1) Perform DDL and DML Commands on student(rno,name,marks) table.

```
SQL> create table students
   (
   rollno varchar2(30),
   name varchar2(30),
   branch varchar2(15)
  );
```

To View Schema

```
SQL> describe students
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                             VARCHAR2(30)
 NAME                                               VARCHAR2(30)
 BRANCH                                             VARCHAR2(15)
```

Altering the table

```
SQL> alter table students
  2  add age integer;

Table altered.

SQL> alter table students
  2  drop column branch;

Table altered.

SQL> desc students
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                             VARCHAR2(30)
 NAME                                               VARCHAR2(30)
 AGE                                                NUMBER(38)
```

Truncate

```
SQL> truncate table students;

Table truncated.
```

Rename a table

```
SQL> rename students to stu_details2  ;

Table renamed.
```

Dropping table

```
SQL> drop table stu_details;
```

Table dropped.

2A Find the names of sailors who have reserved a red boat using IN operator.

```
SQL> select s.sname from sailors s
  2  where s.sid IN(select r.sid from reserves r where r.bid IN(select b.bid from boats b where
b.bcolor='red'));

SNAME
--------------------
dustin
lubber
Horatio
```

2) Find the names of sailors who does not have reserved a red boat using NOT IN operator.

```
SQL> select s.sname from sailors s
  2  where s.sid NOT IN(select r.sid from reserves r where r.bid IN(select b.bid from boats b
where b.bcolor='red'));

SNAME
--------------------
brutus
Andy
Zobra
Ravi
Art
bob
Rusty
Leela
```

Find the names of sailors who have reserved boat number: 103 using EXISTS

```
SQL> select s.sname from sailors s
  2  where EXISTS(select * from reserves r where r.sid=s.sid and r.bid=103);

SNAME
--------------------
dustin
lubber
Ravi
```

2) Find the names of sailors who not have reserved boat number: 103 using NOT EXISTS

```
SQL> select s.sname from sailors s
  2  where NOT EXISTS(select * from reserves r where r.sid=s.sid and r.bid=103);

SNAME
--------------------
Zobra
Art
Horatio
Rusty
Andy
Leela
```

brutus
bob
8 rows selected

2b)Write a pl/sql program to print sequence of n numbers using for loop.

```
Using for loop
SQL> declare
  2  a int:=1;
  3  n int:=&n;
  4  begin
  5  for a in 1..n
  6  loop
  7  dbms_output.put_line(a);
  8  end loop;
  9  end;
 10  /
Enter value for n: 6
old   3: n int:=&n;
new   3: n int:=6;
1
2
3
4
5
6

PL/SQL procedure successfully completed
```

3a)write a pl/sql program to print student progress by taking input student grade using CASE statement

```
DECLARE

    grd CHAR(1);

  BEGIN

    -- Accept value for grade

    grd := '&new_grd';

  CASE grd

    WHEN 'A' THEN dbms_output.Put_line('Your Grade is:
Outstanding');

    WHEN 'B' THEN dbms_output.Put_line('Your Grade is: Excellent');

    WHEN 'C' THEN dbms_output.Put_line('Your Grade is: Very Good');
```

```
    WHEN 'D' THEN dbms_output. Put_line('Your Grade is: Average');
    WHEN 'F' THEN dbms_output.Put_line('Your Grade is: Poor');
    ELSE dbms_output.Put_line('No such grade in the list.');
  END CASE;
  END;
/
```

Copy

Sample Output:

```
Enter value for new_grd: D
old   5:      grd := '&new_grd';
new   5:      grd := 'D';
Your Grade is: Average

PL/SQL procedure successfully completed.
```

4a) write a pl/sql program to handle (built in exception) zero divide.

```
SQL> declare
  2  id number:=12;
  3  BEGIN
  4  id:=12/0;
  5  exception
  6  when zero_divide then
  7  dbms_output.put_line('Divide by zero');
  8  end;
  9  /
```

Divide by zero

PL/SQL procedure successfully completed.

4b) create a explicit cursor using for loop to print sailor id and name.

