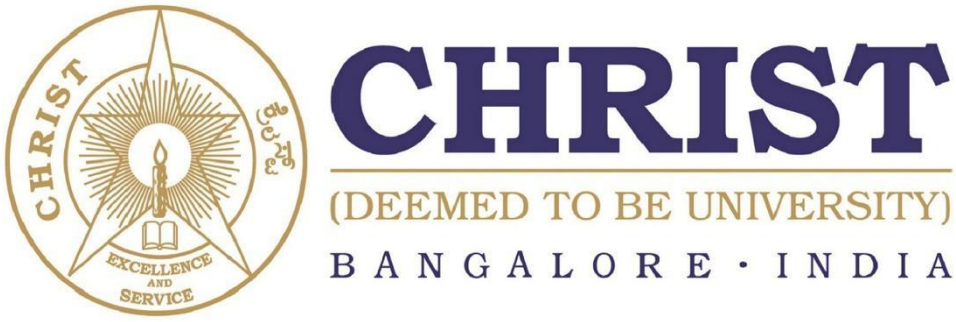

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

Data Base Management Systems

(CS331)

B. Tech Degree – CSE

School of Engineering and Technology,

CHRIST (Deemed to be University),

Kumbalagodu, Bengaluru-560 074

December 2022

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Certificate

This is to certify that …… has successfully completed the record work for Database Management Systems –CS331P in partial fulfillment for the award of Bachelor of Technology in during the year 2021-2022.

Dr. K. Balachandran

HEAD OF DEPARTMENT

FACULTY- IN CHARGE

EXAMINER 1:

EXAMINER 2:

Name :

Register No. :

Examination Center :

Date of Examination :

INDEX

Exp. No	Date	Experiment Name	Page No	Marks	Signature

Ex. No: 1 **Date: 21 – 07 - 2022**

Experiment 1

Consider the following schema for Store Database.

Employee (empno, name, sal)

Store (Store_id, Store_name, Store_ loc)

Write SQL queries to

1. Create tables Employee and Store.
2. Display structure of tables Employee and Store.
3. Insert values into tables Employee and Store.
4. Display the contents of the tables Employee and Store.
5. Display the columns from table Employee and Store.
6. Demonstrate the WHERE clause using tables Employee and Store.
7. Demonstrate the DELETE operation on tables Employee and Store.
8. Demonstrate the ORDER BY clause using table Store.
9. Demonstrate the DROP operation on tables Employee and Store.

Queries:

1. **Create tables Employee and Store.**
   SQL> CREATE TABLE Employee (empno int, name varchar2(10), sal number (10,2));
   Table created.
   SQL> CREATE TABLE Store (Store_id int, Store_name varchar (10),Store_loc varchar (10));
   Table created.
2. **Display structure of tables Employee and Store.**
   SQL> DESC Employee;
   SQL> DESC Store;
3. **Insert values into tables Employee and Store.**
   SQL> INSERT INTO Employee VALUES (001,'RAM',21654);
   1 row created.
   SQL> INSERT INTO Employee VALUES (003,'KAREEM',1644);
   1 row created.
   SQL> INSERT INTO Store VALUES (7704, 'Manna Mart', 'Delhi');
   1 row created.
   SQL> INSERT INTO Store VALUES (5487, 'Neel', 'Bombay');
   1 row created.
4. **Display the contents of the tables Employee and Store.**
   SQL> SELECT * FROM Employee;
   SQL> SELECT * FROM Store;
5. **Display the columns from table Employee and Store.**
   SQL> SELECT name FROM Employee;
   SQL> SELECT Store_loc FROM Store;
6. **Demonstrate the WHERE clause using tables Employee and Store.**
   SQL> SELECT name FROM Employee WHERE empno=002;
   SQL> SELECT Store_loc FROM Store WHERE Store_id=7704;
7. **Demonstrate the DELETE operation on tables Employee and Store.**
   SQL> DELETE Employee WHERE Name='RAM';
   SQL> DELETE Store WHERE Store_loc = 'Bombay';
8. **Demonstrate the ORDER BY clause using table Store.**
   SQL> SELECT Store_loc FROM Store ORDER BY Store_name;
   SQL> SELECT Store_name FROM Store ORDER BY Store_loc ASC;
9. **Demonstrate the DROP operation on tables Employee and Store.**
   SQL> DROP TABLE Store;
   SQL> DROP TABLE employee;

Code:

```
SQL> create table Employee(
  2  empno Int,
  3  name varchar2(10),
  4  sal number(10,2));

Table created.

SQL> select * from Employee;

no rows selected

SQL> desc Employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(38)
 NAME                                               VARCHAR2(10)
 SAL                                                NUMBER(10,2)

SQL> insert into Employee values(001,'RAM',21654);

1 row created.

SQL> insert into Employee values(002,'JOE',15642);

1 row created.

SQL> insert into Employee values(003,'KAREEM',1644);

1 row created.

SQL> select * from Employee;

     EMPNO NAME              SAL
---------- ---------- ----------
         1 RAM             21654
         2 JOE             15642
         3 KAREEM           1644

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(38)
 NAME                                               VARCHAR2(10)
 SAL                                                NUMBER(10,2)

SQL> select NAME from Employee;
```

```
NAME
----------
RAM
JOE
KAREEM

SQL> select NAME from Employee where EMPNO=002;

NAME
----------
JOE

SQL> select SAL from Employee where EMPNO=003;

       SAL
----------
      1644

SQL> delete Employee where NAME='RAM';

1 row deleted.

SQL> select * from Employee;

     EMPNO NAME              SAL
---------- ---------- ----------
         2 JOE             15642
         3 KAREEM           1644

SQL> delete Employee where EMPNO ='3';

1 row deleted.

SQL> select * from Employee;

     EMPNO NAME              SAL
---------- ---------- ----------
         2 JOE             15642

SQL> drop table employee;

Table dropped.

SQL> create table store(store_id int, store_name varchar(10), store_loc varchar(10));

Table created.
```

```
SQL> desc store;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
 STORE_ID                                           NUMBER(38)
 STORE_NAME                                         VARCHAR2(10)
 STORE_LOC                                          VARCHAR2(10)

SQL> insert into store values (7704,'Manna Mart','Delhi');

1 row created.

SQL> insert into store values (1174,'Lulu','Chennai');

1 row created.

SQL> insert into store values (5487,'Neel','Bombay');

1 row created.

SQL> insert into store values (9546,'D-Mart','Banglore');

1 row created.

SQL> select * from store;

  STORE_ID STORE_NAME STORE_LOC
---------- ---------- ----------
      7704 Manna Mart Delhi
      1174 Lulu       Chennai
      5487 Neel       Bombay
      9546 D-Mart     Banglore

SQL> select STORE_NAME from store;

STORE_NAME
----------
Manna Mart
Lulu
Neel
D-Mart

SQL> select STORE_LOC from store WHERE STORE_ID=1174;

STORE_LOC
----------
Chennai
```

```
SQL> select STORE_LOC from store ORDER BY STORE_NAME;

STORE_LOC
----------
Banglore
Chennai
Delhi
Bombay

SQL> select STORE_NAME from store ORDER BY STORE_LOC ASC;

STORE_NAME
----------
D-Mart
Neel
Lulu
Manna Mart

SQL> select STORE_NAME from store ORDER BY STORE_LOC DESC;

STORE_NAME
----------
Manna Mart
Lulu
Neel
D-Mart

SQL> select * from store;

  STORE_ID STORE_NAME STORE_LOC
---------- ---------- ----------
      7704 Manna Mart Delhi
      1174 Lulu       Chennai
      5487 Neel       Bombay
      9546 D-Mart     Banglore

SQL> delete store where STORE_ID = 9546;

1 row deleted.

SQL> select * from store;

  STORE_ID STORE_NAME STORE_LOC
---------- ---------- ----------
      7704 Manna Mart Delhi
      1174 Lulu       Chennai
```

```
      5487 Neel       Bombay

SQL> delete store where STORE_LOC = 'Bombay';

1 row deleted.

SQL> select * from store;

  STORE_ID STORE_NAME STORE_LOC
---------- ---------- ----------
      7704 Manna Mart Delhi
      1174 Lulu       Chennai

SQL> delete store where STORE_NAME = 'Lulu';

1 row deleted.

SQL> select * from store;

  STORE_ID STORE_NAME STORE_LOC
---------- ---------- ----------
      7704 Manna Mart Delhi

SQL> drop table store;

Table dropped.
```

Ex. No: 2 **Date: 28 – 07 - 2022**

Experiment 2

Demonstrate the following Query Concepts / Clauses / Statements / Commands / Operations.

