7. Write a query to find out the project name which is not assigned to any employee(tables :- [EmployeeDetail],[ProjectDetail]).

```
SELECT ProjectName FROM [EmployeeDetail] A RIGHT OUTER JOIN [ProjectDe
tail] B ON A.EmployeeID = B.EmployeeDetailID
WHERE FirstName IS NULL
```

8. Write down the query to fetch EmployeeName & Project who has assign more than one project.

```
Select EmployeeID, FirstName, ProjectName from [EmployeeDetail] E INNER JOIN
[ProjectDetail] P
ON E.EmployeeID = P.EmployeeDetailID
WHERE EmployeeID IN (SELECT EmployeeDetailID FROM [ProjectDetail] GROU
P BY EmployeeDetailID HAVING COUNT(*) >1 )
```

9. Write down the query to fetch ProjectName on which more than one employee are working along with EmployeeName.

```
Select P.ProjectName, E.FName from ProjectDetails P INNER JOIN EmployeeDetail
s E
on p.EmployeId = E.Id where P.ProjectName in(select ProjectName from ProjectDeta
ils group by ProjectName having COUNT(1)>1)
```

PL/SQL

SET-7

1. Write a Pl/SQL program to Q *Hellow world

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  message varchar2(20):='Hello World!';
  3  BEGIN
  4  dbms_output.put_line(message);
  5  END;
  6  /
Hello World!


PL/SQL procedure successfully completed.
```

2. Write a PL/SQL block to find the maximum number from the given three numbers.

```
SQL> DECLARE
  2  a number;
  3  b number;
  4  c number;
  5  begin
  6  a:=&a;
  7  b:=&b;
  8  c:=&c;
  9  if(a>b and a>c)then
 10  dbms_output.put_line('a is maximum'||a);
 11  elsif(b>a and b>c)then
 12  dbms_output.put_line('b is maximum'||b);
```

```
 13  else
 14  dbms_output.put_line('c is maximum'||c);
 15  end if;
 16  end;
 17  /
Enter value for a: 4
old   6: a:=&a;
new   6: a:=4;
Enter value for b: 2
old   7: b:=&b;
new   7: b:=2;
Enter value for c: 5
old   8: c:=&c;
new   8: c:=5;
c is maximum5

PL/SQL procedure successfully completed.
```

3. Write a Pl/SQL program to print integers from 1 to 10 by using PL/SQL FOR loop

```
SQL> DECLARE
  2  n_times NUMBER:=10;
  3  BEGIN
  4  FOR n_i IN 1..n_times LOOP
  5  DBMS_OUTPUT.PUT_LINE(n_i);
  6  END LOOP;
  7  END;
  8  /
1
```

```
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.
```

4. Write a program to accept a number and find the sum of the digits .

```
SQL> declare
  2  n number(5):=&n;
  3  s number:=0;
  4  r number(2):=0;
  5  begin
  6  while n!=0
  7  loop
  8  r:=mod(n,10);
  9  s:=s+r;
 10  n:=trunc(n/10);
 11  end loop;
 12  dbms_output.put_line('sum of digits of given numbers is '||s);
 13  end;
 14  /
Enter value for n: 234
old   2: n number(5):=&n;
```

```
new   2: n number(5):=234;
sum of digits of given numbers is 9


PL/SQL procedure successfully completed.
```

5. Find the greatest number of inputs from the console.

```
SQL> declare
  2  a number(2) :=&value_of_a;
  3  b number(2) :=&value_of_b;
  4  Begin
  5  if a<b then
  6  dbms_output.put_line(' Smaller Value is '||a);
  7  elsif a>b then
  8  dbms_output.put_line(' Smaller Value is '||b);
  9  else
 10   dbms_output.put_line(' Both no. are equal ');
 11  end if;
 12  END;
 13  /
Enter value for value_of_a: 12
old   2: a number(2) :=&value_of_a;
new   2: a number(2) :=12;
Enter value for value_of_b: 33
old   3: b number(2) :=&value_of_b;
new   3: b number(2) :=33;
Smaller Value is 12


PL/SQL procedure successfully completed.
```

6. Reading the values from EMployee table.

Create table employee(ssn number(2),fname varchar(20),lname varchar(20),salary number(38));

Table created.

Insert into employee values(101,'amrutha','biju',75000);

1 row created.

Insert into employee values(102,'anite','jose',75000);

1 row created.

Insert into employee values(103,'anna','maria',75000);

1 row created.

Insert into employee values(104,'bharathi','s',75000);

1 row created.

