AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

LABORATORY MANUAL

DATABASE MANAGEMENT SYSTEM

SUBJECT CODE: 3130703

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

B.E. 3rd SEMESTER

NAME:_____
ENROLLMENTNO:_____
BATCHNO:_____
YEAR: _____

Amiraj College of Engineering and Technology,
Nr.Tata Nano Plant, Khoraj, Sanand, Ahmedabad.

AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

Amiraj College of Engineering and Technology,

Nr.Tata Nano Plant, Khoraj, Sanand, Ahmedabad.

CERTIFICATE

This is to certify that Mr. / Ms. _____

Of class_____ Enrolment No _____has

Satisfactorily completed the course in _____as

by the Gujarat Technological University for ____ Year (B.E.) semester___of

Computer Science & Engineering in the Academic year_____.

Date of Submission:-

Faculty Name and Signature
(Prof. Rupali Patel)

Head of Department
(CSE)

AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

B.E. 3rd SEMESTER

SUBJECT: DATABASE MANAGEMENT SYSTEM

SUBJECT CODE: 3130703

List Of Experiments

Sr. No.	Title	Date of submission	Sign	Remark
1	To study DDL-create and DML-insert commands.			
2	Create the below given table and insert the data accordingly.			
3	To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables			
4	To To study Single-row functions.			
5	Displaying data from Multiple Tables (join).			
6	To apply the concept of Aggregating Data using Group functions.			
7	To solve queries using the concept of sub query			
8	To Manipulate Data using various SQL commands.			
9	To apply the concept of security and privileges.			

10	To study Transaction control commands.			

PRACTICAL -1

AIM: **To study DDL-create and DML-insert commands.**

Create tables according to the following definition.
CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5) ,CNAME VARCHAR2(18) , BNAME VARCHAR2(18) , AMOUNT NUMBER(8,2) ,ADATE DATE);

CREATE TABLE BRANCH(BNAME VARCHAR2(18),CITY VARCHAR2(18));

CREATE TABLECUSTOMERS(CNAME VARCHAR2(19) ,CITY VARCHAR2(18));
CREATE TABLE BORROW(LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));

Insert the data as shown below.

DEPOSIT

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000.00	1-MAR-95
101	SUNIL	AJNI	5000.00	4-JAN-96
102	MEHUL	KAROLBAGH	3500.00	17-NOV-95
104	MADHURI	CHANDI	1200.00	17-DEC-95
105	PRMOD	M.G.ROAD	3000.00	27-MAR-96
106	SANDIP	ANDHERI	2000.00	31-MAR-96
107	SHIVANI	VIRAR	1000.00	5-SEP-95
108	KRANTI	NEHRU PLACE	5000.00	2-JUL-95
109	MINU	POWAI	7000.00	10-AUG-95

BRANCH

VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

CUSTOMERS

ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

BORROW

LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

DEPOSIT:

INSERT INTO DEPOSIT VALUES('100',’ANIL’,'VRCE',1000,'1-MAR-95');
INSERT INTO DEPOSIT VALUES('101','SUNIL','AJNI',5000,'4-JAN-96');
INSERT INTO DEPOSIT VALUES('102','MEHUL','KAROLBAGH',3500,'17-NOV-95');
INSERT INTO DEPOSIT VALUES('104','MADHURI','CHANDI',1200,'17-DEC-95');
INSERT INTO DEPOSITVALUES('105','PRMOD','M.G.ROAD',3000,'27-MAR-96');
INSERT INTO DEPOSIT VALUES('106','SANDIP','ANDHERI',2000,'31-MAR-96');
INSERT INTO DEPOSIT VALUES('107','SHIVANI','VIRAR',1000,'5-SEP-95');
INSERT INTO DEPOSIT VALUES('108','KRANTI','NEHRUPLACE',5000,'2-JUL-95');
INSERT INTO DEPOSIT VALUES('109','MINU','POWAI',7000,'10-AUG-95');

BRANCH:

INSERT INTO BRANCH VALUES('VRCE','NAGPUR');
INSERT INTO BRANCH VALUES('AJNI','NAGPUR');
INSERT INTO BRANCH VALUES('KAROLBAGH','DELHI');
INSERT INTO BRANCH VALUES('CHANDI','DELHI');
INSERT INTO BRANCH VALUES('DHARAMPETH','NAGPUR');
INSERT INTO BRANCH VALUES('M.G.ROAD','BANGLORE');
INSERT INTO BRANCH VALUES('ANDHERI','BOMBAY');
INSERT INTO BRANCH VALUES('VIHAR','BOMBAY');