1) Substitution variables
   (SELECT &c_1, &c_2, &c_3…….,&c_n FROM <TABLE_NAME>)
2) Case Manipulation
   LOWER(<COLUMN_NAME>), UPPER(<COLUMN_NAME>)
3) ORDER BY (ASC, DESC)
4) LIKE Command (‘_a%’, ’%a%’, ‘%a’)
5) Aggregate functions (SUM,MAX,MIN,AVG,COUNT)
6) ROUND Function
7) DATE datatype
8) IN and NOT IN Operators
9) IS NULL and IS NOT NULL operator
10) AS command
11) SET, UPDATE and ALTER

Queries:

10. **Create tables samp having attributes regno and name.**
    SQL> CREATE TABLE samp (regno number, name varchar(15));
11. **Demonstrate Substitution Variables**
    SQL> INSERT INTO samp VALUES (&No, ‘&name’);
    SQL> SELECT &c1, &c2, &c3 from employees;
    Enter value for c1: Salary
    Enter value for c2: last_name
    Enter value for c3: job_id
12. **Demonstrate Case Manipulation**
    SQL> select first_name, lower(first_name) from employees;
    SQL> select first_name,lower(first_name),upper(first_name),initcap(first_name) from employees;
13. **Display Employee ID, First Name and Salary of employee whose First Name is ‘Michael’.**
    SQL> select first_name, employee_id, salary from employees where first_name='Michael';
14. **Demonstrate the ORDER BY clause**
    SQL> select first_name, salary from employees order by salary asc;
    SQL> select first_name, salary from employees order by salary desc;
15. **Demonstrate the LIKE Command**
    SQL> select first_name from employees where first_name like 'A%' and first_name like '%a';
    SQL> select first_name from employees where first_name like '__n%';
    SQL> select first_name from employees where job_id like '%REP%';
16. **Demonstrate Aggregate functions and AS command**
    SQL> select sum(salary) as SUM, avg(salary) as AVG, min(salary) as MIN, max(salary) as MAX, count(salary) as COUNT from employees;
17. **Demonstrate GROUP BY statement**
    SQL> select department_id, sum(salary) from employees group by department_id having sum(salary)>50000 order by sum(salary) desc;
18. **Demonstrate ROUND function**
    SQL> select round(45.93456,3) from dual;
    SQL> select round(-1.1,0) from dual;
19. **Demonstrate ALTER, SET, UPDATE and DATE datatype**
    SQL> alter table samp add(doj date);
    SQL> update samp set doj='22-mar-2020' where regno = 102;
20. **Demonstrate IN, NOT IN, IS NULL AND IS NOT NULL operator.**
    SQL> select first_name, department_id from employees where department_id in(10,20,30);
    SQL> select first_name, department_id from employees where department_id not in(10,20,30);

SQL> select employee_id, commission_pct from employees where commission_pct is null;
SQL> select employee_id, commission_pct from employees where commission_pct is not null;