```
Declare
 v_name employee.fname%type;
 v_job employee.lname%type;
 v_sal employee.salary%type;
Begin
 select fname,lname,salary
 into v_fname, v_lname, v_salary
 from employee
 where ssn =102;
 dbms_output.put_line(v_fname||' '||v_lname||' '||v_salary);
End;
/
```

```
fname  lname  salary
------- ---------- ---------
anite   jose       75000
```

SET-8

Named PL SQL Procedure and Functions

1.

```
SQL> SET SERVEROUTPUT ON;

SQL> CREATE OR REPLACE PROCEDURE welcome_msg (p_name IN
VARCHAR2)
  2  IS
  3  BEGIN
  4  dbms_output.put_line ('Welcome' || p_name);
  5  END;
  6  /

Procedure created.

SQL> EXEC welcome_msg ('Guru99');
WelcomeGuru99

PL/SQL procedure successfully completed.
```

2.

```
SQL> CREATE OR REPLACE PROCEDURE welcome_msg (p_name IN
VARCHAR2,salary out number)
        IS
        BEGIN
        salary:=10000;
       dbms_output.put_line ('Welcome ' || p_name);
        END;
       /
    Procedure created.
    SQL> var sal number;
    SQL> EXEC welcome_msg ('Amrutha',:sal);
```

Welcome Amrutha

PL/SQL procedure successfully completed.

Print sal;

3.

```
SQL> CREATE OR REPLACE FUNCTION welcome_msg_func ( p_name IN
VARCHAR2) RETURN VARCHAR2
    IS
    BEGIN
    RETURN ('Welcome '|| p_name);
    END;
      /
  Function created.
  SQL> DECLARE
    lv_msg VARCHAR2(250);
    BEGIN
   lv_msg:=welcome_msg_func ('Amrutha');
    dbms_output.put_line(lv_msg);
    END;
  7 /
```

Welcome Amrutha

PL/SQL procedure successfully completed.

SQL> SELECT welcome_msg_func('Amrutha') FROM DUAL;

WELCOME_MSG_FUNC('Amrutha')

Welcome Amrutha

SET-9

PL/SQL Cursor , Triggor

1.

```
SQL> create table stud_file(sid number, name varchar(20), m1 number, m2 number);
Table created.


SQL> insert into stud_file values(1,'anu',40,45);
1 row created.
SQL> insert into stud_file values(2,'binu',48,45);
1 row created.
SQL> insert into stud_file values(3,'cini',30,45);
1 row created.
SQL> insert into stud_file values(4,'dini',30,25);
1 row created.
1.SQL> declare
  2  id constant number :=1;
  3  sname studs_file.name%type;
  4  mark1 studs_file.m1%type;
  5  mark2 studs_file.m2%type;
  6  total number:=0;
  7  begin
  8  select name,m1,m2 into sname,mark1,mark2 from studs_file where sid=id;
  9  total:=mark1+mark2;
 10  dbms_output.put_line('Total marks of student '||sname||' with id '||id||' is:
 '||total);
 11  end;
 12  /
```

Output

```
Total marks of student anu with id 1 is: 85
PL/SQL procedure successfully completed.
```

2.

```
SQL> declare
  2  cursor stud_cursor is select * from studs_file;
  3  stud_rec stud_cursor%rowtype;
  4  total number:=0;
  5  begin
  6  open stud_cursor;
  7  loop
  8  fetch stud_cursor into stud_rec;
  9  exit when stud_cursor%notfound ;
 10  total:=stud_rec.m1+stud_rec.m2;
 11  dbms_output.put_line('Total marks of student '||stud_rec.name||' is: '||total);
 12  end loop;
 13  end;
 14  /
```

Output

```
Total marks of student anu is: 85
Total marks of student binu is: 93
Total marks of student cini is: 75
Total marks of student dini is: 55
PL/SQL procedure successfully completed.
SQL> create table stud_mark(sid number,total number);

Table created.
```

3.

