EXERCISE 1

Write a PL/SQL code to create an employee database with the tables and fields specified as below.

a) Employee

Emp_no	Employee_name	Street	City

b) Works

Emp_no	Company_name	Joining_date	Designation	Salary

c) Company

Emp_no	City

d) Manages

Emp_no	Manager_name	Mang_no

Note: Primary keys are underlined.

SOLUTION:

```
SQL> create table employee (emp_no number(10) primary key, employee_name varchar2(20),street varchar2(20),city varchar2(20));

Table created.

SQL> create table works (emp_no number(10) references employee, company_name varchar2(20), joining_date date, designation varchar2(20), salary number(10,2));

Table created.

SQL> create table company (emp_no number(10) references employee, city varchar2(20));

Table created.

SQL> create table manages(emp_no number(10)references employee,manager_name varchar2(20),mang_no number(20));

Table created.

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- -----------------
 EMP_NO                                    NOT NULL NUMBER(10)
 EMPLOYEE_NAME                                      VARCHAR2(20)
 STREET                                             VARCHAR2(20)
 CITY                                               VARCHAR2(20)
```

```
SQL> desc works;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_NO                                              NUMBER(10)
 COMPANY_NAME                                        VARCHAR2(20)
 JOININD_DATE                                        DATE
 DESIGNATION                                         VARCHAR2(20)
 SALARY                                              NUMBER(10,2)

SQL> desc manages;
 Name                                   Null?    Type
 ------------------------- --------------- --------
 EMP_NO                                           NUMBER(10)
 MANAGER_NAME                                     VARCHAR2(20)
 MANG_NO                                          NUMBER(20)



SQL> desc company;
 Name                                      Null?    Type
 ----------------------------- --------------- --------
 EMP_NO                                           NUMBER(10)
 CITY                                             VARCHAR2(20)

SQL> create sequence emp_seq;

Sequence created.

SQL> insert into employee values(emp_seq.nextval,'rajesh','first
cross','gulbarga');

1 row created.

SQL> insert into employee values(emp_seq.nextval,'paramesh','second
cross','bidar');

1 row created.

SQL> insert into employee values(emp_seq.nextval,'pushpa','ghandhi
road','banglore');

1 row created.

SQL> insert into employee values(emp_seq.nextval,'vijaya','shivaji
nagar','manglore');

1 row created.

SQL> insert into employee values(emp_seq.nextval,'keerthi','anand
sagar street','bijapur');

1 row created.

SQL> select * from employee;

   EMP_NO  EMPLOYEE_NAME    STREET          CITY
------------- ------------------------- -------------------- ------------
        1  rajesh           first cross          gulbarga
```

```
       2  paramesh          second cross         bidar
       3  pushpa            ghandhi road         banglore
       4  vijaya            shivaji nagar        manglore
       5  keerthi           anand sagar street   bijapur

SQL> insert into works values(1,'abc','23-nov-2000','project
lead',40000);

1 row created.

SQL> insert into works values(2,'abc','25-dec-2010','software
engg',20000);

1 row created.

SQL> insert into works values(3,'abc','15-jan-2011','software
engg',19000);

1 row created.

SQL> insert into works values(4,'abc','19-jan-2011','software
engg',19000);

1 row created.

SQL> insert into works values(5,'abc','06-feb-2011','software
engg',18000);

1 row created.

SQL> select * from works;

    EMP_NO COMPANY_NAME  JOININD_D  DESIGNATION       SALARY
---------- -------------------- --------- -------------------- ------
       1    abc        23-NOV-00   project lead          40000
       2    abc        25-DEC-10   software engg         20000
       3    abc        15-JAN-11   software engg         19000
       4    abc        19-JAN-11   software engg         19000
       5    abc        06-FEB-11   software engg         18000

 SQL> insert into company values(1,'gulbarga');

1 row created.

SQL> insert into company values(2,'bidar');

1 row created.

SQL> insert into company values(3,'banglore');

1 row created.
SQL> insert into company values(4,'manglore');

1 row created.

SQL> insert into company values(5,'bijapur');
```

```
1 row created.

SQL> select * from company;

    EMP_NO CITY
---------- --------------------
         1 gulbarga
         2 bidar
         3 banglore
         4 manglore
         5 bijapur

SQL> insert into manages values(2,'rajesh',1);

1 row created.

SQL> insert into manages values(3,'rajesh',1);

1 row created.

SQL> insert into manages values(4,'rajesh',1);

1 row created.

SQL> insert into manages values(5,'rajesh',1);

1 row created.



SQL> select * from company;

    EMP_NO CITY
---------- --------------------
         1 gulbarga
         2 bidar
         3 banglore
         4 manglore
         5 bijapur

SQL> select * from manages;
      EMP_NO MANAGER_NAME            MANG_NO
---------- -------------------- ----------
         2 rajesh                        1
         3 rajesh                        1
         4 rajesh                        1
         5 rajesh                        1
```