Code:

```
SQL> @c:/Ora/hr_main;
******  Creating REGIONS table ....

Table created.


Index created.


Table altered.

******  Creating COUNTRIES table ....

Table created.


Table altered.

******  Creating LOCATIONS table ....

Table created.


Index created.


Table altered.


Sequence created.

******  Creating DEPARTMENTS table ....

Table created.


Index created.


Table altered.


Sequence created.
```

```
******  Creating JOBS table ....

Table created.


Index created.


Table altered.

******  Creating EMPLOYEES table ....

Table created.


Index created.


Table altered.


Table altered.


Sequence created.

******  Creating JOB_HISTORY table ....

Table created.


Index created.


Table altered.

******  Creating EMP_DETAILS_VIEW view ...

View created.


Commit complete.


Session altered.

******  Populating REGIONS table ....
```

```
1 row created.
.
.
.
.
.
1 row created.

******  Populating COUNTIRES table ....

1 row created.
.
.
.
.
.
1 row created.

******  Populating LOCATIONS table ....

1 row created.
.
.
.
.
.
1 row created.

******  Populating DEPARTMENTS table ....

Table altered.


1 row created.
.
.
.
.
.
1 row created.

******  Populating JOBS table ....

1 row created.
.
.
```

```
.
.
.
1 row created.

******  Populating EMPLOYEES table ....

1 row created.
.
.
.
.
.
1 row created.

******  Populating JOB_HISTORY table ....

1 row created.
.
.
.
.
.
1 row created.


Table altered.


Commit complete.


Index created.
.
.
.
.
.
Index created.


Commit complete.


Procedure created.
```

```
Trigger created.


Procedure created.


Trigger created.


Commit complete.


Comment created.
.
.
.
.
.
Comment created.


Commit complete.

SQL> select * from employees;

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
        100 Steven               King
SKING                     515.123.4567         17-JUN-87 AD_PRES         24000
                                    90
.
.
.
.
.
        206 William              Gietz
WGIETZ                    515.123.8181         07-JUN-94 AC_ACCOUNT       8300
                     205           110


107 rows selected.

SQL> select * from jobs;
```

```
JOB_ID     JOB_TITLE                           MIN_SALARY MAX_SALARY
---------- ---------------------------------- ---------- ----------
AD_PRES    President                               20000      40000
AD_VP      Administration Vice President           15000      30000
AD_ASST    Administration Assistant                 3000       6000
FI_MGR     Finance Manager                          8200      16000
FI_ACCOUNT Accountant                               4200       9000
AC_MGR     Accounting Manager                       8200      16000
AC_ACCOUNT Public Accountant                        4200       9000
SA_MAN     Sales Manager                           10000      20000
SA_REP     Sales Representative                     6000      12000
PU_MAN     Purchasing Manager                       8000      15000
PU_CLERK   Purchasing Clerk                         2500       5500
ST_MAN     Stock Manager                            5500       8500
ST_CLERK   Stock Clerk                              2000       5000
SH_CLERK   Shipping Clerk                           2500       5500
IT_PROG    Programmer                               4000      10000
MK_MAN     Marketing Manager                        9000      15000
MK_REP     Marketing Representative                 4000       9000
HR_REP     Human Resources Representative           4000       9000
PR_REP     Public Relations Representative          4500      10500

19 rows selected.

SQL> select * from jobs where job_title = 'President';

JOB_ID     JOB_TITLE                           MIN_SALARY MAX_SALARY
---------- ---------------------------------- ---------- ----------
AD_PRES    President                               20000      40000

1 row selected.

SQL> create table samp (regno number, name varchar(15));

Table created.

SQL> desc samp
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 REGNO                                              NUMBER
 NAME                                               VARCHAR2(15)

SQL> insert into samp values(&No,'&name');
Enter value for no: 101
Enter value for name: ashvath
```

```
1 row created.

SQL> /
Enter value for no: 102
Enter value for name: kevin

1 row created.

SQL> select * from samp;

     REGNO NAME
---------- ---------------
       101 ashvath
       102 kevin

2 rows selected.

SQL> desc employees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_ID                               NOT NULL NUMBER(6)
 FIRST_NAME                                         VARCHAR2(20)
 LAST_NAME                                 NOT NULL VARCHAR2(25)
 EMAIL                                     NOT NULL VARCHAR2(25)
 PHONE_NUMBER                                       VARCHAR2(20)
 HIRE_DATE                                 NOT NULL DATE
 JOB_ID                                    NOT NULL VARCHAR2(10)
 SALARY                                             NUMBER(8,2)
 COMMISSION_PCT                                     NUMBER(2,2)
 MANAGER_ID                                         NUMBER(6)
 DEPARTMENT_ID                                      NUMBER(4)

SQL> select first_name, employee_id from employees;

FIRST_NAME           EMPLOYEE_ID
-------------------- -----------
Steven                       100
Neena                        101
Lex                          102
.
.
.
.
.
Hermann                      204
Shelley                      205
William                      206
```

```
107 rows selected.

SQL> select first_name, employee_id, salary from employees;

FIRST_NAME           EMPLOYEE_ID     SALARY
-------------------- ----------- ----------
Steven                       100      24000
Neena                        101      17000
Lex                          102      17000
.
.
.
.
.
Hermann                      204      10000
Shelley                      205      12000
William                      206       8300

107 rows selected.

SQL> select &c1,&c2,&c3 from employees;
Enter value for c1: salary
Enter value for c2: last_name
Enter value for c3: job_id

    SALARY LAST_NAME                JOB_ID
---------- ------------------------- ----------
     24000 King                      AD_PRES
     17000 Kochhar                   AD_VP
     17000 De Haan                   AD_VP
      9000 Hunold                    IT_PROG
                           .
                           .
                           .
                           .
                           .
      6500 Mavris                    HR_REP
     10000 Baer                      PR_REP
     12000 Higgins                   AC_MGR
      8300 Gietz                     AC_ACCOUNT

107 rows selected.

SQL> select first_name, employee_id, salary from employees where salary>10000;

FIRST_NAME           EMPLOYEE_ID     SALARY
```

```
------------------- ----------- ----------
Steven                        100      24000
Neena                         101      17000
.
.
.
.
.
Michael                       201      13000
Shelley                       205      12000

15 rows selected.

SQL> select first_name, employee_id, salary from employees where
first_name='Michael';

FIRST_NAME           EMPLOYEE_ID     SALARY
-------------------- ----------- ----------
Michael                       134       2900
Michael                       201      13000

2 rows selected.

SQL> select first_name, lower(first_name) from employees;

FIRST_NAME           LOWER(FIRST_NAME)
-------------------- --------------------
Ellen                ellen
Sundar               sundar
Mozhe                mozhe
.
.
.
.
.
Matthew              matthew
Jennifer             jennifer
Eleni                eleni

107 rows selected.

SQL> select first_name,lower(first_name),upper(first_name),initcap(first_name) from
employees;

FIRST_NAME   LOWER(FIRST_NAME)     UPPER(FIRST_NAME)  INITCAP(FIRST_NAME)          -
---------- ------------------ ------------------ ---------------------
Ellen                ellen                ELLEN      Ellen
```

```
Sundar               sundar               SUNDAR     Sundar
.
.
.
.
.
Jennifer             jennifer             JENNIFER   Jennifer

Eleni                eleni                ELENI            Eleni


107 rows selected.

SQL> set linesize 1500;
SQL> select first_name,lower(first_name),upper(first_name),initcap(first_name) from
employees;

FIRST_NAME        LOWER(FIRST_NAME)   UPPER(FIRST_NAME)   INITCAP(FIRST_NAME)
----------------- ----------------- -------------------- --------------------
Ellen                ellen                ELLEN                Ellen
Sundar               sundar               SUNDAR               Sundar
Mozhe                mozhe                MOZHE                Mozhe
.
.
.
.
.
Matthew              matthew              MATTHEW              Matthew
Jennifer             jennifer             JENNIFER             Jennifer
Eleni                eleni                ELENI                Eleni

107 rows selected.

SQL> select employee_id,first_name from employees where lower(first_name)='michael';

EMPLOYEE_ID FIRST_NAME
----------- --------------------
        134 Michael
        201 Michael

2 rows selected.

SQL> select first_name, salary from employees order by salary asc;

FIRST_NAME               SALARY
-------------------- ----------
```

```
TJ                         2100
Steven                     2200
Hazel                      2200
.
.
.
.
.
Lex                       17000
Neena                     17000
Steven                    24000

107 rows selected.

SQL> select first_name, salary from employees order by salary desc;

FIRST_NAME               SALARY
-------------------- ----------
Steven                    24000
Neena                     17000
Lex                       17000
.
.
.
.
.
Hazel                      2200
Steven                     2200
TJ                         2100

107 rows selected.

SQL> select first_name, salary from employees order by first_name asc;

FIRST_NAME               SALARY
-------------------- ----------
Adam                       8200
Alana                      3100
Alberto                   12000
.
.
.
.
.
William                    8300
William                    7400
Winston                    3200
```

```
107 rows selected.

SQL> select first_name from employees where first_name like 'A%';

FIRST_NAME
--------------------
Amit
Alexis
.
.
.
.
.
Allan
Alana

10 rows selected.

SQL> select first_name from employees where first_name like 'A%' and first_name like
'%a';

FIRST_NAME
--------------------
Alyssa
Alana

2 rows selected.

SQL> select first_name from employees where first_name like 'A%' or first_name like
'%a';

FIRST_NAME
--------------------
Amit
Laura
Alexis
.
.
.
.
.
Clara
Shanta
Alana

24 rows selected.
```

```
SQL> select first_name from employees where first_name like '_a%';

FIRST_NAME
--------------------
David
Sarah
David
.
.
.
.
.
Martha
Patrick
Matthew

32 rows selected.

SQL> select first_name from employees where first_name like '__n%';

FIRST_NAME
--------------------
Sundar
Nanette
Jennifer
.
.
.
.
.
Jonathon
Winston
Jennifer

19 rows selected.

SQL> select first_name from employees where job_id like '%REP%';

FIRST_NAME
--------------------
Peter
David
Peter
.
.
.
```

```
.
.
Pat
Susan
Hermann

33 rows selected.

SQL> select sum(salary) from employees;

SUM(SALARY)
-----------
     691400

1 row selected.

SQL> select sum(salary),avg(salary),min(salary),max(salary) from employees;

SUM(SALARY) AVG(SALARY) MIN(SALARY) MAX(SALARY)
----------- ----------- ----------- -----------
     691400  6461.68224        2100       24000

1 row selected.

SQL> select sum(salary) as SUM, avg(salary) as AVG, min(salary) as MIN, max(salary)
as MAX, count(salary) as COUNT from employees;

       SUM        AVG        MIN        MAX      COUNT
---------- ---------- ---------- ---------- ----------
    691400 6461.68224       2100      24000        107

1 row selected.

SQL> select department_id, sum(salary) from employees group by department_id;

DEPARTMENT_ID SUM(SALARY)
------------- -----------
          100       51600
           30       24900
                     7000
           90       58000
           20       19000
           70       10000
          110       20300
           50      156400
           80      304500
           40        6500
```

```
           60       28800
           10        4400

12 rows selected.

SQL> select department_id, job_id, sum(salary) from employees' group by
department_id, job_id;

DEPARTMENT_ID JOB_ID     SUM(SALARY)
------------- ---------- -----------
          110 AC_ACCOUNT        8300
           90 AD_VP            34000
           50 ST_CLERK         55700
.
.
.
.
.
           10 AD_ASST           4400
           20 MK_REP            6000
           40 HR_REP            6500

20 rows selected.

SQL> select department_id, sum(salary) from employees group by department_id having
sum(salary)>50000;

DEPARTMENT_ID SUM(SALARY)
------------- -----------
          100       51600
           90       58000
           50      156400
           80      304500

4 rows selected.

SQL> select department_id, sum(salary) from employees group by department_id having
sum(salary)>50000 order by sum(salary) desc;

DEPARTMENT_ID SUM(SALARY)
------------- -----------
           80      304500
           50      156400
           90       58000
          100       51600

4 rows selected.
```

```
SQL> select round(45.93456,3) from dual;

ROUND(45.93456,3)
-----------------
           45.935

1 row selected.

SQL> select round(41,0) from dual;

ROUND(41,0)
-----------
         41

1 row selected.

SQL> select round(41.93456,0) from dual;

ROUND(41.93456,0)
-----------------
               42

1 row selected.

SQL> select round(-1.93456,0) from dual;

ROUND(-1.93456,0)
-----------------
               -2

1 row selected.

SQL> select round(-1.1,0) from dual;

ROUND(-1.1,0)
-------------
           -1

1 row selected.

SQL> alter table samp add(doj date);

Table altered.

SQL> set linesize 20;
SQL> desc samp;
```

```
 Name              Null?    Type
 ----------------- -------- ------------
 REGNO                      NUMBER
 NAME                       VARCHAR2(15)
 DOJ                        DATE

SQL> set linesize 1500;
SQL> select * from samp;

     REGNO NAME            DOJ
---------- --------------- ---------
       101 ashvath
       102 kevin

2 rows selected.

SQL> update samp set doj='01-jan-2021';

2 rows updated.

SQL> update samp set doj='22-mar-2020' where regno = 102;

1 row updated.

SQL> select * from samp;

     REGNO NAME            DOJ
---------- --------------- ---------
       101 ashvath         01-JAN-21
       102 kevin           22-MAR-20

2 rows selected.

SQL> set linesize 20;
SQL> desc employees;
 Name              Null?    Type
 ----------------- -------- ------------
 EMPLOYEE_ID       NOT NULL NUMBER(6)
 FIRST_NAME                 VARCHAR2(20)
 LAST_NAME         NOT NULL VARCHAR2(25)
 EMAIL             NOT NULL VARCHAR2(25)
 PHONE_NUMBER               VARCHAR2(20)
 HIRE_DATE         NOT NULL DATE
 JOB_ID            NOT NULL VARCHAR2(10)
 SALARY                     NUMBER(8,2)
 COMMISSION_PCT             NUMBER(2,2)
 MANAGER_ID                 NUMBER(6)
```

```
 DEPARTMENT_ID              NUMBER(4)

SQL> select sysdate from dual;

SYSDATE
---------
28-JUL-22

1 row selected.

SQL> set linesize 1500;
SQL> select employee_id, first_name, hire_date, sysdate-hire_date as exp from
employees;

EMPLOYEE_ID FIRST_NAME           HIRE_DATE        EXP
----------- -------------------- --------- ----------
        100 Steven               17-JUN-87 12825.4861
        101 Neena                21-SEP-89 11998.4861
        102 Lex                  13-JAN-93 10788.4861
                                 .
                                 .
                                 .
                                 .
                                 .
        204 Hermann              07-JUN-94 10278.4861
        205 Shelley              07-JUN-94 10278.4861
        206 William              07-JUN-94 10278.4861

107 rows selected.

SQL> select employee_id, first_name, hire_date, (sysdate-hire_date)/30 as exp from
employees;

EMPLOYEE_ID FIRST_NAME           HIRE_DATE        EXP
----------- -------------------- --------- ----------
        100 Steven               17-JUN-87 427.516265
        101 Neena                21-SEP-89 399.949598
        102 Lex                  13-JAN-93 359.616265
                                 .
                                 .
                                 .
                                 .
                                 .
        204 Hermann              07-JUN-94 342.616265
        205 Shelley              07-JUN-94 342.616265
        206 William              07-JUN-94 342.616265
```

```
107 rows selected.

SQL> select employee_id, first_name, hire_date, round(((sysdate-hire_date)/30),0) as
exp from employees;

EMPLOYEE_ID FIRST_NAME           HIRE_DATE        EXP
----------- -------------------- --------- ----------
        100 Steven               17-JUN-87        428
        101 Neena                21-SEP-89        400
        102 Lex                  13-JAN-93        360
                                 .
                                 .
                                 .
                                 .
                                 .
        204 Hermann              07-JUN-94        343
        205 Shelley              07-JUN-94        343
        206 William              07-JUN-94        343

107 rows selected.

SQL> select first_name, department_id from employees where department_id
in(10,20,30);

FIRST_NAME           DEPARTMENT_ID
-------------------- -------------
Jennifer                        10
Michael                         20
Pat                             20
Den                             30
Alexander                       30
Shelli                          30
Sigal                           30
Guy                             30
Karen                           30

9 rows selected.

SQL> select first_name, department_id from employees where department_id not
in(10,20,30);

FIRST_NAME           DEPARTMENT_ID
-------------------- -------------
Steven                          90
Neena                           90
Lex                             90
.
```

```
.
.
.
.
Hermann                         70
Shelley                        110
William                        110

97 rows selected.

SQL> select employee_id, commission_pct from employees;

EMPLOYEE_ID COMMISSION_PCT
----------- --------------
        100
        101
        102
             .
             .
             .
             .
             .

        145             .4
        146             .3
        147             .3
             .
             .
             .
             .
             .
        204
        205
        206

107 rows selected.

SQL> select employee_id, commission_pct from employees where commission_pct is null;

EMPLOYEE_ID COMMISSION_PCT
----------- --------------
        100
        101
        102
             .
             .
             .
```

```
             .
             .
        204
        205
        206

72 rows selected.

SQL> select employee_id, commission_pct from employees where commission_pct is not
null;

EMPLOYEE_ID COMMISSION_PCT
----------- --------------
        145             .4
        146             .3
        147             .3
        148             .3
.
.
.
.
.
        175            .25
        176             .2
        177             .2
        178            .15
        179             .1

35 rows selected.
```

Ex. No: 3 **Date: 11 – 08 - 2022**

Experiment 3

Consider the following schema for a Library Database:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)
BOOK_AUTHORS (Book_id, Author_Name)
PUBLISHER (Name, Address, Phone)
BOOK_COPIES (Book_id, Branch_id, No-of_Copies)
CARD (Card_No)
BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)
LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the library.

