Ex.No.6	PL/SQL STATEMENTS

AIM:-

To execute PL/SQL statements in various programs.

PL/SQL: PL/SQL is a combination of SQL along with the procedural features of programming languages. PL/SQL is one of the key programming languages embedded in the Oracle Database along with SQL.

PL/SQL Control structures:-

1.Program to print Hello World:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2   msg varchar(50):='Kongu Engineering College,Perundurai!';
  3  begin
  4    dbms_output.put_line(msg);
  5  end;
  6  /
Kongu Engineering College,Perundurai!

PL/SQL procedure successfully completed.
```

2.Simple IF-THEN Statements:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  n number;
  3  BEGIN
  4  n:=&n;
  5  IF n > 0 THEN
  6  Dbms_output.put_line('Given number is greater than zero');
  7  END IF;
```

```
  8  END;
  9  /
Enter value for n: 4
old   4: n:=&n;
new   4: n:=4;
```

PL/SQL procedure successfully completed.

3.Simple IF-THEN-ELSE Statements:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  n number;
  3  BEGIN
  4  IF n > 0 THEN
  5  Dbms_output.put_line('Given number is greater than zero');
  6  ELSE
  7  Dbms_output.put_line('Given number is less than zero');
  8  END IF;
  9  END;
 10  /
Given number is less than zero
```

PL/SQL procedure successfully completed.

4. Nested IF-THEN-ELSE Statements:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  n number;
  3  BEGIN
  4  n:=&n;
  5  IF n > 0 THEN
  6  Dbms_output.put_line('Given number is greater than zero');
  7  ELSE
  8  IF n > 0 THEN
  9  Dbms_output.put_line('Given number is Equal than zero');
 10  ELSE
```

```
 11  Dbms_output.put_line('Given number is less than zero');
 12  END IF;
 13  END IF;
 14  END;
 15  /
Enter value for n: 8
old   4: n:=&n;
new   4: n:=8;
Given number is greater than zero

PL/SQL procedure successfully completed.
```

5. Nested IF-THEN-ELSE Statements:-

```
SQL> SET SERVEROUTPUT ON;
SQL>  DECLARE
  2   n number;
  3   BEGIN
  4   n:=&n;
  5   IF n > 0 THEN
  6   Dbms_output.put_line('Given number is Greater than ZERO');
  7   ELSE
  8   IF n = 0 THEN
  9   Dbms_output.put_line('Given number is Equal to ZERO');
 10   ELSE
 11   Dbms_output.put_line('Given number is Less than ZERO');
 12   END IF;
 13   END IF;
 14   END;
 15  /
Enter value for n: 0
old   4:  n:=&n;
new   4:  n:=0;
Given number is Equal to ZERO

PL/SQL procedure successfully completed.
```

6. IF-THEN-ELSEIF Statement:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  n number;
  3  BEGIN
  4  n:=&n;
  5  IF n > 0 THEN
  6  Dbms_output.put_line('Given number is Greater than ZERO');
  7  ELSIF n = 0 THEN
  8  Dbms_output.put_line('Given number is Equal to ZERO');
  9  ELSE
 10  Dbms_output.put_line('Given number is Less than ZERO');
 11  END IF;
 12  END;
 13  /
Enter value for n: 5
old   4: n:=&n;
new   4: n:=5;
Given number is Greater than ZERO

PL/SQL procedure successfully completed.
```

7. Extended IF-THEN Statement:-

```
SQL> SET SERVEROUTPUT ON;
SQL>  DECLARE
  2  grade CHAR(1);
  3    BEGIN
  4  grade := 'A';
  5   IF grade = 'A' THEN
  6    DBMS_OUTPUT.PUT_LINE('Excellent');
  7  ELSIF grade = 'B' THEN
  8   DBMS_OUTPUT.PUT_LINE('Very Good');
  9   ELSIF grade = 'C' THEN
 10   DBMS_OUTPUT.PUT_LINE('Good');
 11   ELSIF grade = 'D' THEN
 12   DBMS_OUTPUT.PUT_LINE('Fair');
 13   ELSE
 14   DBMS_OUTPUT.PUT_LINE('No Grade');
 15    END IF;
 16  END;
 17  /
Excellent

PL/SQL procedure successfully completed.
```

8. Simple CASE Statement:-

```
SQL>  SET SERVEROUTPUT ON;
SQL> DECLARE
  2  grade CHAR(1);
  3  BEGIN
  4   grade := 'A';
  5  case grade
  6  When 'A' THEN DBMS_OUTPUT.PUT_LINE('Excellent');
  7  When 'B' THEN DBMS_OUTPUT.PUT_LINE('Very Good');
  8  When 'C' THEN DBMS_OUTPUT.PUT_LINE('Good');
  9  WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('Fair');
 10  ELSE
 11  DBMS_OUTPUT.PUT_LINE('No such grade');
 12  END CASE;
 13  END;
 14  /
Excellent

PL/SQL procedure successfully completed.
```

9. Searched CASE Statement:-

```
SQL>  SET SERVEROUTPUT ON;
SQL> DECLARE
  2  grade Char(1);
  3  BEGIN
  4  grade:='C';
  5  CASE
  6  When grade = 'A' THEN DBMS_OUTPUT.PUT_LINE('Excellent');
  7  When grade = 'B' THEN DBMS_OUTPUT.PUT_LINE('Very Good');
  8  When grade = 'C' THEN DBMS_OUTPUT.PUT_LINE('Good');
  9  When grade = 'D' THEN DBMS_OUTPUT.PUT_LINE('Fair');
 10  END CASE;
 11  END;
 12  /
Good

PL/SQL procedure successfully completed.
```

10.EXCEPTION Instead of ELSE Clause in CASE Statement:-