Program:

```
SQL> declare
2 cursor sail_cur is
3 select sid,sname from sailors;
4 ab sail_cur%rowtype;
5 begin
6 for ab in sail_cur
7 loop
8 dbms_output.put_line(ab.sid||' '||ab.sname);
```

```
9 end loop;
10 end;
11 /
```

Output:

```
421 leela
22 dustin
29 brutus
31 lubber
32 andy
64 horatio
71 zobra
85 art
74 ravi
95 bob
58 rusty
PL/SQL procedure successfully completed.
```

5a)write queries using string functions(concat, lpad, rpad,reverse, upper,lower)

STRING FUNCTIONS:

```
SQL> select concat('aditya','engg') from dual;

CONCAT('AD
----------
adityaengg

SQL> select concat(concat('aditya','engg'),'college') from dual;

CONCAT(CONCAT('AD
-----------------
adityaenggcollege

SQL> select 'aditya'||'engg' from dual;

'ADITYA'||
----------
adityaengg

SQL> select lpad('aditya',15,'*')as lpad from dual;

LPAD
---------------
61616161*aditya

SQL> select rpad('aditya',15,'*')as rpad from dual;

RPAD
---------------
aditya61616161*
```

```
SQL> select ltrim('123123123rama123','123')from dual;

LTRIM('
-------
rama123

SQL> select rtrim('123123123rama123','123')from dual;

RTRIM('123123
-------------
123123123rama

SQL> select upper('aditya') from dual;

UPPER(
------
ADITYA

SQL> select lower('ADITYA') from dual;

LOWER(
------
aditya

SQL> select length('aditya') from dual;

LENGTH('ADITYA')
----------------
               6
```

5b) Write a pl/sql program to print sequence of n numbers using while loop

```
SQL> declare
  2  a int:=1;
  3  n int:=&n;
  4  begin
  5  for a in 1..n
  6  loop
  7  dbms_output.put_line(a);
  8  end loop;
  9  end;
 10  /
Enter value for n: 6
old   3: n int:=&n;
new   3: n int:=6;
1
2
3
4
5
6
```

PL/SQL procedure successfully completed.

6a) write queries using set operations on sailors,reserves,boats.

Program for creatind sailors table:

```
SQL> create table sailors
  2  (
  3  sid integer,
  4  sname varchar2(33),
  5  age number(3,1),
  6  rating integer,
  7  constraints pk_sailors primary key(sid)
  8  );

Table created.
SQL> create table boats
  2  (
  3  bid integer,
  4  bname varchar2(20),
  5  bcolor varchar2(20),
  6  constraints pk_boats primary key(bid)
  7  );

Table created.
SQL> create table reserves
  2  (
  3  sid integer,
  4  bid integer,
  5  rdate date,
  6  constraints fk_sailors foreign key (sid) references sailors(sid),
  7  constraints fk_boats foreign key(bid) references boats(bid)
  8  );

Table created.
SQL> insert into sailors values(&sid,'&sname',&age,&rating);
Enter value for sid: 22
Enter value for sname: dustin
Enter value for age: 45
Enter value for rating: 7
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(22,'dustin',45,7)

1 row created.
SQL> /
Enter value for sid: 29
Enter value for sname: brutus
Enter value for age: 33
Enter value for rating: 1
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
```

```
new   1: insert into sailors values(29,'brutus',33,1)

1 row created.

SQL> /
Enter value for sid: 31
Enter value for sname: lubber
Enter value for age: 55.5
Enter value for rating: 8
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(31,'lubber',55.5,8)

1 row created.

SQL> /
Enter value for sid: 32
Enter value for sname: andy
Enter value for age: 25.5
Enter value for rating: 8
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(32,'andy',25.5,8)

1 row created.
SQL> /
Enter value for sid: 64
Enter value for sname: horatio
Enter value for age: 35
Enter value for rating: 7
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(64,'horatio',35,7)

1 row created.

SQL> /
Enter value for sid: 71
Enter value for sname: zobra
Enter value for age: 16
Enter value for rating: 10
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(71,'zobra',16,10)

1 row created.

SQL> /
Enter value for sid: 74
Enter value for sname: ravi
Enter value for age: 35
Enter value for rating: 9
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(74,'ravi',35,9)
```