INSERT INTO BRANCH VALUES('NEHRU PLACE','DELHI');
INSERT INTO BRANCH VALUES('POWAI','BOMBAY');

CUSTOMER:

INSERT INTO CUSTOMERS VALUES ('ANIL','CALCUTTA');
INSERT INTO CUSTOMERS VALUES ('SUNIL','DELHI');
INSERT INTO CUSTOMERS VALUES ('MEHUL','BARODA');
INSERT INTO CUSTOMERS VALUES ('MANDAR','PATNA');
INSERT INTO CUSTOMERS VALUES ('MADHURI','NAGPUR');
INSERT INTO CUSTOMERS VALUES ('PRAMOD','NAGPUR');
INSERT INTO CUSTOMERS VALUES ('SANDIP','SURAT');
INSERT INTO CUSTOMERS VALUES ('SHIVANI','BOMBAY');
INSERT INTO CUSTOMERS VALUES ('KRANTI','BOMBAY');
INSERT INTO CUSTOMERS VALUES ('NAREN','BOMBAY');

BORROW:

INSERT INTO BORROW VALUES ('201','ANIL','VRCE',1000);
INSERT INTO BORROW VALUES ('206','MEHUL','VRCE',5000);
INSERT INTO BORROW VALUES ('311','SUNIL','DHARAMPETH',3000);
INSERT INTO BORROW VALUES ('321','MADHURI','ANDHERI',2000);
INSERT INTO BORROW VALUES ('375','PRMOD','VIHAR',8000);
INSERT INTO BORROW VALUES ('481','KRANTI','NEHRU PLACE',3000);

From the above given tables perform the following queries:

1. **Describe deposit, branch.**

DESC DEPOSIT;

```
SQL> desc deposite;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(20)
 C_NAME                                             VARCHAR2(20)
 B_NAME                                             VARCHAR2(20)
 AMMOUNT                                            NUMBER(20)
 ADATE                                              DATE
```

DESC BRANCH;

```
SQL> desc branch;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BNAME                                              VARCHAR2(10)
 CITY                                               VARCHAR2(10)
```

2. Describe borrow, customers.

DESC BORROW;

```
SQL> DESC BORROW;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOAN_NO                                            NUMBER(5)
 CNAME                                              VARCHAR2(18)
 BNAME                                              VARCHAR2(18)
 AMOUNT                                             NUMBER(8,2)
```

DESC CUSTOMERS;

```
SQL> DESC CUSTOMERS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CNAME                                              VARCHAR2(20)
 CITY                                               VARCHAR2(20)
```

3. List all data from table DEPOSIT.

SELECT * FROM DEPOSIT;

```
SQL>  SELECT * FROM DEPOSITE;

    ACC_NO C_NAME               B_NAME                  AMMOUNT ADATE
---------- -------------------- -------------------- ---------- ---------
       100 aman                 vrce                       1000 01-MAR-17
       101 sunil                ajni                       5000 04-JAN-17
       102 mehul                karolbagh                  3500 17-NOV-17
       103 madhuri              chandi                     1200 17-DEC-17
       104 prmod                mg road                    3000 27-MAR-17
       105 sandip               andheri                    2000 31-MAR-17
       106 shivani              virar                      1000 05-SEP-17
       107 kranit               nehru place                5000 02-JUL-17
       108 minu                 powai                      7000 10-AUG-17

9 rows selected.
```

4. List all data from table BORROW.

SELECT * FROM BORROW;

```
SQL> SELECT * FROM BORROW_42;

LOANN C_NAME               B_NAME                  AMOUNT
----- -------------------- -------------------- ----------
201   anil                 vrce                       1000
206   mehul                ajni                       5000
311   sunil                dharmampeth                3000
321   madhuri              andheri                    2000
375   pramod               virar                      8000
481   kranti               nehru                      3000

6 rows selected.
```

5. List all data from table CUSTOMERS.

SELECT * FROM CUSTOMERS;

```
SQL> SELECT * FROM CUSTOMERS;

CNAME                CITY
-------------------- --------------------
aman                 calcutta
sunil                delhi
mehul                baroda
mandar               patna
madhuri              nagpur
prmod                nagpur
sandip               surat
shivani              bombay
kranit               bombay
naren                bombay

10 rows selected.
```

