

```
when 'B' then dbms_output.put_line('Very good');
```

```
when 'C' then dbms_output.put_line('Good');
```

```
when 'D' then dbms_output.put_line('Average');
```

```
when 'F' then dbms_output.put_line('Passed with Grace');
```

```
else dbms_output.put_line('Failed');
```

```
END CASE;
```

```
END;
```

Excellent

```
PL/SQL procedure successfully completed.
```

Loop:

1) FOR

```
DECLARE
```

```
VAR1 NUMBER;
```

```
BEGIN
```

```
VAR1:=10;
```

```
FOR VAR2 IN 1..10
```

```
LOOP
```

```
DBMS_OUTPUT.PUT_LINE (VAR1*VAR2);
```

```
END LOOP;
```

```
END;
```

```
10  
20  
30  
40  
50  
60  
70  
80  
90  
100
```

```
PL/SQL procedure successfully completed.
```

2) WHILE

```
DECLARE
```

```
VAR1 NUMBER;
```

```
VAR2 NUMBER;
```

```
BEGIN
```

```
VAR1:=200;
```

```
VAR2:=1;
```

```
WHILE (VAR2<=10)
```

```
LOOP
```

```
DBMS_OUTPUT.PUT_LINE (VAR1*VAR2);
```

```
VAR2:=VAR2+1;
```

```
END LOOP;
```

```
END;
```

```
200  
400  
600  
800  
1000  
1200  
1400  
1600  
1800  
2000
```

```
PL/SQL procedure successfully completed.
```

Conclusion: Various forms of PL/SQL queries were demonstrated.