1 row created.

```
SQL> /
Enter value for sid: 85
Enter value for sname: art
Enter value for age: 25
Enter value for rating: 3
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(85,'art',25,3)
```

1 row created.

```
SQL> /
Enter value for sid: 95
Enter value for sname: bob
Enter value for age: 63
Enter value for rating: 3
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(95,'bob',63,3)
```

1 row created.

```
SQL> /
Enter value for sid: 58
Enter value for sname: rusty
Enter value for age: 35
Enter value for rating: 10
old   1: insert into sailors values(&sid,'&sname',&age,&rating)
new   1: insert into sailors values(58,'rusty',35,10)
```

1 row created.

```
SQL> select * from sailors;
```

SID	SNAME	AGE	RATING
22	dustin	45	7
29	brutus	33	1
31	lubber	55.5	8
32	andy	25.5	8
64	horatio	35	7
71	zobra	16	10
74	ravi	35	9
85	art	25	3
95	bob	63	3
58	rusty	35	10

10 rows selected.

```
SQL> insert into boats values(&bid,'&bname','&bcolor');
Enter value for bid: 101
Enter value for bname: interlake
Enter value for bcolor: blue
old   1: insert into boats values(&bid,'&bname','&bcolor')
new   1: insert into boats values(101,'interlake','blue')

1 row created.

SQL> /
Enter value for bid: 102
Enter value for bname: interlake
Enter value for bcolor: red
old   1: insert into boats values(&bid,'&bname','&bcolor')
new   1: insert into boats values(102,'interlake','red')

1 row created.

SQL> /
Enter value for bid: 103
Enter value for bname: clipper
Enter value for bcolor: green
old   1: insert into boats values(&bid,'&bname','&bcolor')
new   1: insert into boats values(103,'clipper','green')

1 row created.

SQL> /
Enter value for bid: 104
Enter value for bname: marine
Enter value for bcolor: red
old   1: insert into boats values(&bid,'&bname','&bcolor')
new   1: insert into boats values(104,'marine','red')

1 row created.

SQL> select * from boats;

       BID BNAME                          BCOLOR
---------- ------------------------------ ------------------------------
       101 interlake                      blue
       102 interlake                      red
       103 clipper                        green
       104 marine                         red

SQL> insert into reserves values(&sid,&bid,'&rdate');
Enter value for sid: 22
Enter value for bid: 101
Enter value for rdate: 10-oct-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
```

```
new   1: insert into reserves values(22,101,'10-oct-98')

1 row created.

SQL> /
Enter value for sid: 22
Enter value for bid: 102
Enter value for rdate: 10-oct-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(22,102,'10-oct-98')

1 row created.

SQL> /
Enter value for sid: 22
Enter value for bid: 103
Enter value for rdate: 10-aug-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(22,103,'10-aug-98')

1 row created.

SQL> /
Enter value for sid: 22
Enter value for bid: 104
Enter value for rdate: 10-july-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(22,104,'10-july-98')

1 row created.

SQL> /
Enter value for sid: 31
Enter value for bid: 103
Enter value for rdate: 11-jun-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(31,103,'11-jun-98')

1 row created.

SQL> /
Enter value for sid: 31
Enter value for bid: 104
Enter value for rdate: 11-dec-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(31,104,'11-dec-98')

1 row created.

SQL> /
```

```
Enter value for sid: 31
Enter value for bid: 102
Enter value for rdate: 11-oct-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(31,102,'11-oct-98')

1 row created.

SQL> /
Enter value for sid: 64
Enter value for bid: 101
Enter value for rdate: 09-may-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(64,101,'09-may-98')

1 row created.

SQL> /
Enter value for sid: 64
Enter value for bid: 102
Enter value for rdate: 09-aug-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(64,102,'09-aug-98')

1 row created.

SQL> /
Enter value for sid: 74
Enter value for bid: 103
Enter value for rdate: 09-aug-98
old   1: insert into reserves values(&sid,&bid,'&rdate')
new   1: insert into reserves values(74,103,'09-aug-98')

1 row created.

SQL> select * from reserves;

       SID        BID RDATE
---------- ---------- ---------
        22        101 10-OCT-98
        22        102 10-OCT-98
        22        103 10-AUG-98
        22        104 10-JUL-98
        31        103 11-JUN-98
        31        104 11-DEC-98
        31        102 11-OCT-98
        64        101 09-MAY-98
        64        102 09-AUG-98
        74        103 09-AUG-98
10 rows selected.
```