EXERCISE 2

Write a PL/SQL code to create an student database with the tables and fields specified as below.

a) Student

Roll no	Student_name	Course	Gender

b) Student_personal

Roll no	DOB	Father_name	Address	Place

c) Student_enrollment

Roll no	Course	Course_code	Sem	Total_marks	Percentage

SOLUTION:

```
SQL> create table student(roll_no number(10)primary key,student_name
varchar2(20),course varchar2(5),gender varchar2(10));

Table created.

SQL> create table student_personal(roll_no number(10)references student,
dob date, father_name varchar2(20),address varchar2(20),place
varchar2(20));

Table created.

SQL> create table student_enrollment(roll_no number(10)references
student, course varchar2(10),course_code varchar2(10),sem
number(2),total_marks number(30),percentage number(10));

Table created.

SQL> insert into student values(111,'ravi','cs','male');

1 row created.

SQL> insert into student values(112,'praveen','cs','male');

1 row created.

SQL> insert into student values(113,'bhuvana','is','female');

1 row created.

SQL> insert into student values(114,'apparna','is','female');

1 row created.

SQL> insert into student_personal values(111,'12-nov-
1099','annayya','#50','gulbarga');
```

```
1 row created.

SQL> insert into student_personal values(112,'13-dec-
1099','poornayya','#34','gulbarga');

1 row created.

SQL> insert into student_personal values(113,'14-jan-
1098','ramayya','#56','gulbarga');

1 row created.

SQL> insert into student_personal values(114,'15-feb-
1098','ganesh','#78','gulbarga');

1 row created.

SQL> insert into student_enrollment values(111,'cs','1001','1',500,83);

1 row created.

SQL> insert into student_enrollment values(112,'cs','1001','1',555,92);

1 row created.

SQL> insert into student_enrollment values(113,'is','1002','1',465,77);

1 row created.

SQL> insert into student_enrollment values(114,'is','1002','1',585,97);

1 row created.

SQL> commit;

Commit complete.

SQL> select * from student;

   ROLL_NO STUDENT_NAME         COURS GENDER
---------- -------------------- ----- ----------
       111 ravi                 cs    male
       112 praveen              cs    male
       113 bhuvana              is    female
       114 apparna              is    female

SQL> select * from student_personal;

   ROLL_NO   DOB      FATHER_NAME        ADDRESS    PLACE
---------- --------- -------------------- --------------------

   111     12-NOV-99   annayya             #50    gulbarga
   112     13-DEC-99   poornayya           #34    gulbarga
   113     14-JAN-98   ramayya             #56    gulbarga
   114     15-FEB-98   ganesh              #78    gulbarga
```

```
SQL> select * from student_enrollment;

   ROLL_NO COURSE     COURSE_COD        SEM TOTAL_MARKS PERCENTAGE
---------- ---------- ---------- ---------- ----------- ----------
       111 cs         1001                1         500         83
       112 cs         1001                1         555         92
       113 is         1002                1         465         77
       114 is         1002                1         585         97
```

EXERCISE 3

Write a PL/SQL code to retrieve the employee name, join_date, and designation from employee database of an employee whose number is input by the user.