```
SQL> create  or replace trigger stud_trig after insert on studs_file
  2  for each row
  3  declare
  4  tot number:=0;
```

```
 5  begin
 6  tot:=:new.m1+:new.m2;
 7  insert into stud_mark values(:new.sid,tot);
 8  DBMS_OUTPUT.PUT_LINE('AFTER INSERT trigger activated:');
 9
10  end;
11  /
```

Trigger created.

```
SQL> insert into studs_file values(5,'rani',40,45);
AFTER INSERT trigger activated:

1 row created.

SQL> select * from stud_mark;
```

Output

```
       SID      TOTAL
---------- ----------
         5         85
```

SET-10

Student(rollno,name,date_of_birth,course_id,city,fees_paid,marks)

Course(course_id,course_sdesc,duration,course_fees)

1. Create above table with proper constraints(Enter atleast 5 valid records).

```
SQL> create table course(course_id varchar(5) primary key,course_desc
varchar(10),duration varchar(10),course_fees number(6));

Table created.

SQL> create table student(rollno number(2),name varchar(20),date_of_birth
date,course_id varchar(5)references course(course_id),city varchar(20),fees_paid
number(5),marks number(3));

Table created.
```

```
SQL> insert into course values('co1','bca','3year',100000);

1 row created.

SQL> insert into course values('co2','bba','3year',50000);

1 row created.

SQL> insert into course values('co3','mca','2year',200000);

1 row created.

SQL> insert into course values('co4','bcom','3year',80000);

1 row created.

SQL> insert into course values('co5','btech','4year',300000);

1 row created.

SQL> insert into student values(01,'ammu','15-aug-87','co1','pala',10000,75);

1 row created.

SQL> insert into student values(02,'anu','16-dec-86','co2','pala',5000,60);

1 row created.

SQL> insert into student values(03,'manu','15-aug-87','co3','kottayam',20000,45);

1 row created.

SQL> insert into student values(04,'vinu','12-dec-99','co4','idukki',15000,55);

1 row created.

SQL> insert into student values(05,'maya','11-jan-91','co5','kottayam',0,35);

1 row created.

SQL> select *from student;
```

```
ROLLNO NAME       DATE_OF_B COURS CITY          FEES_PAID     MARKS
---------- -------------------- --------- ----- -------------------- ---------- ---------- -------------
mjk
  1        ammu          15-AUG-87         co1   pala                  10000         75
  2        anu           16-DEC-86         co2   pala                   5000         60
  3        manu          15-AUG-87         co3   kottayam              20000         45
  4        vinu          12-DEC-99         co4   idukki                15000         55
  5        maya           11-JAN-91        co5   kottayam                  0         35
```

SQL> select *from course;

```
COURS   COURSE_  DES DURATION    COURSE_FEES
----- ---------- ---------- ----------- ----------------- ---------------
co1       bca                3year              100000
co2       bba                3year              50000
co3       mca                2year              200000
co4       bcom               3year              80000
co5       btech              4year              300000
```

2. List details of student whose birth date is 15th august 87.

SQL> select *from student where date_of_birth='15-aug-87';

```
 ROLLNO   NAME     DATE_OF_B COURS CITY      FEES_PAID      MARKS
---------- -------------------- --------- ----- -------------------- ---------- -------------------
   1       ammu            15-AUG-87   co1     pala               10000         75
   3       manu            15-AUG-87   co3     kottayam           20000         45
```

3. Display details of students whose marks are less than 50 and not paid a fee.

SQL> select * from student where fees_paid=0 AND marks<50;

```
ROLLNO   NAME     DATE_OF_B COURS CITY      FEES_PAID      MARKS
---------- -------------------- --------- ----- -------------------- ---------- -------------------
   5       maya            11-JAN-91   co5       kottayam               0         35
```

4.Display city wise count of students.

SQL> select city,count(city)from student group by city;

```
CITY                 COUNT(CITY)
-------------------- -----------
```

kottayam 2
pala 2
idukki 1

5. Display total fees paid.

SQL> select sum(fees_paid)as total_fees from student;

TOTAL_FEES

50000

6. write PL/SQL block to display the name and mark of the top student.

```
SQL> set serveroutput on
SQL>  declare
  2  name student.name%type;
  3  marks student.marks%type;
  4  begin
  5  select name,marks into name,marks from student where marks=(select
max(marks)from student);
  6  dbms_output.put_line('name'||name||'marks'||marks);
  7  end;
  8  /
```

output
nameammumarks75

PL/SQL procedure successfully completed.