Set Operators:

UNION:

Find the sailors name who have reserved a red boat or a green boat.

```
SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and
b.bcolor='red'
  2  UNION
  3  select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid
and b1.bcolor='green';

SNAME
--------------------
dustin
horatio
lubber
ravi
```

UNION ALL:

Find the sailors name who have reserved a red boat or a green boat.

```
SQL>  select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and
b.bcolor='red'
  2   UNION ALL
  3   select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and
b1.bid=r1.bid and b1.bcolor='green';
SNAME
--------------------
dustin
dustin
lubber
lubber
horatio
dustin
lubber
ravi
8 rows selected.
```

Without using union:

Find the sailors name who have reserved a red boat or a green boat.

```
SQL>  select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and
(b.bcolor='red' or b.bcolor='green');
SNAME
--------------------
dustin
dustin
dustin
lubber
lubber
lubber
horatio
ravi
8 rows selected.
```

INTERSECT:

Find the sailors name who have reserved a red boat and a green boat.

```
SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and
b.bcolor='red'
  2  INTERSECT
  3  select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid
and b1.bcolor='green';

SNAME
--------------------
dustin
lubber
```

Without using intersect:

```
SQL>select s.sname from sailors s,boats b, boats b1,reserves r,reserves r1 where s.sid=r.sid
and (b.bid=r.bid and b1.bid=r1.bid) and (b.bcolor='red' and b1.bcolor='green';

SNAME
--------------------
dustin
dustin
lubber
lubber
```

MINUS:

Find the sailors name who have reserved a red boat but not a green boat.

```
SQL>  select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and
b.bcolor='red'
  2  MINUS
  3  select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid
and b1.bcolor='green';

SNAME
--------------------
horatio
```

6 b) Create a stored procedure to add two numbers.

```
SQL> declare
  2  a integer;
  3  b integer;
  4  c integer;
  5  begin
  6  a:=2;
  7  b:=3;
  8  c:=a+b;
  9  dbms_output.put_line('value of a is'||a);
 10  dbms_output.put_line('value of b is'||b);
 11  dbms_output.put_line('value of c is'||c);
 12  end;
 13  /
value of a is2
value of b is3
value of c is5
```

PL/SQL procedure successfully completed.

7a)Writequeriesusingdatefunctions(months_between,last_day,next_day,add_months)

DATE FUNCTIONS:

```
SQL> select sysdate from dual;

SYSDATE
---------
07-OCT-21

SQL> select sysdate+1 from dual;

SYSDATE+1
---------
08-OCT-21

SQL> select sysdate-1 from dual;

SYSDATE-1
---------
06-OCT-21

SQL> select extract(year from sysdate) from dual;

EXTRACT(YEARFROMSYSDATE)
------------------------
                    2021

SQL> select extract(month from sysdate) from dual;

EXTRACT(MONTHFROMSYSDATE)
-------------------------
                      10

SQL> select extract(day from sysdate) from dual;

EXTRACT(DAYFROMSYSDATE)
-----------------------
                     7

SQL> select to_char(sysdate,'yyyy/mm/dd') from dual;

TO_CHAR(SY
----------
```

```
2021/10/07

SQL> select to_char(sysdate,'HH:MM:SS') from dual;

TO_CHAR(
--------
05:10:24

SQL> select add_months(sysdate,2) from dual;

ADD_MONTH
---------
07-DEC-21

SQL> select next_day(sysdate,'Thursday') from dual;

NEXT_DAY(
---------
14-OCT-21

SQL> select next_day('10-dec-2019','Tuesday') from dual;

NEXT_DAY(
---------
17-DEC-19

SQL> select last_day(sysdate) from dual;

LAST_DAY(
---------
31-OCT-21

SQL> select months_between(to_date('09-dec-2020','dd-mm-yyyy'),to_date('09-dec-2019','dd-mm-
yyyy'))
from dual;

MONTHS_BETWEEN(TO_DATE('09-DEC-2020','DD-MM-YYYY'),TO_DATE('09-DEC-
2019','DD-MM-
--------------------------------------------------------------------------------
```