6. List all data from table BRANCH.

SELECT * FROM BRANCH;

```
SQL> SELECT * FROM BRANCH;

BNAME      CITY
---------- ----------
vrce       nagpur
ajni       nagpur
karolbagh  delhi
chandi     delhi
andheri    banglore
virar      bombay
nehruplace bombay
powai      delhi
Vrce       Nagpur
Mni        Nagpur
Mehul      baroda

BNAME      CITY
---------- ----------
mandar     patna
madhuri    nagpur
sandeep    surat
shivani    mumbai
kranti     mumbai
Narem      bombay

17 rows selected.
```

7. Give account no and amount of depositors.

SELECT ACT NO,AMOUNT FROM DEPOSIT;

```
SQL> SELECT ACC_NO,AMMOUNT FROM DEPOSITE;

    ACC_NO    AMMOUNT
---------- ----------
       100       1000
       101       5000
       102       3500
       103       1200
       104       3000
       105       2000
       106       1000
       107       5000
       108       7000

9 rows selected.
```

8. Give name of depositors having amount greater than 4000.

SELECT CNAME FROM DEPOSITE WHERE AMOUNT >4000;

```
SQL>
SQL> SELECT C_NAME FROM DEPOSITE WHERE AMMOUNT >4000;

C_NAME
--------------------
sunil
kranit
minu
```

9. Give name of customers who opened account after date '1-12-96'.

SELECT C_NAME FROM DEPOSITE WHERE DATE > '1-DEC-96';

```
SQL>  SELECT C_NAME FROM DEPOSITE WHERE ADATE > '1-DEC-96';

C_NAME
--------------------
aman
sunil
mehul
madhuri
prmod
sandip
shivani
kranit
minu

9 rows selected.
```

PRACTICAL -2

AIM: Create the below given table and insert the data accordingly.

Create Table Job (job_id, job_title, min_sal, max_sal);

COLUMN NAME	DATA TYPE
job_id	Varchar2(15)
job_title	Varchar2(30)
min_sal	Number(7,2)
max_sal	Number(7,2)

Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no);

COLUMN NAME	DATA TYPE
emp_no	Number(3)
emp_name	Varchar2(30)
emp_sal	Number(8,2)
emp_comm	Number(6,1)
dept_no	Number(3)

Create table deposit(a_no,cname,bname,amount,a_date);

COLUMN NAME	DATA TYPE
a_no	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Number(7,2)
a_date	Date

Create table borrow(loanno,cname,bname,amount);

COLUMN NAME	DATA TYPE
loanno	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Varchar2(7,2)

Insert following values in the table **Employee**.

emp_n	emp_name	emp_sal	emp_comm	dept _no
101	Smith	800		20
102	Snehal	1600	300	25
103	Adama	1100	0	20
104	Aman	3000		15
105	Anita	5000	50,000	10
106	Sneha	2450	24,500	10
107	Anamika	2975		30

Insert following values in the table **job**.

job_id	job_name	min_sal	max_sal
IT_PROG	Programmer	4000	10000
MK_MGR	Marketing manager	9000	15000
FI_MGR	Finance manager	8200	12000
FI_ACC	Account	4200	9000
LEC	Lecturer	6000	17000
COMP_OP	Computer Operator	1500	3000

Insert following values in the table **deposit**.

A_no	cname	Bname	Amount	date
101	Anil	andheri	7000	01-jan-06
102	sunil	virar	5000	15-jul-06
103	jay	villeparle	6500	12-mar-06
104	vijay	andheri	8000	17-sep-06
105	keyur	dadar	7500	19-nov-06
106	mayur	borivali	5500	21-dec-06

Perform following queries

1. **Retrieve all data from employee, jobs and deposit.**

SELECT * FROM EMPLOYEE;

```
SQL> SELECT * FROM EMPLOYEE;

    EMP_NO EMP_NAME               EMP_SAL   EMP_COMM    DEPT_NO
---------- -------------------- ---------- ---------- ----------
         8 Aman                        500        600          5
         9 janak                      5200        800        110
        10 priyank                    5200        800        110
         1 smith                       500        400          4
         2 snehal                      600        500          4
         3 adama                       700        600          6
         4 aman                        800        700          7
         5 anita                       900        800          8
         6 sneha                      1000        900          9
         7 anamika                    1100       1000         10
         7 xyz                        1100       1000         11

    EMP_NO EMP_NAME               EMP_SAL   EMP_COMM    DEPT_NO
---------- -------------------- ---------- ---------- ----------
         7 xyz                        1200       1100         12

12 rows selected.
```