SOLUTION:

```
SQL> select * from employee;

    EMP_NO EMPLOYEE_NAME        STREET               CITY
---------- -------------------- -------------------- ------------------
         1 rajesh               first cross          gulbarga
         2 paramesh             second cross         bidar
         3 pushpa               ghandhi road         banglore
         4 vijaya               shivaji nagar        manglore
         5 keerthi              anand sagar street   bijapur

NOTE :( THE PL/SQL CODE HAS BEEN TYPED IN NOTEPAD AND SAVED AS P1.SQL
UNDER E: DIRECTORY. HENCE THE COMMAND E:/P1.SQL)

SQL> get e:/P1.sql;
  1   declare
  2   eno employee.emp_no%type;
  3   ename employee.employee_name%type;
  4   begin
  5   eno:=&eno;
  6   select emp_no,employee_name into eno,ename from employee where
      emp_no=eno;
  7   dbms_output.put_line('---------output----------');
  8   dbms_output.put_line('employee no  :'||eno);
  9   dbms_output.put_line('employee name :'||ename);
 10*  end;
SQL> set serveroutput on;
SQL> /
Enter value for eno: 1
old   5:  eno:=&eno;
new   5:  eno:=1;

---------output----------
employee no  :1
employee name :rajesh
PL/SQL procedure successfully completed.

SQL> /
Enter value for eno: 3
old   5:  eno:=&eno;
new   5:  eno:=3;

---------output----------
employee no  :3
employee name :pushpa

PL/SQL procedure successfully completed.
```

EXERCISE 4

Write a PL/SQL code to show TABLE type of data(Array)

SOLUTION:

```
SQL> create or replace type A1 is table of Number(2);
  2  .
SQL> /

Type created.

SQL> create or replace type A2 is table of A1;
  2  .
SQL> /

Type created.

SQL> declare
  2   a A2;
  3  begin
  4   a := new A2(A1( 1,2,3,4 ),A1( 5,6,7,8 ),
  5               A1( 9,10,11,12 ),A1( 13,14,15,16 ));
  6
  7   DBMS_OUTPUT.PUT_LINE('     OUTPUT     ');
  8   DBMS_OUTPUT.PUT_LINE('----------------');
  9
 10   for x in 1..a.Count
 11   loop
 12    for y in 1..a(x).Count
 13    loop
 14     DBMS_OUTPUT.PUT(rpad(a(x)(y),4));
 15    end loop;
 16    DBMS_OUTPUT.PUT_LINE('');
 17   end loop;
 18  end;
 19  .
SQL> /
OUTPUT
----------------
1   2   3   4
5   6   7   8
9   10  11  12
13  14  15  16

PL/SQL procedure successfully completed.
```

EXERCISE 5

Write a PL/SQL code to calculate tax for an employee of an organization –XYZ and to display his/her name & tax, by creating a table under employee database as below.

a) Employee_salary

Emp_no	Basic	HRA	DA	Total_deduction	Net_salary	Gross_salary

SOLUTION:

```
SQL> select * from employee_salary;

    EMP_NO  BASIC     HRA   DA TOTAL_DEDUCTION NET_SALARY  GROSS_SALARY
---------- ---------- ---------- ---------- --------------- ----------
     2      15000    4000   1000      5000      15000         20000
     1      31000    8000   1000      5000      35000         40000
     3      14000    4000   1000      5000      15000         19000
     4      14000    4000   1000      5000      15000         19000
     5      13000    4000   1000      5000      15000         18000


SQL> get e:/l5.sql
  1  declare
  2  tax number:=0;
  3  net number;
  4  eno employee.emp_no%type;
  5  name employee.employee_name%type;
  6  begin
  7  eno:=&eno;
  8  select net_salary into net from employee_salary where
  9  emp_no=eno;
 10  select employee_name into name from employee where
 11  emp_no=eno;
 12  net:=net*12;
 13  if net>190000 then
 14  net:=net-190000;
 15  tax:=net*0.2;
 16  end if;
 17  dbms_output.put_line('name of the employee is '||name);
 18  dbms_output.put_line('Taxable amount is '||tax);
 19* end;
 20  .

SQL> /

Enter value for eno: 1
old   7: eno:=&eno;
new   7: eno:=1;
name of the employee is rajesh
Taxable amount is 46000

PL/SQL procedure successfully completed.

SQL> /
```

```
Enter value for eno: 2
old   7: eno:=&eno;
new   7: eno:=2;
name of the employee is paramesh
Taxable amount is 0

PL/SQL procedure successfully completed.
```

