COURSE OUTCOME 2

Apply PL/SQL for processing databases.

PROGRAM 1

AIM: Create a PL/SQL to add two numbers.

```
SQL> set serveroutput on;
SQL> declare
  2  A integer;
  3  B integer;
  4  begin
  5  A := &A;
  6  B := &B;
  7  DBMS_OUTPUT.PUT_LINE('Sum:'||(A+B));
  8  end;
  9  /
Enter value for a: 10
old   5: A := &A;
new   5: A := 10;
Enter value for b: 30
old   6: B := &B;
new   6: B := 30;
Sum:40

PL/SQL procedure successfully completed.
```

PROGRAM 2

AIM: Create a PL/SQL to find the greatest of three numbers.

```
SQL> declare
  2  A integer;
  3  B integer;
  4  C integer;
  5  begin
  6  A := &A;
  7  B := &B;
  8  C := &C;
  9  if (A>B and B>C) then
 10  DBMS_OUTPUT.PUT_LINE('Greatest number is '||A);
 11  elsif (b>c) then
 12  DBMS_OUTPUT.PUT_LINE('Greatest number is '||B);
 13  else
 14  DBMS_OUTPUT.PUT_LINE('Greatest number is '||C);
 15  end if;
 16  end;
 17  /
Enter value for a: 10
old   6: A := &A;
new   6: A := 10;
Enter value for b: 30
old   7: B := &B;
new   7: B := 30;
Enter value for c: 4
old   8: C := &C;
new   8: C := 4;
Greatest number is 30

PL/SQL procedure successfully completed.
```

PROGRAM 3

AIM: Create a procedure to display all odd and even numbers between 1 and 20.

```
SQL> create or replace procedure display_odd_even IS BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Even Numbers between 1 and 20');
  3  FOR i IN 1..20
  4  LOOP
  5      IF MOD(i, 2) = 0 THEN
  6         DBMS_OUTPUT.PUT_LINE(i);
  7      END IF;
  8  END LOOP;
  9
 10  DBMS_OUTPUT.PUT_LINE('Odd Numbers between 1 and 20');
 11  FOR i IN 1..20
 12  LOOP
 13      IF MOD(i, 2) != 0 THEN
 14         DBMS_OUTPUT.PUT_LINE(i);
 15      END IF;
 16  END LOOP;
 17  END;
 18  /

Procedure created.
```

```
SQL> set serveroutput on;
SQL> exec display_odd_even;
Even Numbers between 1 and 20
2
4
6
8
10
12
14
16
18
20
Odd Numbers between 1 and 20
1
3
5
7
9
11
13
15
17
19

PL/SQL procedure successfully completed.
```

PROGRAM 4

AIM: Create a PL/SQL to check whether the entered number is odd or even.

```
SQL> declare
  2  A integer;
  3  begin
  4  A := &A;
  5  if mod(A,2)=0 then
  6  DBMS_OUTPUT.PUT_LINE(A||' is even number');
  7  else
  8  DBMS_OUTPUT.PUT_LINE(A||' is odd number');
  9  end if;
 10  end;
 11  /
Enter value for a: 9
old   4: A := &A;
new   4: A := 9;
9 is odd number

PL/SQL procedure successfully completed.

SQL> /
Enter value for a: 4
old   4: A := &A;
new   4: A := 4;
4 is even number

PL/SQL procedure successfully completed.
```

PROGRAM 5

AIM: Create a procedure to display highest marks of a student.

```
SQL> select * from student_info;

   STUD_ID STUD_CODE  STUD_NAME                 SUBJECT
---------- ---------- ------------------------- -------------------------
     MARKS PHONE_NUMBE
---------- -----------
         1 101        Mark                      English
        68 9909076920

         2 102        Joseph                    Physics
        70 9909076920

         3 103        John                      Maths
        70 9909076920


   STUD_ID STUD_CODE  STUD_NAME                 SUBJECT
---------- ---------- ------------------------- -------------------------
     MARKS PHONE_NUMBE
---------- -----------
         4 104        Aiden                     Science
        80 9903076920
```

```
SQL> create or replace procedure display_max_mark (highestmark out number) is
  2  begin
  3  select max(marks) into highestmark from student_info;
  4  end;
  5  /

Procedure created.

SQL> variable v_highest number;
SQL> exec display_max_mark(:v_highest);

PL/SQL procedure successfully completed.

SQL> print v_highest;

 V_HIGHEST
----------
        80
```

PROGRAM 6

AIM: Create a function to display highest salary of employees.

```
SQL> create or replace function highest_sal return number is highest number;
  2  begin
  3  select max(salary) into highest from employee;
  4  return highest;
  5  end;
  6  /

Function created.

SQL> select highest_sal from dual;

HIGHEST_SAL
-----------
      65000
```

PROGRAM 7

AIM: Create a function to find the largest of two numbers.

```
SQL> create or replace function large(a in number, b in number) return number is l number;
  2  begin
  3  if (a>b) then
  4  l:=a;
  5  else
  6  l:=b;
  7  end if;
  8  return l;
  9  end large;
 10  /

Function created.

SQL> select large(10,20) from dual;

LARGE(10,20)
------------
          20
```

PROGRAM 8

AIM: Create a cursor to display names of employees.

```
SQL> declare
  2  cursor c is
  3  select first_name
  4  from employee
  5  where salary > 50000;
  6  v_ename employee.first_name%TYPE;
  7  begin
  8  open c;
  9  loop
 10  fetch c into v_ename;
 11  exit when c%NOTFOUND;
 12  dbms_output.put_line('Employee Name: ' || v_ename);
 13  end loop;
 14  close c;
 15  end;
 16  /
Employee Name: Riya
Employee Name: Alice
Employee Name: Sarah
Employee Name: Sandy

PL/SQL procedure successfully completed.
```

PROGRAM 9

AIM: Create a trigger for a table in the database.

```
SQL> create or replace trigger emp_trigger
  2  before insert on employee
  3  for each row
  4  enable
  5  begin
  6  dbms_output.put_line('New Record Inserted');
  7  end;
  8  /

Trigger created.
```

```
SQL> set serveroutput on;
SQL> insert into employee values(11,'Amith','Boby',90000,11);
New Record Inserted

1 row created.
```

PROGRAM 10

AIM: Create a procedure to calculate the sum of three numbers (without parameters)

```
SQL> create table numbers (num int);

Table created.
```

```
SQL> insert into numbers values(6);

1 row created.

SQL> insert into numbers values(4);

1 row created.

SQL> insert into numbers values(9);
```

```
SQL> create or replace procedure sum_three is
  2  s number;
  3  begin
  4  select sum(num) into s from numbers;
  5  dbms_output.put_line('Sum is ' || s);
  6  end;
  7  /

Procedure created.

SQL> exec sum_three;
Sum is 19

PL/SQL procedure successfully completed.
```

PROGRAM 11

AIM: Create a cursor to display the details of students.

```
SQL> declare
  2  s_id student_info.stud_id%type;
  3  s_name student_info.stud_name%type;
  4  s_phone student_info.phone_number%type;
  5  cursor c is
  6  select stud_id,stud_name,phone_number from student_info;
  7  begin
  8  open c;
  9  loop
 10  fetch c into s_id,s_name,s_phone;
 11  exit when c%notfound;
 12  dbms_output.put_line(s_id || ' ' || s_name || ' ' || s_phone);
 13  end loop;
 14  close c;
 15  end;
 16  /
1 Mark 9909076920
2 Joseph 9909076920
3 John 9909076920
4 Aiden 9903076920

PL/SQL procedure successfully completed.
```