7b) Create a function to add two numbers.

```
SQL> create or replace function add_c(a in number,b in number)
  2  return number
  3  as
  4  c number;
  5  begin
  6  c:=a+b;
  7  return c;
```

```
  8  end;
  9  /

Function created.

SQL> declare
  2  d number;
  3  begin
  4  d:=add_c(10,20);
  5  dbms_output.put_line(d);
  6  end;
  7  /
30
PL/SQL procedure successfully completed.
SQL> select add_c(20,30) from dual;
ADD_C(20,30)
------------
          50
```

8 b) Perform cross, natural, inner,outer join on a table.

```
SQL> create table th1
  2  (
  3  rno integer,
  4  name varchar2(30),
  5  marks integer
  6  );

Table created.

SQL> create table th2
  2  (
  3  rno integer,
  4  fee integer
  5  );

Table created.

SQL> insert into th1 values(&rno,'&name',&marks);
Enter value for rno: 501
Enter value for name: abhi
Enter value for marks: 50
old   1: insert into th1 values(&rno,'&name',&marks)
new   1: insert into th1 values(501,'abhi',50)

1 row created.

SQL> /
Enter value for rno: 502
Enter value for name: ravi
Enter value for marks: 40
old   1: insert into th1 values(&rno,'&name',&marks)
```

```
new   1: insert into th1 values(502,'ravi',40)

1 row created.

SQL> /
Enter value for rno: 503
Enter value for name: suma
Enter value for marks: 30
old   1: insert into th1 values(&rno,'&name',&marks)
new   1: insert into th1 values(503,'suma',30)

1 row created.

SQL> /
Enter value for rno: 504
Enter value for name: raju
Enter value for marks: 35
old   1: insert into th1 values(&rno,'&name',&marks)
new   1: insert into th1 values(504,'raju',35)

1 row created.

SQL> /
Enter value for rno: 505
Enter value for name: ramu
Enter value for marks: 45
old   1: insert into th1 values(&rno,'&name',&marks)
new   1: insert into th1 values(505,'ramu',45)

1 row created.

SQL> insert into th2 values(&rno,&fee);
Enter value for rno: 501
Enter value for fee: 3000
old   1: insert into th2 values(&rno,&fee)
new   1: insert into th2 values(501,3000)

1 row created.

SQL> /
Enter value for rno: 502
Enter value for fee: 2000
old   1: insert into th2 values(&rno,&fee)
new   1: insert into th2 values(502,2000)

1 row created.

SQL> /
Enter value for rno: 503
Enter value for fee: 1500
```

```
old   1: insert into th2 values(&rno,&fee)
new   1: insert into th2 values(503,1500)

1 row created.

SQL> /
Enter value for rno: 504
Enter value for fee: 4000
old   1: insert into th2 values(&rno,&fee)
new   1: insert into th2 values(504,4000)

1 row created.
```

```
SQL> select * from th1;

      RNO NAME                           MARKS
---------- ------------------------------ ----------
      501 abhi                           50
      502 ravi                           40
      503 suma                            30
      504 raju                           35
      505 ramu                            45
```

```
SQL> select * from th2;

      RNO        FEE
---------- ----------
      501       3000
      502       2000
      503       1500
      504       4000
```

Inner Join:

```
SQL> select * from th1 inner join th2 on th1.rno=th2.rno;

      RNO NAME                           MARKS       RNO        FEE
---------- ------------------------------ ---------- ---------- ----------
      501 abhi                           50        501       3000
      502 ravi                           40        502       2000
      503 suma                            30        503       1500
      504 raju                           35        504       4000
```

```
SQL>  select * from th1 join th2 on th1.rno=th2.rno;

      RNO NAME                           MARKS       RNO        FEE
---------- ------------------------------ ---------- ---------- ----------
      501 abhi                           50        501       3000
      502 ravi                           40        502       2000
```

503	suma	30	503	1500
504	raju	35	504	4000

Outer Join:

Left outer join:

SQL> select * from th1 left outer join th2 on th1.rno=th2.rno;

RNO	NAME	MARKS	RNO	FEE
501	abhi	50	501	3000
502	ravi	40	502	2000
503	suma	30	503	1500
504	raju	35	504	4000
505	ramu	45		

Right outer join:

SQL> select * from th2 right outer join th1 on th2.rno=th1.rno;

RNO	FEE	RNO	NAME	MARKS
501	3000	501	abhi	50
502	2000	502	ravi	40
503	1500	503	suma	30
504	4000	504	raju	35
		505	ramu	45

Natural Join:

SQL> select * from th1 natural join th2;

RNO	NAME	MARKS	FEE
501	abhi	50	3000
502	ravi	40	2000
503	suma	30	1500
504	raju	35	4000

SQL> select * from th1 cross join th2;

RNO	NAME	MARKS	RNO	FEE
501	abhi	50	501	3000
502	ravi	40	501	3000
503	suma	30	501	3000
504	raju	35	501	3000
505	ramu	45	501	3000
501	abhi	50	502	2000
502	ravi	40	502	2000
503	suma	30	502	2000
504	raju	35	502	2000
505	ramu	45	502	2000

```
501 abhi                          50       503       1500

      RNO NAME                          MARKS       RNO       FEE
---------- ------------------------------ ---------- ---------- ----------
      502 ravi                          40       503       1500
      503 suma                           30       503       1500
      504 raju                          35       503       1500
      505 ramu                           45       503       1500
      501 abhi                          50       504       4000
      502 ravi                          40       504       4000
      503 suma                           30       504       4000
      504 raju                          35       504       4000
      505 ramu                           45       504       4000

20 rows selected.

SQL> select * from th1,th2;

      RNO NAME                          MARKS       RNO       FEE
---------- ------------------------------ ---------- ---------- ----------
      501 abhi                          50       501       3000
      502 ravi                          40       501       3000
      503 suma                           30       501       3000
      504 raju                          35       501       3000
      505 ramu                           45       501       3000
      501 abhi                          50       502       2000
      502 ravi                          40       502       2000
      503 suma                           30       502       2000
      504 raju                          35       502       2000
      505 ramu                           45       502       2000
      501 abhi                          50       503       1500

      RNO NAME                          MARKS       RNO       FEE
---------- ------------------------------ ---------- ---------- ----------
      502 ravi                          40       503       1500
      503 suma                           30       503       1500
      504 raju                          35       503       1500
      505 ramu                           45       503       1500
      501 abhi                          50       504       4000
      502 ravi                          40       504       4000
      503 suma                           30       504       4000
      504 raju                          35       504       4000
      505 ramu                           45       504       4000

20 rows selected.
SQL> select t1.rno,t1.name from th1 t1,th2 t2 where t1.rno=t2.rno;

      RNO NAME
---------- ------------------------------
      501 abhi
```

502 ravi
503 suma
504 raju

9a) create a view on student relation and perform DDL and DML commands.

```
SQL> create table students
  2  (
  3  rollno varchar2(30),
  4  name varchar2(30),
  5  branch varchar2(15)
  6  );

Table created.
```

To View Schema

```
SQL> describe students
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                             VARCHAR2(30)
 NAME                                               VARCHAR2(30)
 BRANCH                                             VARCHAR2(15)
```

Altering the table

```
SQL> alter table students
  2  add age integer;

Table altered.

SQL> alter table students
  2  drop column branch;

Table altered.

SQL> desc students
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                             VARCHAR2(30)
 NAME                                               VARCHAR2(30)
 AGE                                                NUMBER(38)
```

Truncate

```
SQL> truncate table students;

Table truncated.
```

Rename a table

```
SQL> rename students to stu_details2  ;

Table renamed.
```

Dropping table

```
SQL> drop table stu_details;

Table dropped.

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
STUDENT1                        TABLE
STUDENTS2                       TABLE
STU                            TABLE
STU1                           TABLE
STU2                           TABLE
STUD                            TABLE
STU3                           TABLE
STU4                           TABLE
STU5                           TABLE
BIN$QdFqkisyT1CzNapxV9Kqpw==$0 TABLE
BIN$OYWM4Kc6S3+WIMslmtnp/w==$0 TABLE

11 rows selected.
```

Queries to Retrieve and Change Data: Select, Insert, Delete and Update.