EXERCISE 6

Write a PL/SQL code to calculate total and percentage of marks of the students in four subjects.

SOLUTION:

```
SQL> get e:/p6.sql;
  1  declare
  2  rno number(10);
  3  s1 number(10);
  4  s2 number(10);
  5  s3 number(10);
  6  s4 number(10);
  7  tot number(10);
  8  per number(4);
  9  begin
 10  rno:=&rno;
 11  s1:=&s1;
 12  s2:=&s2;
 13  s3:=&s3;
 14  s4:=&s4;
 15  tot:=s1+s2+s3+s4;
 16  per:=tot*0.25;
 17  dbms_output.put_line('Regno  s1  s2  s3  s4  total  per');
 18  dbms_output.put_line(rno||'  '||s1||'  '||s2||'  '||s3||'  '||s4||'
     '||tot||'  '||per);
 19* end;
 20  .

SQL> set serveroutput on;
SQL> /
Enter value for rno: 111
old  10: rno:=&rno;
new  10: rno:=111;
Enter value for s1: 78
old  11: s1:=&s1;
new  11: s1:=78;
Enter value for s2: 68
old  12: s2:=&s2;
new  12: s2:=68;
Enter value for s3: 89
old  13: s3:=&s3;
new  13: s3:=89;
Enter value for s4: 56
old  14: s4:=&s4;
new  14: s4:=56;

Regno  s1  s2  s3  s4  total  per
   111  78  68  89  56  291  73

PL/SQL procedure successfully completed.
```

EXERCISE 7

Write a PL/SQL code to calculate the total and the percentage of marks of the students in four subjects from the table- Student with the schema given below.

STUDENT (RNO , S1 , S2, S3, S4, total, percentage)

SOLUTION:

```
SQL> create table student(rno number(10),s1 number(10),s2 number(10),s3
number(10),s4 number(10),total number(20),percentage number(6));

Table created.

SQL> insert into student(rno,s1,s2,s3,s4)values(10011,56,78,79,56);

1 row created.

SQL> insert into student(rno,s1,s2,s3,s4)values(10012,45,67,34,58);

1 row created.

SQL> insert into student(rno,s1,s2,s3,s4)values(10013,76,86,94,58);

1 row created.

SQL> insert into student(rno,s1,s2,s3,s4)values(10014,57,48,39,92);

1 row created.

SQL> select * from student;

       RNO      S1      S2      S3       S4   TOTAL   PERCENTAGE
---------- ---------- ---------- ---------- ---------- ---------- ------
     10011      56      78      79       56
     10012      45      67      34       58
     10013      76      86      94       58
     10014      57      48      39       92


SQL> get e:/plsql/l7.sql;
  1  declare
  2  t student.total%type;
  3  p student.percentage%type;
  4  cursor stu is select * from student;
  5  rw stu%rowtype;
  6  begin
  7  open stu;
  8  loop
  9  fetch stu into rw;
 10  exit when stu%notfound;
 11  t:=rw.s1+rw.s2+rw.s3+rw.s4;
 12  p:=t*0.25;
 13  update student set total=t,percentage=p where rno=rw.rno;
 14  end loop;
 15  close stu;
 16* end;
 17  .
SQL> /

PL/SQL procedure successfully completed.

SQL> select * from student;

     RNO       S1        S2        S3        S4     TOTAL  PERCENTAGE
---------- ---------- ---------- ---------- ---------- ---------- ------
    10011      56        78        79        56       269         67
    10012      45        67        34        58       204         51
    10013      76        86        94        58       314         79
    10014      57        48        39        92       236         59
```

EXERCISE 8

Write a PL/SQL code to display employee number, name and basic of 5 highest paid employees.