Creation of Tables

SQL> create table publisher (Name varchar2(20) primary key, Phone Integer, Address varchar2(20));

SQL> create table book (Bookid Integer primary key, Title varchar2(20), Pub_Year Integer, Publisher_name references Publisher (Name) on delete cascade);

SQL> create table book_authors (Author_name varchar (20), Book_id references book (Book_id) on delete cascade, primary key (Book_id, author_name));

SQL> create table library_branch (Branch_id integer primary key, Address varchar (20), Branch_name varchar (20));

SQL> create table book_copies (no_of_copies integer, Book_id references book(book_id) on delete cascade, Branch_id references library_branch (Branch_id) on delete cascade);

SQL> create table card (Card_no integer primary key);

SQL> create table book_lending (Date_Out date, Due_date date, Branch_id references library_branch (Branch_id) on delete cascade, Book_id references book(book_id) on delete cascade, Card_no references card(card_no) on delete cascade, primary key (Branch_id, Book_id, Card_no));

Insertion of Values to Tables

INSERT INTO PUBLISHER VALUES (‘MCGRAW-HILL’, 9989076587, ‘BANGALORE’);
INSERT INTO PUBLISHER VALUES (‘PEARSON’, 9889076565, ‘NEWDELHI’);
INSERT INTO PUBLISHER VALUES (‘RANDOM HOUSE’, 7455679345, ‘HYDRABAD’);
INSERT INTO PUBLISHER VALUES (‘HACHETTE LIVRE’, 8970862340, ‘CHENAI’);
INSERT INTO PUBLISHER VALUES (‘GRUPO PLANETA’, 7756120238, ‘BANGALORE’);
INSERT INTO BOOK VALUES (1,’DBMS’,’JAN-2017’, ‘MCGRAWHILL’);
INSERT INTO BOOK VALUES (2,’ADBMS’,’JUN-2016’, ‘MCGRAW-HILL’);
INSERT INTO BOOK VALUES (3,’CN’,’SEP-2016’, ‘PEARSON’);
INSERT INTO BOOK VALUES (4,’CG’,’SEP-2015’, ‘GRUPO PLANETA’);
INSERT INTO BOOK VALUES (5,’OS’,’MAY-2016’, ‘PEARSON’);
INSERT INTO BOOK_AUTHORS VALUES (’NAVATHE’, 1);
INSERT INTO BOOK_AUTHORS VALUES (’NAVATHE’, 2);
INSERT INTO BOOK_AUTHORS VALUES (’TANENBAUM’, 3);

INSERT INTO BOOK_AUTHORS VALUES (’EDWARD ANGEL’, 4);
INSERT INTO BOOK_AUTHORS VALUES (’GALVIN’, 5);
INSERT INTO LIBRARY_BRANCH VALUES (10,’RR NAGAR’,’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (11,’RNSIT’,’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (12,’RAJAJI NAGAR’, ’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (13,’NITTE’,’MANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (14,’MANIPAL’,’UDUPI’);
INSERT INTO BOOK_COPIES VALUES (10, 1, 10);
INSERT INTO BOOK_COPIES VALUES (5, 1, 11);
INSERT INTO BOOK_COPIES VALUES (2, 2, 12);
INSERT INTO BOOK_COPIES VALUES (5, 2, 13);
INSERT INTO BOOK_COPIES VALUES (7, 3, 14);
INSERT INTO BOOK_COPIES VALUES (1, 5, 10);
INSERT INTO BOOK_COPIES VALUES (3, 4, 11);
INSERT INTO CARD VALUES (100);
INSERT INTO CARD VALUES (101);
INSERT INTO CARD VALUES (102);
INSERT INTO CARD VALUES (103);
INSERT INTO CARD VALUES (104);
INSERT INTO BOOK_LENDING VALUES (’01-JAN-17’,’01-JUN-17’, 1, 10, 101);
INSERT INTO BOOK_LENDING VALUES (’11-JAN-17’,’11-MAR-17’, 3, 14, 101);
INSERT INTO BOOK_LENDING VALUES (’21-FEB-17’,’21-APR-17’, 2, 13, 101);
INSERT INTO BOOK_LENDING VALUES (’15-MAR-17’,’15-JUL-17’, 4, 11, 101);
INSERT INTO BOOK_LENDING VALUES (‘12-APR-17’,’12-MAY-17’, 1, 11, 104);

Queries:

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.

SQL> Select A.Book_id, A.Title, A.pub_name, B.author_name, C.no_of_copies, D.Branch_id from Book A, Book_Authors B, Book_copies C, Library_Branch D where A.book_id = B.Book_id and A.book_id=C.Book_id and C.Branch_id=D.Branch_id;

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017

SQL> Select card_no from book_lending where date_out between '01-JAN-2017' and '01-JUN-2020' group by card_no having count(*)>3;

3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

SQL> Delete from book where book_id = 4;

4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

SQL> create view v_publication as select pub_year from book;

5. Create a view of all books and its number of copies that are currently available in the library.

SQL> create view bcop as select A.Title, A.Book_id, B.no_of_copies, C.Branch_id from Book A, Book_copies B, Library_Branch C where A.Book_id = B.Book_id and B.Branch_id = C.Branch_id;