SELECT * FROM JOB;

```
SQL> SELECT * FROM JOB;

JOB_ID               JOB_TITLE             MIN_SAL    MAX_SAL
-------------------- -------------------- ---------- ----------
pro_gamer            gamer                      8000      20000
IT                   Programmer                 4000      10000
MK                   Marketing Manager          9000      15000
FIN                  Finance Manager            8200      12000
FI                   Accountant                 4200       9000
LEC                  Lecturer                   6000      17000
COMP                 Comp Operator              1500       3000

7 rows selected.
```

SELECT * FROM DEPOSIT;

```
SQL> SELECT * FROM DEPOSITE;

    ACC_NO C_NAME               B_NAME                 AMMOUNT ADATE
---------- -------------------- -------------------- ---------- ---------
       100 aman                 vrce                       1000 01-MAR-17
       101 sunil                ajni                       5000 04-JAN-17
       102 mehul                karolbagh                  3500 17-NOV-17
       103 madhuri              chandi                     1200 17-DEC-17
       104 prmod                mg road                    3000 27-MAR-17
       105 sandip               andheri                    2000 31-MAR-17
       106 shivani              virar                      1000 05-SEP-17
       107 kranit               nehru place                5000 02-JUL-17
       108 minu                 powai                      7000 10-AUG-17

9 rows selected.
```

2. **Give details of account no. and deposited rupees of customershaving account opened between dates 01-01-06 and 25-07-06.**

   SELECT ACC_NO, AMMOUNT FROM DEPOSITE
   WHERE ADATE BETWEEN '1-JAN-06' AND '25-
   JUL-06';

```
SQL> SELECT ACC_NO,AMMOUNT FROM DEPOSITE WHERE ADATE BETWEEN '01-MAR-17' AND '27-MAR-17'

    ACC_NO    AMMOUNT
---------- ----------
       100       1000
       104       3000
```

3. **Display all jobs with minimum salary is greater than 4000.**

   SELECT * FROM JOB WHERE MIN_SAL > 4000.

```
SQL> SELECT * FROM JOB WHERE MIN_SAL > 4000;

JOB_ID               JOB_TITLE             MIN_SAL    MAX_SAL
-------------------- -------------------- ---------- ----------
pro_gamer            gamer                      8000      20000
MK                   Marketing Manager          9000      15000
FIN                  Finance Manager            8200      12000
FI                   Accountant                 4200       9000
LEC                  Lecturer                   6000      17000
```

4. **Display name and salary of employee whose department no is 20. Give alias name to name of employee.**

   SELECT EMP_NAME AS "NAME",EMP_SAL FROM EMPLOYEE WHERE DEPT_NO=20;

```
SQL> SELECT EMP_NAME AS "NAME",EMP_SAL FROM EMPLOYEE WHERE DEPT_NO=10;

NAME                    EMP_SAL
-------------------- ----------
anamika                   1100
```

5. **Display employee no,name and department details of thoseemployee whose department lies in(10,20)**

   SELECT EMP_NO,EMP_NAME,DEPT_NO FROM
   EMPLOYEE WHERE DEPT_NO IN (10,20);

```
SQL> SELECT EMP_NO,EMP_NAME,DEPT_NO FROM EMPLOYEE WHERE DEPT_NO IN (10,20);

    EMP_NO EMP_NAME               DEPT_NO
---------- -------------------- ----------
         7 anamika                      10
```

6. **Display all employee whose name start with 'A' and third character is ''a'.**

   SELECT * FROM EMPLOYEE WHERE NAME LIKE 'A_a%';

```
SQL> SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE 'A_a%';

    EMP_NO EMP_NAME                EMP_SAL   EMP_COMM    DEPT_NO
---------- -------------------- ---------- ---------- ----------
         8 Aman                        500        600          5
```

7. **Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.**

   SELECT EMP_NAME,EMP_NO,EMP_SAL FROM EMPLOYEE WHERE EMP_NAME LIKE 'Ani___';

```
SQL> SELECT EMP_NAME,EMP_NO,EMP_SAL FROM EMPLOYEE WHERE EMP_NAME LIKE 'ani__';

EMP_NAME                EMP_NO    EMP_SAL
-------------------- ---------- ----------
anita                        5        900
```