SOLUTION:

```
SQL> select * from employee;

    EMP_NO EMPLOYEE_NAME        STREET               CITY
---------- -------------------- -------------------- ------------------
         1 rajesh               first cross          gulbarga
         2 paramesh             second cross         bidar
         3 pushpa               ghandhi road         banglore
         4 vijaya               shivaji nagar        manglore
         5 keerthi              anand sagar street   bijapur
         6 raghu                navneeth cross       Gulbarga

SQL> select * from employee_salary;

    EMP_NO  BASIC     HRA   DA TOTAL_DEDUCTION NET_SALARY  GROSS_SALARY
---------- ---------- ---------- ---------- --------------- ----------
     2      15000    4000   1000      5000      15000         20000
     1      31000    8000   1000      5000      35000         40000
     3      14000    4000   1000      5000      15000         19000
     4      14000    4000   1000      5000      15000         19000
     5      13000    4000   1000      5000      15000         18000
     6      12000    3000    800      4000      11800         15800

SQL> get e:/p8.sql;
  1  declare
  2  i number:=0;
  3  cursor ec is select employee.emp_no,employee_name,basic from
     employee, employee_salary where
     employee.emp_no=employee_salary.emp_no  order by gross_salary desc;
  4  r ec%rowtype;
  5  begin
  6  open ec;
  7  loop
  8  exit when i=5;
  9  fetch ec into r;
 10  dbms_output.put_line(r.emp_no||' '||r.employee_name||' '||r.basic);
 11  i:=i+1;
 12  end loop;
 13  close ec;
 14* end;
 15  .
SQL> /
1 rajesh 31000
2 paramesh 15000
3 pushpa 14000
4 vijaya 14000
5 keerthi 13000

PL/SQL procedure successfully completed.
```

EXERCISE 9

Write a PL/SQL code to calculate the total salary of first n records of emp table. The value of n is passed to cursor as parameter.

SOLUTION:

```
SQL> select * from employee_salary;

    EMP_NO  BASIC      HRA   DA TOTAL_DEDUCTION NET_SALARY  GROSS_SALARY
---------- ---------- ---------- ---------- --------------- ----------
         2      15000    4000   1000       5000      15000         20000
         1      31000    8000   1000       5000      35000         40000
         3      14000    4000   1000       5000      15000         19000
         4      14000    4000   1000       5000      15000         19000
         5      13000    4000   1000       5000      15000         18000
         6      12000    3000    800       4000      11800         15800

SQL> get e:/p9.sql;
  1  declare
  2  no_of_employee number;
  3  total_salary number:=0;
  4  cursor ec(n number) is select * from employee_salary where
     emp_no<=n;
  5  rw ec%rowtype;
  6  begin
  7  no:=&no;
  8  open ec(no_of_employee);
  9  loop
 10  fetch ec into rw;
 11  exit when ec%notfound;
 12  total_salary:=rw.gross_salary+total_salary;
 13  end loop;
 14  dbms_output.put_line('Total salary of'||no||' employee is '
     ||total_salary);
 15* end;
 16  .
SQL> /
Enter value for no_of_employee: 2
old   7: no_of_employee:=& no_of_employee;
new   7: no_of_employee:=2;
Total salary of2 employee is60000

PL/SQL procedure successfully completed.

SQL> /
Enter value for no_of_employee: 3
old   7: no_of_employee:=& no_of_employee;
new   7: no_of_employee:=3;
Total salary of3 employee is79000

PL/SQL procedure successfully completed.
```

EXERCISE 10

Write a PL/SQL code to update the salary of employees who earn less than the average salary.