Code:

```
SQL> create table publisher(Name varchar2(20) primary key, Phone Integer, Address
varchar2(20));
```

```
Table created.

SQL> create table book(Bookid Integer primary key, Title varchar2(20), Pub_Year
Integer, Publisher_name references Publisher(Name) on delete cascade);

Table created.

SQL> create table book_authors(Author_name varchar(20), Book_id references
book(Book_id) on delete cascade, primary key(Book_id,author_name));

Table created.

SQL> create table library_branch(Branch_id integer primary key, Address varchar(20),
Branch_name varchar(20));

Table created.

SQL> create table book_copies(no_of_copies integer, Book_id references book(book_id)
on delete cascade, Branch_id references library_branch(Branch_id) on delete cascade);

Table created.

SQL> create table card(Card_no integer primary key);

Table created.

SQL> create table book_lending(Date_Out date, Due_date date, Branch_id references
library_branch(Branch_id) on delete cascade, Book_id references book(book_id) on
delete cascade, Card_no references card(card_no) on delete cascade, primary
key(Branch_id, Book_id, Card_no));

Table created.

SQL> INSERT INTO PUBLISHER VALUES ('MCGRAW-HILL', 9989076587,'BANGALORE');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('PEARSON', 9889076565, 'NEWDELHI');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('RANDOM HOUSE', 7455679345,'HYDRABAD');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('HACHETTE LIVRE', 8970862340, 'CHENAI');

1 row created.
```

```
SQL> INSERT INTO PUBLISHER VALUES ('GRUPO PLANETA', 7756120238, 'BANGALORE');

1 row created.

SQL> INSERT INTO BOOK VALUES (1,'DBMS',2017, 'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (2,'ADBMS',2016,'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (3,'CN',2016,'PEARSON');

1 row created.

SQL> INSERT INTO BOOK VALUES (4,'CG',2015, 'GRUPO PLANETA');

1 row created.

SQL> INSERT INTO BOOK VALUES (5,'OS',2016,'PEARSON');

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES('NAVATHE', 1);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('NAVATHE', 2);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('TANENBAUM', 3);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('EDWARDANGEL', 4);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES('GALVIN', 5);

1 row created.

SQL> INSERT INTO LIBRARY_BRANCH VALUES (10,'BANGALORE','RR NAGAR');
```

```
1 row created.

SQL> INSERT INTO LIBRARY_BRANCH VALUES (11,'BANGALORE','RNSIT');

1 row created.

SQL> INSERT INTO LIBRARY_BRANCH VALUES (12, 'BANGALORE','RAJAJI NAGAR');

1 row created.

SQL> INSERT INTO LIBRARY_BRANCH VALUES (13,'MANGALORE','NITTE');

1 row created.

SQL> INSERT INTO LIBRARY_BRANCH VALUES (14,'UDUPI','MANIPAL');

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (10, 1, 10);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (5, 1, 11);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (2, 2, 12);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (5, 2, 13);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (7, 3, 14);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (1, 5, 10);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (3, 4, 11);

1 row created.

SQL> INSERT INTO CARD VALUES (100);
```

```
1 row created.

SQL> INSERT INTO CARD VALUES (101);

1 row created.

SQL> INSERT INTO CARD VALUES (102);

1 row created.

SQL> INSERT INTO CARD VALUES (103);

1 row created.

SQL> INSERT INTO CARD VALUES (104);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('01-JAN-17','01-JUN-17', 10, 1, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('11-JAN-17','11-MAR-17', 14, 3, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('21-FEB-17','21-APR-17', 13, 2, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('15-MAR-17','15-JUL-17', 11, 4, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('12-APR-17','12-MAY-17', 11, 1, 104);

1 row created.
```

```
SQL> Select * from publisher;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL          9989076587 BANGALORE
PEARSON              9889076565 NEWDELHI
RANDOM HOUSE         7455679345 HYDRABAD
HACHETTE LIVRE       8970862340 CHENAI
GRUPO PLANETA        7756120238 BANGALORE

SQL> Select * from book;

   BOOK_ID TITLE                   PUB_YEAR PUB_NAME
---------- -------------------- ---------- --------------------
         1 DBMS                       2017 MCGRAW-HILL
         2 ADBMS                      2016 MCGRAW-HILL
         3 CN                         2016 PEARSON
         5 OS                         2016 PEARSON
         4 CG                         2015 GRUPO PLANETA

SQL> Select * from book_authors;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                      1
NAVATHE                      2
TANENBAUM                    3
EDWARDANGEL                  4
GALVIN                       5

SQL> Select * from library_branch;

 BRANCH_ID ADDRESS              BRANCH_NAME
---------- -------------------- --------------------
        10 BANGALORE            RR NAGAR
        11 BANGALORE            RNSIT
        12 BANGALORE            RAJAJI NAGAR
        13 MANGALORE            NITTE
        14 UDUPI                MANIPAL

SQL> Select * from book_copies;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
```

```
           5          2         13
           7          3         14
           1          5         10
           3          4         11

7 rows selected.

SQL> Select * from card;

   CARD_NO
----------
       100
       101
       102
       103
       104

SQL> Select * from book_lending;

DATE_OUT  DUE_DATE   BRANCH_ID    BOOK_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17         10          1        101
11-JAN-17 11-MAR-17         14          3        101
21-FEB-17 21-APR-17         13          2        101
15-MAR-17 15-JUL-17         11          4        101
12-APR-17 12-MAY-17         11          1        104

SQL> set linesize 1500;
SQL> Select A.Book_id, A.Title, A.pub_name, B.author_name, C.no_of_copies,
D.Branch_id from Book A, Book_Authors B, Book_copies C, Library_Branch D where
A.book_id = B.Book_id and A.book_id=C.Book_id and C.Branch_id=D.Branch_id;

   BOOK_ID TITLE                PUB_NAME             AUTHOR_NAME
NO_OF_COPIES  BRANCH_ID
---------- -------------------- -------------------- -------------------- -----------
- ----------
         1 DBMS                 MCGRAW-HILL          NAVATHE
10         10
         1 DBMS                 MCGRAW-HILL          NAVATHE
5         11
         2 ADBMS                MCGRAW-HILL          NAVATHE
2         12
         2 ADBMS                MCGRAW-HILL          NAVATHE
5         13
         3 CN                   PEARSON              TANENBAUM
7         14
         5 OS                   PEARSON              GALVIN
1         10
```

```
         4 CG                    GRUPO PLANETA        EDWARDANGEL
3         11

7 rows selected.

SQL> Select card_no from book_lending where date_out between '01-JAN-2017' and '01-
JUN-2020' group by card_no having count(*)>3;
   CARD_NO
----------
       101
SQL> delete from book where book_id = 4;

1 row deleted.

SQL> SELECT * FROM PUBLISHER;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL          9989076587 BANGALORE
PEARSON              9889076565 NEWDELHI
RANDOM HOUSE         7455679345 HYDRABAD
HACHETTE LIVRE       8970862340 CHENAI
GRUPO PLANETA        7756120238 BANGALORE

SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                   PUB_YEAR PUB_NAME
---------- -------------------- ---------- --------------------
         1 DBMS                       2017 MCGRAW-HILL
         2 ADBMS                      2016 MCGRAW-HILL
         3 CN                         2016 PEARSON
         5 OS                         2016 PEARSON
SQL> SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                      1
NAVATHE                      2
TANENBAUM                    3
GALVIN                       5
SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID ADDRESS              BRANCH_NAME
---------- -------------------- --------------------
        10 BANGALORE            RR NAGAR
        11 BANGALORE            RNSIT
        12 BANGALORE            RAJAJI NAGAR
```

```
        13 MANGALORE            NITTE
        14 UDUPI                MANIPAL
SQL> SELECT * FROM BOOK_COPIES;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
           5          2         13
           7          3         14
           1          5         10

6 rows selected.

SQL> SELECT * FROM CARD;

   CARD_NO
----------
       100
       101
       102
       103
       104
SQL> SELECT * FROM BOOK_LENDING;

DATE_OUT  DUE_DATE   BRANCH_ID    BOOK_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17         10          1        101
11-JAN-17 11-MAR-17         14          3        101
21-FEB-17 21-APR-17         13          2        101
12-APR-17 12-MAY-17         11          1        104

SQL> create view V_Publications as select pub_year from book;

View created.

SQL> select * from V_publications;

  PUB_YEAR
----------
      2017
      2016
      2016
      2016
```

```
SQL> create view bcop as select A.Title, A.Book_id, B.no_of_copies, C.Branch_id from
Book A, Book_copies B, Library_Branch C where A.Book_id = B.Book_id and B.Branch_id =
C.Branch_id;

View created.

SQL> Select * from bcop;

TITLE                    BOOK_ID NO_OF_COPIES  BRANCH_ID
-------------------- ---------- ------------ ----------
DBMS                          1           10         10
DBMS                          1            5         11
ADBMS                         2            2         12
ADBMS                         2            5         13
CN                            3            7         14
OS                            5            1         10

6 rows selected.
```

Ex. No: 4 **Date: 18 – 08 - 2022**

Experiment 4

Consider the following schema for Order Database:

SALESMAN (Salesman_id, Name, City, Commission)

CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)

ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

Write SQL queries to

1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesmen who had more than one customer.
3. List all salesmen and indicate those who have and do not have customers in their cities.
(Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order
of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his
orders must also be deleted.

Creation of Tables

SQL> CREATE TABLE Salesman (Salesman_id number(4), Name varchar2(20), City varchar2(20), Commission varchar2(20), primary key(Salesman_id));

SQL> CREATE TABLE Customer1 (Customer_id number(4), Cust_Name varchar2(20), City varchar2(20), Grade number(3), primary key(Customer_id), Salesman_id references Salesman(Salesman_id) on delete set null);

SQL> CREATE TABLE Orders (Ord_No number(5), Purchase_Amt number(10,2), Ord_Date Date, Customer_id references Customer1(Customer_id) on delete cascade , Salesman_id references Salesman(Salesman_id) on delete cascade);

Insertion of Values to Tables

SQL> INSERT INTO Salesman VALUES (1000, 'JOHN', 'BANGALORE', '25%');

SQL> INSERT INTO Salesman VALUES (2000, 'RAVI', 'BANGALORE', '20%');

SQL> INSERT INTO Salesman VALUES (3000, 'KUMAR', 'MYSORE', '15%');

SQL> INSERT INTO Salesman VALUES (4000, 'SMITH', 'DELHI', '30%');

SQL> INSERT INTO Salesman VALUES (5000, 'HARSHA', 'HYDRABAD', '15%');

SQL> INSERT INTO Customer1 VALUES (10, 'PREETHI', 'BANGALORE', 100, 1000);

SQL> INSERT INTO Customer1 VALUES (11, 'VIVEK', 'MANGALORE', 300, 1000);

SQL> INSERT INTO Customer1 VALUES (12, 'BHASKAR', 'CHENNAI', 400, 2000);

SQL> INSERT INTO Customer1 VALUES (13, 'CHETHAN', 'BANGALORE', 200, 2000);

SQL> INSERT INTO Customer1 VALUES (14, 'MAMATHA', 'BANGALORE', 400, 3000);

SQL> INSERT INTO Orders VALUES (50, 5000, '04-MAY-2017', 10, 1000);

SQL> INSERT INTO Orders VALUES (51, 450, '20-JAN-2017', 10, 2000);

SQL> INSERT INTO Orders VALUES (52, 1000, '24-FEB-2017', 13, 2000);

SQL> INSERT INTO Orders VALUES (53, 3500, '13-APR-2017', 14, 3000);

SQL> INSERT INTO Orders VALUES (54, 550, '09-MAR-2017', 12, 2000);

Queries:

1. Count the customers with grades above Bangalore's average.

SQL> SELECT Grade, COUNT (*) FROM Customer1 GROUP BY Grade HAVING Grade > (SELECT AVG(Grade) FROM Customer1 WHERE City = 'BANGALORE');

2. Find the name and numbers of all salesmen who had more than one customer.

SQL> SELECT Salesman_id, Name FROM Salesman S WHERE 1 < (SELECT COUNT(*) FROM Customer WHERE Salesman_id=S.Salesman_id);

3. List all salesmen and indicate those who have and do not have customers in their cities.
(Use UNION operation.)

SQL> SELECT S.Salesman_id, Name, Cust_name, Commission FROM Salesman S, Customer1 C WHERE S.City = C.City UNION (SELECT Salesman_id, Name, 'NO MATCH', Commission FROM Salesman WHERE NOT City = ANY (SELECT City FROM Customer1)) ORDER BY 2 DESC;

4. Create a view that finds the salesman who has the customer with the highest order of a day.

SQL> CREATE VIEW bestsalesman AS SELECT B.Ord_date, S.Salesman_id, S.Name FROM Salesman S, Orders B WHERE S.Salesman_id = B.Salesman_id AND B. Purchase_Amt = (SELECT MAX (Purchase_Amt) FROM Orders O WHERE O.Ord_date = B.Ord_date);

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

SQL> DELETE FROM Salesman WHERE Salesman_id=1000;