Creating table

```
SQL> create table students
  2  (
  3  rollno varchar2(30),
  4  name varchar2(30)
  5  );

Table created.
```

Inserting Data into the table

```
SQL> insert into students values('20A91A0501','Ravi');

1 row created.

SQL> insert into students values('20A91A0502','Suma');

1 row created.
```

Displaying Data from the table

```
SQL> select * from students;

ROLLNO                         NAME
------------------------------ ------------------------------
20A91A0501                      Ravi
20A91A0502                      Suma
```

```
SQL> select name from students;

NAME
------------------------------
Ravi
Suma

SQL> select * from students where rollno='20A91A0501';

ROLLNO                         NAME
------------------------------ ------------------------------
20A91A0501                      Ravi
```

Deleting a row from the table

```
SQL> delete from students where rollno='20A91A0501';

1 row deleted.
```

Updating a row in the table

```
SQL> update students
  2  set name='Rose'
  3  where rollno='20A91A0502';

1 row updated.
```

9b) Write queries using Group by and having for company table.

4.1 experiment

a) create a student(rollno, name, marks) table using primary and foreign key constraints

Primary key

```
SQL> create table stu2
(
rno integer,
name varchar2(20),
primary key(rno)
);
Table created.

SQL> insert into stu2 values(501,'rani');

1 row created.

SQL> insert into stu2 values(501,'rani');
insert into stu2 values(501,'rani')
*
```

```
ERROR at line 1:
ORA-00001: unique constraint (CSE20561.SYS_C005730) violated


SQL> insert into stu2 values(null,'kamala');
insert into stu2 values(null,'kamala')
                 *
ERROR at line 1:
ORA-01400: cannot insert NULL into ("CSE20561"."STU2"."RNO")

SQL> create table stu5
  2  (
  3  rno integer,
  4  name varchar2(20)
  5  );

Table created.

SQL> alter table stu5
  2  add primary key(rno);

Table altered.

SQL> desc stu5
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RNO                                       NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(20)
```

Foreign key

```
SQL> create table stud
  2  (
  3  rno integer,
  4  fee integer,
  5  foreign key(rno) references stu2(rno)
  6  );

Table created.

SQL> insert into stud values(501,6000);

1 row created.

SQL> insert into stud values(502,8000);
insert into stud values(502,8000)
*
ERROR at line 1:
ORA-02291: integrity constraint (CSE20561.SYS_C005731) violated - parent key
not found
```

10b) Write queries using IS NULL, IS NOT NULL, LIKE operator.

```
SQL> create table customers
  2  (
  3  name varchar2(30),
  4  city varchar2(30),
  5  );

Table created.
SQL> desc customers
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                               VARCHAR2(20)
 CITY                                               VARCHAR2(30)
SQL> insert into customers values('ajay','perry ridge');

1 row created.

SQL> insert into customers values('pavani','downtown');

1 row created.
SQL> insert into customers values('ravi','paris');

1 row created.
SQL> select * from student;

NAME                 CITY
-------------------- ------------------------------
ajay                 perry ridge
pavani               downtown
ravi                 paris

SQL> select * from customers where city like '%idge%';
NAME                 CITY
-------------------- ------------------------------
ajay                 perry ridge

SQL> select * from customers where city like 'p____';
NAME                 CITY
-------------------- ------------------------------
ravi                 paris
```

Is null and Is not null:

```
SQL> desc student

 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                               VARCHAR2(20)
 SNAME                                             VARCHAR2(30)
 AGE                                               NUMBER(38)

SQL> insert into student values(503,'ajay',NULL);
```

1 row created.

SQL> insert into student values(504,'suma',NULL);

1 row created.

SQL> select * from student;

SID	SNAME	AGE
501	akash	21
502	thanmayi	24
503	ajay	
504	suma	

SQL> select sid,sname from student where age IS NULL;

SID	SNAME
503	ajay
504	suma

SQL> select sid,sname from student where age IS NOT NULL;

SID	SNAME
501	akash
502	thanmayi