SOLUTION:

```
SQL> select * from employee_salary;

    EMP_NO  BASIC      HRA   DA TOTAL_DEDUCTION NET_SALARY  GROSS_SALARY
---------- ---------- ---------- ---------- --------------- ----------
     2      15000    4000   1000      5000      15000         20000
     1      31000    8000   1000      5000      35000         40000
     3      14000    4000   1000      5000      15000         19000
     4      14000    4000   1000      5000      15000         19000
     5      13000    4000   1000      5000      15000         18000

 SQL> get e:/p10.sql;
  1  declare
  2  average number;
  3  bs number;
  4  gs number;
  5  diff number;
  6  cursor ec is select * from employee_salary;
  7  rw ec%rowtype;
  8  begin
  9  select avg(basic) into average from employee_salary;
 10  dbms_output.put_line('the average salary is '||average);
 11  open ec;
 12  loop
 13  fetch ec into rw;
 14  exit when ec%notfound;
 15  if(rw.basic<=average)
 16  then
 17  diff:=rw.basic-average;
 18  update employee_salary set basic=average, gross_salary =
     gross_salary + diff where emp_no=rw.emp_no;
 19  select basic,gross_salary into bs,gs from employee_salary where
     emp_no = rw.emp_no;
 20  dbms_output.put_line('the emploee number is '||rw.emp_no);
 21  dbms_output.put_line('old basic ='||rw.basic||'old gross_salary =’
     || rw.gross_salary);
 22  dbms_output.put_line('updated new basic ='||bs||' new gross salary
     is ='||gs);
 23  end if;
 24  end loop;
 25* end;
 26  .

SQL> /

the average salary is 17400

the emploee number is 2
old basic =15000 old gross_salary=20000
updated new basic =17400 new gross salary is =17600
```

```
the emploee number is 3
old basic =14000 old gross_salary=19000
updated new basic =17400 new gross salary is =15600

the emploee number is 4
old basic =14000 old gross_salary=19000
updated new basic =17400 new gross salary is =15600

the emploee number is 5
old basic =13000 old gross_salary=18000
updated new basic =17400 new gross salary is =13600

PL/SQL procedure successfully completed.
```

EXERCISE 11

Write a row trigger to insert the existing values of the salary table in to a new table when the salary table is updated.

SOLUTION:

```
SQL> select * from employee_salary;

    EMP_NO  BASIC     HRA   DA TOTAL_DEDUCTION NET_SALARY  GROSS_SALARY
---------- ---------- ---------- ---------- --------------- ----------
     2      15000    4000   1000       5000      15000         20000
     1      31000    8000   1000       5000      35000         40000
     3      14000    4000   1000       5000      15000         19000
     4      14000    4000   1000       5000      15000         19000
     5      13000    4000   1000       5000      15000         18000


SQL> get e:/p11.sql;
  1  create or replace trigger t
  2  after update on employee_salary
  3  for each row
  4  begin
  5  insert into backup values
     (:old.emp_no,:old.gross_salary,:new.gross_salary);
  6* end;
SQL> /

Trigger created.

SQL> update employee_salary set gross_salary=44000 where emp_no=1;

1 row updated.

SQL> select * from backup;

     EMPNO OLD_GROSS_SALARY NEW_GROSS_SALARY
---------- ---------------- ----------------
         1           40000            44000

SQL> update employee_salary set gross_salary=20000 where emp_no=2;

1 row updated.

SQL> select * from backup;

     EMPNO OLD_GROSS_SALARY NEW_GROSS_SALARY
---------- ---------------- ----------------
         1           40000            44000
         2           17600            20000

SQL> update employee_salary set gross_salary=48000 where emp_no=1;

1 row updated.
```

```
SQL> select * from backup;

     EMPNO OLD_GROSS_SALARY NEW_GROSS_SALARY
---------- ---------------- ----------------
         1            40000            44000
         2            17600            20000
         1            44000            48000
```

EXERCISE 12

Write a trigger on the employee table which shows the old values and new values of Ename after any updations on ename on Employee table.