Code:

```
SQL> CREATE TABLE Salesman (Salesman_id number(4), Name varchar2(20), City
varchar2(20), Commission varchar2(20), primary key(Salesman_id));

Table created.

SQL> CREATE TABLE Customer1 (Customer_id number(4), Cust_Name varchar2(20), City
varchar2(20), Grade number(3), primary key(Customer_id), Salesman_id references
Salesman(Salesman_id) on delete set null);

Table created.

SQL> CREATE TABLE Orders (Ord_No number(5), Purchase_Amt number(10,2), Ord_Date Date,
Customer_id references Customer1(Customer_id) on delete cascade , Salesman_id
references Salesman(Salesman_id) on delete cascade);

Table created.

SQL> desc salesman;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALESMAN_ID                               NOT NULL NUMBER(4)
 NAME                                               VARCHAR2(20)
 CITY                                               VARCHAR2(20)
 COMMISSION                                         VARCHAR2(20)

SQL> desc customer1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(4)
 CUST_NAME                                          VARCHAR2(20)
 CITY                                               VARCHAR2(20)
 GRADE                                              NUMBER(3)
 SALESMAN_ID                                        NUMBER(4)

SQL> desc orders;
 Name                                      Null?    Type
```

```
 ---------------------------------------- -------- ----------------------------
 ORD_NO                                            NUMBER(5)
 PURCHASE_AMT                                      NUMBER(10,2)
 ORD_DATE                                          DATE
 CUSTOMER_ID                                       NUMBER(4)
 SALESMAN_ID                                       NUMBER(4)

SQL> INSERT INTO Salesman VALUES (1000, 'JOHN', 'BANGALORE', '25%');

1 row created.

SQL> INSERT INTO Salesman VALUES (2000, 'RAVI', 'BANGALORE', '20%');

1 row created.

SQL> INSERT INTO Salesman VALUES (3000, 'KUMAR', 'MYSORE', '15%');

1 row created.

SQL> INSERT INTO Salesman VALUES (4000, 'SMITH', 'DELHI', '30%');

1 row created.

SQL> INSERT INTO Salesman VALUES (5000, 'HARSHA', 'HYDRABAD', '15%');

1 row created.

SQL> select * from salesman;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- --------------------
       1000 JOHN                 BANGALORE            25%
       2000 RAVI                 BANGALORE            20%
       3000 KUMAR                MYSORE               15%
       4000 SMITH                DELHI                30%
       5000 HARSHA               HYDRABAD             15%

SQL> INSERT INTO Customer1 VALUES (10, 'PREETHI', 'BANGALORE', 100, 1000);

1 row created.

SQL> INSERT INTO Customer1 VALUES (11, 'VIVEK', 'MANGALORE', 300, 1000);

1 row created.

SQL> INSERT INTO Customer1 VALUES (12, 'BHASKAR', 'CHENNAI', 400, 2000);
```

```
1 row created.

SQL> INSERT INTO Customer1 VALUES (13, 'CHETHAN', 'BANGALORE', 200, 2000);

1 row created.

SQL> INSERT INTO Customer1 VALUES (14, 'MAMATHA', 'BANGALORE', 400, 3000);

1 row created.

SQL> select * from customer1;

CUSTOMER_ID CUST_NAME            CITY                      GRADE SALESMAN_ID
----------- -------------------- -------------------- ---------- -----------
         10 PREETHI              BANGALORE                   100        1000
         11 VIVEK                MANGALORE                   300        1000
         12 BHASKAR              CHENNAI                     400        2000
         13 CHETHAN              BANGALORE                   200        2000
         14 MAMATHA              BANGALORE                   400        3000

SQL> INSERT INTO Orders VALUES (50, 5000, '04-MAY-2017', 10, 1000);

1 row created.

SQL> INSERT INTO Orders VALUES (51, 450, '20-JAN-2017', 10, 2000);

1 row created.

SQL> INSERT INTO Orders VALUES (52, 1000, '24-FEB-2017', 13, 2000);

1 row created.

SQL> INSERT INTO Orders VALUES (53, 3500, '13-APR-2017', 14, 3000);

1 row created.

SQL> INSERT INTO Orders VALUES (54, 550, '09-MAR-2017', 12, 2000);

1 row created.

SQL> select * from orders;

    ORD_NO PURCHASE_AMT ORD_DATE  CUSTOMER_ID SALESMAN_ID
---------- ------------ --------- ----------- -----------
        50         5000 04-MAY-17          10        1000
        51          450 20-JAN-17          10        2000
        52         1000 24-FEB-17          13        2000
```

```
        53         3500 13-APR-17          14        3000
        54          550 09-MAR-17          12        2000

SQL> SELECT Grade, COUNT (*)
  2  FROM Customer1
  3  GROUP BY Grade
  4  HAVING Grade >
  5           (SELECT AVG(Grade)
  6            FROM Customer1
  7            WHERE City = 'BANGALORE');

     GRADE   COUNT(*)
---------- ----------
       400          2
       300          1

SQL> SELECT Salesman_id, Name
  2  FROM Salesman S
  3  WHERE 1 <
  4      (SELECT COUNT(*)
  5       FROM Customer
  6       WHERE Salesman_id=S.Salesman_id);

SALESMAN_ID NAME
----------- --------------------
       1000 JOHN
       2000 RAVI
       3000 KUMAR
       4000 SMITH
       5000 HARSHA

SQL> SELECT S.Salesman_id, Name, Cust_name, Commission
  2  FROM Salesman S, Customer1 C
  3  WHERE S.City = C.City
  4  UNION
  5  (SELECT Salesman_id, Name, 'NO MATCH', Commission
  6   FROM Salesman
  7   WHERE NOT City = ANY
  8   (SELECT City
  9    FROM Customer1))
 10  ORDER BY 2 DESC;

SALESMAN_ID NAME                 CUST_NAME            COMMISSION
----------- -------------------- -------------------- --------------------
       4000 SMITH                NO MATCH             30%
       2000 RAVI                 CHETHAN              20%
       2000 RAVI                 MAMATHA              20%
```

```
       2000 RAVI                 PREETHI              20%
       3000 KUMAR                NO MATCH             15%
       1000 JOHN                 CHETHAN              25%
       1000 JOHN                 MAMATHA              25%
       1000 JOHN                 PREETHI              25%
       5000 HARSHA               NO MATCH             15%

9 rows selected.

SQL> CREATE VIEW bestsalesman
  2  AS SELECT B.Ord_date, S.Salesman_id, S.Name
  3  FROM Salesman S, Orders B
  4  WHERE S.Salesman_id = B.Salesman_id
  5  AND B. Purchase_Amt=
  6           (SELECT MAX (Purchase_Amt)
  7            FROM Orders O
  8            WHERE O.Ord_date = B.Ord_date);

View created.

SQL> SELECT * FROM bestsalesman;

ORD_DATE  SALESMAN_ID NAME
--------- ----------- --------------------
04-MAY-17        1000 JOHN
20-JAN-17        2000 RAVI
24-FEB-17        2000 RAVI
13-APR-17        3000 KUMAR
09-MAR-17        2000 RAVI

SQL> DELETE FROM Salesman
  2  WHERE Salesman_id=1000;

1 row deleted.

SQL> select * from salesman;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- --------------------
       2000 RAVI                 BANGALORE            20%
       3000 KUMAR                MYSORE               15%
       4000 SMITH                DELHI                30%
       5000 HARSHA               HYDRABAD             15%

SQL> select * from customer1;

CUSTOMER_ID CUST_NAME            CITY                      GRADE SALESMAN_ID
```

```
---------- -------------------- -------------------- ---------- ----------
        10 PREETHI              BANGALORE                   100
        11 VIVEK                MANGALORE                   300
        12 BHASKAR              CHENNAI                     400       2000
        13 CHETHAN              BANGALORE                   200       2000
        14 MAMATHA              BANGALORE                   400       3000

SQL> select * from orders;

    ORD_NO PURCHASE_AMT ORD_DATE  CUSTOMER_ID SALESMAN_ID
---------- ------------ --------- ----------- -----------
        51          450 20-JAN-17          10        2000
        52         1000 24-FEB-17          13        2000
        53         3500 13-APR-17          14        3000
        54          550 09-MAR-17          12        2000
```

Ex. No: 5 **Date: 25 – 08 - 2022**

Experiment 5

Consider the following schema of a database:

books (id, title, type, author_id, editor_id, translator_id)
authors (id, first_name, last_name)
editors (id, first_name, last_name)
translators (id, first_name, last_name)

Tables:

books

id	title	Type	author_id	editor_id	translator_id
1	Time to Grow Up!	Original	11	21	
2	Your Trip	translated	15	22	32
3	Lovely Love	Original	14	24	
4	Dream Your Life	Original	11	24	
5	Oranges	translated	12	25	31
6	Your Happy Life	translated	15	22	33
7	Applied AI	translated	13	23	34
8	My Last Book	Original	11	27	

authors

id	first_name	last_name
11	Ellen	Writer
12	Olga	Savelieva
13	Jack	Smart
14	Donald	Brain
15	Yao	Dou

editors

id	first_name	last_name
21	Daniel	Brown
22	Mark	Johnson
23	Maria	Evans
24	Cathrine	Roberts
25	Sebastian	Wright
26	Barbara	Jones
27	Matthew	Smith

translators

id	first_name	last_name
31	Ira	Davies
32	Ling	Weng
33	Kristian	Green
34	Roman	Edwards

Write SQL queries to

- **Using INNER JOIN:**
  - Show book titles along with their authors (i.e., the author's first name and last name).
  - Display books along with their translators (i.e., the translator's last name). Only half of our books have been translated and thus have a corresponding translator.
- **Using LEFT JOIN:**

- Display information about each book's author and translator (i.e., their last names). We also want to keep the basic information about each book (i.e., id, title, and type).
- Show the basic book information (i.e., ID and title) along with the last names of the corresponding editors. Again, we want to keep all the books in the result set.

- **Using RIGHT JOIN:**
  - Let us repeat our previous example, but this time, our task will be to keep all the records from the editor's table.
- **Using FULL OUTER JOIN:**
  - Let us again join the books and editor's tables, but this time, we will be keeping all records from both tables.
  - Join all four tables to get information about all the books, authors, editors, and translators in one table.