```
SQL>  SET SERVEROUTPUT ON;
SQL> DECLARE
  2   grade CHAR(1);
  3   BEGIN
```

```
  4   grade := 'C';
  5  CASE
  6  When grade = 'A' THEN DBMS_OUTPUT.PUT_LINE('Excellent');
  7  When grade = 'D' THEN DBMS_OUTPUT.PUT_LINE('Fair');
  8  END CASE;
  9  EXCEPTION
 10  WHEN CASE_NOT_FOUND THEN
 11  DBMS_OUTPUT.PUT_LINE('No such grade');
 12  END;
 13  /
No such grade

PL/SQL procedure successfully completed.
```

11. WHILE-LOOP Statement:-

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  A NUMBER;
  3  I NUMBER :=0;
  4  BEGIN
  5  A:=10;
  6  WHILE I<A LOOP
  7  DBMS_OUTPUT.PUT_LINE('VALUE :'||I);
  8  I:=I+1;
  9  END LOOP;
 10  END;
 11  /
VALUE :0
VALUE :1
VALUE :2
VALUE :3
VALUE :4
VALUE :5
VALUE :6
VALUE :7
VALUE :8
VALUE :9

PL/SQL procedure successfully completed.
```

12. FOR-LOOP Statement:-

```
SQL> SET SERVEROUTPUT ON;
SQL> BEGIN
  2  FOR I IN 1..3 LOOP
  3  DBMS_OUTPUT.PUT_LINE (TO_CHAR(i));
```

```
  4  END LOOP;
  5  END;
  6  /
1
2
3
```

PL/SQL procedure successfully completed.

13.Reverse FOR-LOOP Statement:-

```
SQL> SET SERVEROUTPUT ON;
SQL> BEGIN
  2  FOR I IN REVERSE 1..3 LOOP
  3  DBMS_OUTPUT.PUT_LINE (TO_CHAR(i));
  4  END LOOP;
  5  END;
  6  /
3
2
1
```

PL/SQL procedure successfully completed.

14. Simple GOTO Statement:-

```
SQL>  SET SERVEROUTPUT ON;
SQL> DECLARE
  2  M VARCHAR2(30);
  3  N INTEGER :=35;
  4  BEGIN
  5  FOR  J in 2..ROUND(SQRT(N)) LOOP
  6  IF N MOD J = 0 THEN
  7  M := ' is not a prime number';
  8  GOTO print_now;
  9  END IF;
 10  END LOOP;
 11   M := ' is a prime number';
 12  <<print_now>>
 13  DBMS_OUTPUT.PUT_LINE(TO_CHAR(N) || M);
 14  END;
 15  /
35 is not a prime number
```

PL/SQL procedure successfully completed.

15. PL/SQL block for updating multiple rows into a table:-

```
SQL> declare s
  2  sp.sid%type; s_ename
  3  sp.ename%type;s_deptno
  4  sp.s_deptno%type;
  5  begin
  6  s_sid:=&s_sid;
  7  s_ename:=&s_ename;
  8  s_deptno:=&s_deptno;
  9  insert into sp(sid,ename,deptno)values(s_sid,s_ename,s_deptno);
 10  dbms_output.put_line('Row inserted successfully.');    exception
 11  when others then    dbms_output.put_line('Error:'|| SQLERRM);
 12  end;
 13  /
Enter value for s_sid: 201
old   6: s_sid:=&s_sid;
new   6: s_sid:=201;
Enter value for s_ename: Mithun
old   7: s_ename=&s_ename;
new   7: s_ename=Mithun;
Enter value for s_deptno: IT
old   8: s_deptno=&s_deptno;
new   8: s_deptno=IT;
Enter value for s_sid: 202
old   6: s_sid:=&s_sid;
new   6: s_sid:=202;
Enter value for s_ename: Jeevan
old   7: s_ename=&s_ename;
new   7: s_ename=Jeevan;
Enter value for s_deptno: IT
old   8: s_deptno=&s_deptno;
new   8: s_deptno=IT;
Row inserted successfully.
PL/SQL procedure successfully completed.
```

16. SQL block for updating multiple rows into a table:-

```
SQL> declare new_dept
  2  varchar(20):='IT';
  3  begin
  4  update sp set ename=new_dept where sid=201;
  5  commit;
  6  end;
  7  /

PL/SQL procedure successfully completed.
```

```
SQL> declare new_name
  2  varchar(20):='JEEVAN';
  3  begin
  4  update sp set ename=new_name where sid=201;
  5  commit;
  6  end;
  7  /
```

PL/SQL procedure successfully completed.

17. Do…While Statement:-

```
SQL> declare s
  2   number:=2;
  3   begin
  4   loop
  5   dbms_output.put_line(s||',');
  6  s:=s+2;
  7  exit
  8   when s>10;
  9  end loop;
 10   dbms_output.put_line('Final Number='||s);
 11  end;
 12  /
```

PL/SQL procedure successfully completed.

18. GOTO Statement to Branch to an Enclosing Block

```
SQL> declare
  2  n number:=&n;
  3  limit number:=&limit;
  4  begin
  5  for i in n..limit loop
  6  if mod(i,5)=0 then
  7  dbms_output.put_line('First multiple of 5 found'||i);
  8  gotoexit_label;
  9  end if;
 10  end loop;
 11  dbms_output.put_line('No multiple of 5 found in the given range.');
 12  <<exit_label>>
 13  null;
 14  end;
 15  /
```

```
Enter value for n: 3
old   2: n number:=&n;
new   2: n number:=3;
Enter value for limit: 10
old   3: limit number:=&limit;
new   3: limit number:=10;

PL/SQL procedure successfully completed.
```

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim , Algorithm, SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT:-

Thus the procedure and functions are executed successfully.