SOLUTION:

```
SQL> select * from employee;

    EMP_NO EMPLOYEE_NAME        STREET               CITY
---------- -------------------- -------------------- ------------------
         1 rajesh               first cross          gulbarga
         2 paramesh             second cross         bidar
         3 pushpa               ghandhi road         banglore
         4 vijaya               shivaji nagar        manglore
         5 keerthi              anand sagar street   bijapur

SQL> get e:/plsql/l12.sql;
  1  create or replace trigger show
  2  before update on employee
  3  for each row
  4  begin
  5  dbms_output.put_line('the old name was :');
  6  dbms_output.put_line(:old.employee_name);
  7  dbms_output.put_line('the updated new name is :');
  8  dbms_output.put_line(:new.employee_name);
  9* end;
SQL> /

Trigger created.

SQL> update employee set employee_name='kiran' where emp_no=1;
the old name was :
rajesh
the updated new name is :
kiran

1 row updated.

SQL> select * from employee;

    EMP_NO EMPLOYEE_NAME        STREET               CITY
---------- -------------------- -------------------- -------------------
         1 kiran                first cross          gulbarga
         2 paramesh             second cross         bidar
         3 pushpa               ghandhi road         banglore
         4 vijaya               shivaji nagar        manglore
         5 keerthi              anand sagar street   bijapur
```

EXERCISE 13

Writ a PL/SQL procedure to find the number of students ranging from 100-70%, 69-60%, 59-50% & below 49% in each course from the student_course table given by the procedure as parameter.

SOLUTION:

```
SQL> select * from student_enrollment;

   ROLL_NO COURSE     COURSE_COD        SEM TOTAL_MARKS PERCENTAGE
---------- ---------- ---------- ---------- ----------- ----------
       111 cs         1001                1         300         50
       112 cs         1001                1         400         66
       113 is         1002                1         465         77
       114 is         1002                1         585         97

SQL> get e:/p13.sql;
  1  create or replace procedure rank(crc varchar)
  2  is
  3  dis number:=0;
  4  first number:=0;
  5  sec number:=0;
  6  pass number:=0;
  7  cursor st is select * from student_enrollment;
  8  r st%rowtype;
  9  begin
 10  open st;
 11  loop
 12  fetch st into r;
 13  exit when st%notfound;
 14  if(r.course=crc)
 15  then
 16  if(r.percentage>=70 and r.percentage<=100)
 17  then
 18  dis:=dis+1;
 19  end if;
 20   if(r.percentage>=60 and r.percentage<70)
 21  then
 22  first:=first+1;
 23  end if;
 24  if(r.percentage>=50 and r.percentage<60)
 25  then
 26  sec:=sec+1;
 27  end if;
 28  if(r.percentage>=35 and r.percentage<50)
 29  then
 30  pass:=pass+1;
 31  end if;
 32  end if;
 33  end loop;
 34  close st;
 35  dbms_output.put_line('distinction is '||dis);
 36  dbms_output.put_line('first class is '||first);
 37  dbms_output.put_line('second class is '||sec);
 38  dbms_output.put_line('just pass is '||pass);
 39* end;
 40  .
SQL> /

Procedure created.

SQL> exec rank('cs');
distinction is 0
first class is 1
second class is 1
just pass is 0

PL/SQL procedure successfully completed.

SQL> exec rank('is');
distinction is 2
first class is 0
second class is 0
just pass is 0

PL/SQL procedure successfully completed.
```

EXERCISE 14

Create a store function that accepts 2 numbers and returns the addition of passed values. Also write the code to call your function.

SOLUTION:

```
SQL> get e:/p14.sql;
  1  create or replace function addition(a number,b number)
  2  return number
  3  is
  4  begin
  5  dbms_output.put('the sum of '||a||' and '||b||' is :');
  6  return (a+b);
  7* end;
  8  .
SQL> /

Function created.

SQL> begin
  2  dbms_output.put_line(addition(6,78));
  3  end;
  4  .
SQL> /
the sum of 6 and 78 is: 84

PL/SQL procedure successfully completed.
```

EXERCISE 15

Write a PL/SQL function that accepts department number and returns the total salary of the department. Also write a function to call the function.