Queries:

- **Using INNER JOIN:**
  - Show book titles along with their authors (i.e., the author's first name and last name).

```
SQL>  SELECT B.id, B.title, A.first_name, A.last_name
        FROM books B
        INNER JOIN authors A
        ON B.author_id = A.id
        ORDER BY B.id;
```

  - Display books along with their translators (i.e., the translator's last name). Only half of our books have been translated and thus have a corresponding translator.

```
SQL>  SELECT B.id, B.title, T.last_name AS Translator
        FROM books B
        INNER JOIN translators T
        ON B.translator_id = T.id
        ORDER BY b.id;
```

- **Using LEFT JOIN:**
  - Display information about each book's author and translator (i.e., their last names). We also want to keep the basic information about each book (i.e., id, title, and type).

```
SQL>  SELECT B.id, B.title, B.type, A.last_name AS Author,
        T.last_name AS Translator
        FROM books B
        LEFT JOIN authors A
        ON B.author_id = A.id
        LEFT JOIN translators T
        ON B.translator_id = T.id
        ORDER BY B.id;
```

  - Show the basic book information (i.e., ID and title) along with the last names of the corresponding editors. Again, we want to keep all the books in the result set.

```
SQL>  SELECT B.id, B.title, E.last_name AS Editor
        FROM books B
        LEFT JOIN editors E
        ON B.editor_id = E.id
        ORDER BY B.id;
```

- **Using RIGHT JOIN:**
  - Let us repeat our previous example, but this time, our task will be to keep all the records from the editor's table.

```
SQL>  SELECT B.id, b.title, E.last_name AS Editor
```

FROM books B
RIGHT JOIN editors E
ON B.editor_id = E.id
ORDER BY B.id;

- **Using FULL OUTER JOIN:**
  - Let us again join the books and editor's tables, but this time, we will be keeping all records from both tables.

    SQL> SELECT B.id, B.title, E.last_name AS Editor
    FROM books B
    FULL JOIN editors E
    ON B.editor_id = E.id
    ORDER BY B.id;
  - Join all four tables to get information about all the books, authors, editors, and translators in one table.

    SQL> SELECT B.id, B.title, A.last_name AS Author, E.last_name AS Editor,
    T.last_name AS Translator
    FROM books B
    FULL JOIN authors A
    ON B.author_id = A.id
    FULL JOIN editors E
    ON B.editor_id = E.id
    FULL JOIN translators T
    ON B.translator_id = T.id
    ORDER BY B.id;

Code:

```
SQL> create table authors(id number(2),
  2  first_name varchar2(20),
  3  last_name varchar2(20),
  4  primary key(id));

Table created.

SQL> desc authors;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 ID                                        NOT NULL NUMBER(2)
 FIRST_NAME                                         VARCHAR2(20)
 LAST_NAME                                          VARCHAR2(20)

SQL> create table editors(id number(2),
  2  first_name varchar2(20),
  3  last_name varchar2(20),
  4  primary key(id));

Table created.

SQL> desc editors;
 Name                                      Null?    Type
```

```
 ---------------------------------------- -------- ----------------------
 ID                                       NOT NULL NUMBER(2)
 FIRST_NAME                                        VARCHAR2(20)
 LAST_NAME                                         VARCHAR2(20)

SQL> create table translators(id number(2),
  2  first_name varchar2(20),
  3  last_name varchar2(20),
  4  primary key(id));

Table created.

SQL> desc translators;
 Name                                     Null?    Type
 ---------------------------------------- -------- ----------------------
 ID                                       NOT NULL NUMBER(2)
 FIRST_NAME                                        VARCHAR2(20)
 LAST_NAME                                         VARCHAR2(20)

SQL> create table books1(id number(1),
  2  title varchar2(40),
  3  type varchar2(20),
  4  author_id number(2) references authors(id) on delete set null,
  5  editor_id number(2) references editors(id) on delete set null,
  6  translator_id number(2) references translators(id) on delete set null,
  7  primary key(id));

Table created.

SQL> desc books1;
 Name                                     Null?    Type
 ---------------------------------------- -------- ----------------------
 ID                                       NOT NULL NUMBER(1)
 TITLE                                             VARCHAR2(40)
 TYPE                                              VARCHAR2(20)
 AUTHOR_ID                                         NUMBER(2)
 EDITOR_ID                                         NUMBER(2)
 TRANSLATOR_ID                                     NUMBER(2)

SQL> insert into authors values(11,'Ellen','Writer');

1 row created.

SQL> insert into authors values(12,'Olga','Savelieva');

1 row created.
```

```
SQL> insert into authors values(13,'Jack','Smart');

1 row created.

SQL> insert into authors values(14,'Donald','Brain');

1 row created.

SQL> insert into authors values(15,'Yao','Dou');

1 row created.

SQL> insert into editors values(21,'Daniel','Brown');

1 row created.

SQL> insert into editors values(22,'Mark','Johnson');

1 row created.

SQL> insert into editors values(23,'Maria','Evans');

1 row created.

SQL> insert into editors values(24,'Cathrine','Roberts');

1 row created.

SQL> insert into editors values(25,'Sebastine','Wright');

1 row created.

SQL> insert into editors values(26,'Barbara','Jones');

1 row created.

SQL> insert into editors values(27,'Matthew','Smith');

1 row created.

SQL> insert into translators values(31,'Ira','Davies');

1 row created.

SQL> insert into translators values(32,'Ling','Weng');

1 row created.
```

```
SQL> insert into translators values(33,'Kristian','Green');

1 row created.

SQL> insert into translators values(34,'Roman','Edwards');

1 row created.

SQL> insert into books1 values(1,'Time to Grow up!','original',11,21,NULL);

1 row created.

SQL> insert into books1 values(2,'Your Trip','translated',15,22,32);

1 row created.

SQL> insert into books1 values(3,'Lovely Love','original',14,24,NULL);

1 row created.

SQL> insert into books1 values(4,'Dream Your Life','original',11,24,NULL);

1 row created.

SQL> insert into books1 values(5,'Oranges','translated',12,25,31);

1 row created.

SQL> insert into books1 values(6,'Your Happy Life','translated',15,22,33);

1 row created.

SQL> insert into books1 values(7,'Applied AI','translated',13,23,34);

1 row created.

SQL> insert into books1 values(8,'My Last Book','original',11,27,NULL);

1 row created.

SQL> set linesize 1500;
SQL> select * from books1;

        ID TITLE                                          TYPE                    AUTHOR_ID
EDITOR_ID TRANSLATOR_ID
```

```
---------- ---------------------------------------- ------------------- ---------- -
--------- -------------
         1 Time to Grow up!                         original                   11
21
         2 Your Trip                                translated                 15
22            32
         3 Lovely Love                              original                   14
24
         4 Dream Your Life                          original                   11
24
         5 Oranges                                  translated                 12
25            31
         6 Your Happy Life                          translated                 15
22            33
         7 Applied AI                               translated                 13
23            34
         8 My Last Book                             original                   11
27

8 rows selected.

SQL> select * from authors;

        ID FIRST_NAME           LAST_NAME
---------- -------------------- --------------------
        11 Ellen                Writer
        12 Olga                 Savelieva
        13 Jack                 Smart
        14 Donald               Brain
        15 Yao                  Dou

SQL> select * from editors;

        ID FIRST_NAME           LAST_NAME
---------- -------------------- --------------------
        21 Daniel               Brown
        22 Mark                 Johnson
        23 Maria                Evans
        24 Cathrine             Roberts
        25 Sebastine            Wright
        26 Barbara              Jones
        27 Matthew              Smith

7 rows selected.

SQL> select * from translators;

        ID FIRST_NAME           LAST_NAME
```

```
---------- -------------------- --------------------
        31 Ira                  Davies
        32 Ling                 Weng
        33 Kristian             Green
        34 Roman                Edwards
SQL> select B.id, B.title, A.first_name, A.last_name
  2  from books1 B
  3  inner join authors A
  4  on B.author_id = A.id
  5  order by B.id;

        ID TITLE                                  FIRST_NAME           LAST_NAME
---------- ---------------------------------------- -------------------- ------------
--------
         1 Time to Grow up!                       Ellen                Writer
         2 Your Trip                              Yao                  Dou
         3 Lovely Love                            Donald               Brain
         4 Dream Your Life                        Ellen                Writer
         5 Oranges                                Olga                 Savelieva
         6 Your Happy Life                        Yao                  Dou
         7 Applied AI                             Jack                 Smart
         8 My Last Book                           Ellen                Writer

8 rows selected.

SQL> select B.id, B.title, T.last_name AS Translator
  2  from books1 B
  3  inner join translators T
  4  on B.translator_id = T.id
  5  order by b.id;

        ID TITLE                                  TRANSLATOR
---------- ---------------------------------------- --------------------
         2 Your Trip                              Weng
         5 Oranges                                Davies
         6 Your Happy Life                        Green
         7 Applied AI                             Edwards

SQL> select B.id, B.title, B.type, A.last_name AS Author,
  2  T.last_name AS Translator
  3  from books1 B
  4  left join authors A
  5  on B.author_id = A.id
  6  left join translators T
  7  on B.translator_id = T.id
  8  order by B.id;
```

```
        ID TITLE                                    TYPE                AUTHOR
TRANSLATOR
---------- ---------------------------------------- ------------------- ------------
-------- --------------------
         1 Time to Grow up!                         original            Writer
         2 Your Trip                                translated          Dou
Weng
         3 Lovely Love                              original            Brain
         4 Dream Your Life                          original            Writer
         5 Oranges                                  translated          Savelieva
Davies
         6 Your Happy Life                          translated          Dou
Green
         7 Applied AI                               translated          Smart
Edwards
         8 My Last Book                             original            Writer

8 rows selected.

SQL> select B.id, B.title, E.last_name AS Editor
  2  from books1 B
  3  left join editors E
  4  on B.editor_id = E.id
  5  order by B.id;

        ID TITLE                                    EDITOR
---------- ---------------------------------------- --------------------
         1 Time to Grow up!                         Brown
         2 Your Trip                                Johnson
         3 Lovely Love                              Roberts
         4 Dream Your Life                          Roberts
         5 Oranges                                  Wright
         6 Your Happy Life                          Johnson
         7 Applied AI                               Evans
         8 My Last Book                             Smith

8 rows selected.

SQL> select B.id, b.title, E.last_name AS Editor
  2  from books1 B
  3  right join editors E
  4  on B.editor_id = E.id
  5  order by B.id;

        ID TITLE                                    EDITOR
---------- ---------------------------------------- --------------------
         1 Time to Grow up!                         Brown
         2 Your Trip                                Johnson
```

```
         3 Lovely Love                             Roberts
         4 Dream Your Life                         Roberts
         5 Oranges                                 Wright
         6 Your Happy Life                         Johnson
         7 Applied AI                              Evans
         8 My Last Book                            Smith
                                                   Jones

9 rows selected.

SQL> select B.id, B.title, E.last_name AS Editor
  2  from books1 B
  3  full join editors E
  4  on B.editor_id = E.id
  5  order by B.id;

        ID TITLE                                    EDITOR
---------- ---------------------------------------- --------------------
         1 Time to Grow up!                         Brown
         2 Your Trip                                Johnson
         3 Lovely Love                              Roberts
         4 Dream Your Life                          Roberts
         5 Oranges                                  Wright
         6 Your Happy Life                          Johnson
         7 Applied AI                               Evans
         8 My Last Book                             Smith
                                                    Jones
9 rows selected.

SQL> select  B.id, B.title, A.last_name AS Author, E.last_name AS Editor,
  2  T.last_name AS Translator
  3  from  books1 B
  4  full join authors A
  5  on B.author_id = A.id
  6  full join editors E
  7  on B.editor_id = E.id
  8  full join translators T
  9  on B.translator_id = T.id
 10  order by B.id;

        ID TITLE                                    AUTHOR               EDITOR
TRANSLATOR
---------- ---------------------------------------- -------------------- ------------
-------- --------------------
         1 Time to Grow up!                         Writer               Brown
         2 Your Trip                                Dou                  Johnson
Weng
```

```
         3 Lovely Love                          Brain               Roberts
         4 Dream Your Life                      Writer              Roberts
         5 Oranges                              Savelieva           Wright
Davies
         6 Your Happy Life                      Dou                 Johnson
Green
         7 Applied AI                           Smart               Evans
Edwards
         8 My Last Book                         Writer              Smith

                                                                    Jones

9 rows selected.
```

Ex. No: 6 **Date: 01 – 09 - 2022**

Experiment 6

Consider the following schema of a database:

Client (ID, Name, E_ID)

Country (ID, Country, Country_Code)

Tables:

Client

ID	Name	Email_ID
1	George	ge.com
2	David	da.com
3	Chris	ch.com
4	Morrison	mo.com
5	Brian	cor.com

Country

ID	Country	Country_Code
1	India	IND
2	Spain	ESP
3	France	FRA
4	England	ENG
5	Poland	POL

Write SQL queries to

1) Create a view from a single table.
   Create a simple view for client table.
2) Create a view from multiple tables.
   Create a view that will display Client_ID, Name, Country, and Country_Code columns data for all five clients.
3) Inserting a new roll in a view.
4) Updating a row in a view.
5) Deleting a row in a view.
6) Drop a view in SQL.

Queries:

1) Create a view from a single table.
Create a simple view for client table.

```
SQL> CREATE VIEW Client_Email_IDs AS
  2  SELECT Email_ID
  3  FROM Client;
```

2) Create a view from multiple tables.
Create a view that will display Client_ID, Name, Country, and Country_Code columns data for all five clients.

```
SQL> CREATE VIEW Client_Details AS
  2  SELECT Cl.ID, Cl.Name, Co.Country, Co.Country_Code
  3  FROM Country Co, Client Cl
  4  WHERE Cl.ID = Co.ID;
```

3) Inserting a new roll in a view.

```
SQL> INSERT INTO Client_Email_IDs
  2  VALUES(6,'jo.com');
```

4) Updating a row in a view.

```
SQL> CREATE OR REPLACE VIEW Client_Details AS
```

```
  2  SELECT Cl.ID, Cl.Name, Cl.Email_ID, Co.Country, Co.Country_Code,
  4  FROM Country Co, Client Cl
  5  WHERE Cl.ID = Co.ID;
```

5) Deleting a row in a view.

```
SQL> DELETE FROM Client_Email_IDs
  2  WHERE ID = 6;
```

6) Drop a view in SQL.

```
SQL> DROP VIEW Client_Email_IDs;
```

Code:

```
SQL> create table Client(ID number(1), Name varchar2(20), Email_ID varchar2(20),
primary key(ID));

Table created.

SQL> desc client
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(1)
 NAME                                               VARCHAR2(20)
 EMAIL_ID                                           VARCHAR2(20)

SQL> create table Country(ID number(1), Country varchar2(20), Country_Code
varchar2(20), primary key(ID));

Table created.

SQL> desc country
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(1)
 COUNTRY                                            VARCHAR2(20)
 COUNTRY_CODE                                       VARCHAR2(20)

SQL> insert into client values(1,'George','ge.com');

1 row created.

SQL> insert into client values(2,'David','da.com');

1 row created.

SQL> insert into client values(3,'Chris','ch.com');

1 row created.

SQL> insert into client values(4,'Morrison','mo.com');

1 row created.
```

```
SQL> insert into client values(5,'Brian','cor.com');

1 row created.

SQL> insert into Country values(1,'India','IND');

1 row created.

SQL> insert into Country values(2,'Spain','ESP');

1 row created.

SQL> insert into Country values(3,'France','FRA');

1 row created.

SQL> insert into Country values(4,'Poland','POL');

1 row created.

SQL> DELETE FROM country WHERE Country = 'Poland';

1 row deleted.

SQL> insert into Country values(4,'England','ENG');

1 row created.

SQL> insert into Country values(5,'Poland','POL');

1 row created.

SQL> select * from Country;

        ID COUNTRY              COUNTRY_CODE
---------- -------------------- --------------------
         1 India                IND
         2 Spain                ESP
         3 France               FRA
         4 England              ENG
         5 Poland               POL

SQL> select * from Client;

        ID NAME                 EMAIL_ID
---------- -------------------- --------------------
```

```
         1 George               ge.com
         2 David                da.com
         3 Chris                ch.com
         4 Morrison             mo.com
         5 Brian                cor.com

SQL> CREATE VIEW Client_Email_IDs AS
  2  SELECT Email_ID
  3  FROM Client;

View created.

SQL> select * from Client_Email_IDs;

EMAIL_ID
--------------------
ge.com
da.com
ch.com
mo.com
cor.com

SQL> CREATE VIEW Client_Details AS
  2  SELECT Cl.ID, Cl.Name, Co.Country, Co.Country_Code
  3  FROM Country Co, Client Cl
  4  WHERE Cl.ID = Co.ID;

View created.

SQL> select * from Client_Details;

        ID NAME                 COUNTRY              COUNTRY_CODE
---------- -------------------- -------------------- --------------------
         1 George               India                IND
         2 David                Spain                ESP
         3 Chris                France               FRA
         4 Morrison             England              ENG
         5 Brian                Poland               POL

SQL> CREATE OR REPLACE VIEW Client_Details AS
  2  SELECT Cl.ID, Cl.Name, Cl.Email_ID, Co.Country, Co.Country_Code
  3  FROM Country Co, Client Cl
  4  WHERE Cl.ID = Co.ID;

View created.

SQL> set linesize 1500;
```

```
SQL> select * from Client_Details;

        ID NAME                 EMAIL_ID            COUNTRY
COUNTRY_CODE
---------- -------------------- -------------------- -------------------- -----------
---------
         1 George               ge.com               India                IND
         2 David                da.com               Spain                ESP
         3 Chris                ch.com               France               FRA
         4 Morrison             mo.com               England              ENG
         5 Brian                cor.com              Poland               POL
SQL> drop view client_email_ids;

View dropped.

SQL> CREATE VIEW Client_Email_IDs AS select id, email_id from client;

View created.

SQL> select * from client_email_ids;

        ID EMAIL_ID
---------- --------------------
         1 ge.com
         2 da.com
         3 ch.com
         4 mo.com
         5 cor.com
SQL> insert into client_email_ids values (6,'jo.com');

1 row created.

SQL> select * from client_email_ids;

        ID EMAIL_ID
---------- --------------------
         1 ge.com
         2 da.com
         3 ch.com
         4 mo.com
         5 cor.com
         6 jo.com

6 rows selected.

SQL> select * from client;
```

```
        ID NAME                 EMAIL_ID
---------- -------------------- --------------------
         1 George               ge.com
         2 David                da.com
         3 Chris                ch.com
         4 Morrison             mo.com
         5 Brian                cor.com
         6                      jo.com

6 rows selected.

SQL> DELETE FROM Client_email_ids where id=6;

1 row deleted.

SQL> select * from client_email_ids;

        ID EMAIL_ID
---------- --------------------
         1 ge.com
         2 da.com
         3 ch.com
         4 mo.com
         5 cor.com

SQL> select * from client;

        ID NAME                 EMAIL_ID
---------- -------------------- --------------------
         1 George               ge.com
         2 David                da.com
         3 Chris                ch.com
         4 Morrison             mo.com
         5 Brian                cor.com
```