SOLUTION:

```
SQL> select * from works;

    EMP_NO COMPANY_NAME  JOINING_D DESIGNATION       SALARY       DEPTNO
---------- -------------------- --------- -------------------- ---------
       1    abc         23-NOV-00  project lead        40000        1
       2    abc         25-DEC-10  software engg       20000        2
       3    abc         15-JAN-11  software engg       1900         1
       4    abc         19-JAN-11  software engg       19000        2
       5    abc         06-FEB-11  software engg       18000        1


SQL> get e:/plsql/p15.sql;
  1  create or replace function tot_sal_of_dept(dno number)
  2  return number
  3  is
  4  tot_sal number:=0;
  5  begin
  6  select sum(salary) into tot_sal from works where deptno=dno;
  7  return tot_sal;
  8* end;
SQL> .
SQL> /

Function created.

SQL> begin
  2  dbms_output.put_line('Total salary of DeptNo 1 is :' ||
     tot_sal_of_dept(1));
  3  end;
  4  .

SQL> set serveroutput on;
SQL> /
Total salary of DeptNo 1 is :77000

PL/SQL procedure successfully completed.

SQL> begin
  2  dbms_output.put_line('total salary of dept 2 is
:'||tot_sal_of_dept(2));
  3  end;
  4  .
SQL> /
Total salary of DeptNo 2 is :39000

PL/SQL procedure successfully completed.
```

EXERCISE 16

Write a PL/SQL code to create,

a) Package specification
b) Package body.

For the insert, retrieve, update and delete operations on a student table.

SOLUTION:

```
SQL> get e:/plsql/l16p.sql;
  1  create or replace package alloperation
  2  is
  3  procedure forinsert(rno number,sname varchar,crc varchar,gen
     varchar);
  4  procedure forretrive(rno number);
  5  procedure forupdate(rno number,sname varchar);
  6  procedure fordelete(rno number);
  7* end alloperation;
SQL> .
SQL> /

Package created.

SQL> get e:/plsql/l16pbody.sql;
  1  create or replace package body alloperation
  2  is
  3  procedure forinsert(rno number,sname varchar,crc varchar,gen
     varchar)
  4  is
  5  begin
  6  insert into student values(rno,sname,crc,gen);
  7  end forinsert;
  8  procedure forretrive(rno number)
  9  is
 10  sname student.student_name%type;
 11  crc student.course%type;
 12  gen student.gender%type;
 13  begin
 14  select student_name,course,gender into sname,crc,gen
 15  from student where roll_no=rno;
 16  dbms_output.put_line(sname||' '||crc||' '||gen);
 17  end forretrive;
 18  procedure forupdate(rno number,sname varchar)
 19  is
 20  begin
 21  update student set student_name=sname where roll_no=rno;
 22  end forupdate;
 23  procedure fordelete(rno number)
 24  is
 25  begin
 26  delete student where roll_no=rno;
 27  end fordelete;
 28* end alloperation;
 29  .
SQL> /
```

```
Package body created.

SQL> select * from student;

   ROLL_NO STUDENT_NAME         COURS GENDER
---------- -------------------- ----- ----------
       111 ravi                 cs    male
       112 praveen              cs    male
       113 bhuvana              is    female
       114 apparna              is    female

SQL> begin
  2  alloperation.forinsert(444,'vivekananda','ec','male');
  3  alloperation.forretrive(444);
  4  alloperation.forupdate(111,'swamy');
  5  end;
  6  .
SQL> /

vivekananda ec male

PL/SQL procedure successfully completed.

SQL> select * from student;

   ROLL_NO STUDENT_NAME         COURS GENDER
---------- -------------------- ----- ----------
       111 swamy                cs    male
       112 praveen              cs    male
       113 bhuvana              is    female
       114 apparna              is    female
       444 vivekananda          ec    male

SQL> begin
  2  alloperation.fordelete(444);
  3  end;
  4  .
SQL> /

PL/SQL procedure successfully completed.

SQL> select * from student;

   ROLL_NO STUDENT_NAME         COURS GENDER
---------- -------------------- ----- ----------
       111 swamy                cs    male
       112 praveen              cs    male
       113 bhuvana              is    female
       114 apparna              is    female
```