8. **Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long.**

   SELECT * FROM EMPLOYEE WHERE EMP_COMM IS NOT NULL AND EMP_NAME LIKE '_n____';

```
SQL> SELECT * FROM EMPLOYEE WHERE EMP_COMM IS NOT NULL AND EMP_NAME LIKE '_n___';

    EMP_NO EMP_NAME                EMP_SAL   EMP_COMM    DEPT_NO
---------- -------------------- ---------- ---------- ----------
         5 anita                        900        800          8
         6 sneha                       1000        900          9
```

9. **Display the null values of employee and also employee name's third character should be 'a'.**

   SELECT * FROM EMPLOYEE WHERE
   EMP_COMM IS NULLAND EMP_NAME LIKE
   '_____a%';

```
SQL> SELECT * FROM EMPLOYEE WHERE EMP_COMM IS NULL AND EMP_NAME LIKE '__a%';

no rows selected
```

10. What will be output if you are giving LIKE predicate as '%_%' ESCAPE'\'

SELECT * FROM JOB WHERE JOB_ID LIKE '%_%' ESCAPE '\';

```
SQL> SELECT * FROM JOB WHERE JOB_ID LIKE '%\_%' ESCAPE '\';

JOB_ID             JOB_TITLE            MIN_SAL    MAX_SAL
------------------ ------------------ ---------- ----------
pro_gamer          gamer                    8000      20000
```

PRACTICAL -3

AIM: To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

1. **List total deposit from deposit.**

   SELECT SUM (AMOUNT) FROM DEPOSIT;

```
SQL> SELECT SUM (AMMOUNT) FROM DEPOSITE;

SUM(AMMOUNT)
------------
       28700
```

2. **List total loan from karolbagh branch.**

   SELECT SUM (AMOUNT) FROM BORROW WHERE BNAME='KAROLBAGH';

```
SQL> SELECT SUM (AMOUNT) FROM BORROW_42  WHERE B_NAME=' dharmampeth';

SUM(AMOUNT)
-----------
```

3. **Give maximum loan from branch vrce.**

   SELECT MAX (AMOUNT) FROM BORROW_42 WHERE B_NAME='VRCE';

```
SQL> SELECT MAX (AMOUNT) FROM BORROW_42 WHERE B_NAME='VRCE';

MAX(AMOUNT)
-----------
      10000
```

4. **Count total number of customers.**

   SELECT COUNT (CNAME) FROM CUSTOMERS;

```
SQL> SELECT COUNT (CNAME) FROM CUSTOMERS;

COUNT(CNAME)
------------
          10
```

5. **Count total number of customer's cities.**

   SELECT COUNT (DISTINCT CITY) FROM CUSTOMERS;

6. **Create table supplier from employee with all the columns.**

CREATE TABLE SUPPLIER AS SELECT * FROM EMPLOYEE;

```
SQL>  CREATE TABLE SUPPLIER_42 AS SELECT * FROM EMPLOYEE;

Table created.
```

7. **Create table sup1 from employee with first two columns.**

   CREATE TABLE SUP1 AS SELECT EMP_NO,EMP_NAME FROM EMPLOYEE;

```
SQL> CREATE TABLE SUP1 AS SELECT EMP_NO,EMP_NAME FROM EMPLOYEE;

Table created.
```

8. **Create table sup2 from employee with no data.**

   CREATE TABLE SUPPLIER AS SELECT * FROM EMPLOYEE WHERE EMP_NO=NULL;

```
SQL>  CREATE TABLE SUPPLIER_43 AS SELECT * FROM EMPLOYEE WHERE EMP_NO=NULL;

Table created.
```

9. **Insert the data into sup2 from employee whose second charactershould be 'n' and string should be 5 characters long in employee name field.**

   INSERT INTO SUP2 (ACTNO) SELECT (EMP_NO) FROM EMPLOYEE WHERE EMP_NAME LIKE '_a____';

```
SQL>  INSERT INTO SUP1 ( EMP_NO) SELECT (EMP_NO) FROM EMPLOYEE WHERE EMP_NAME LIKE '_n__

0 rows created.
```

10. **Delete all the rows from sup1.**

    TRUNCATE TABLE SUP1;

```
SQL> TRUNCATE TABLE SUP1;

Table truncated.
```

11. **Delete the detail of supplier whose sup_no is 103.**

    DELETE FROM SUPPLIER WHERE SUP_NO=103;

```
SQL> DELETE FROM SUPPLIER_42  WHERE EMP_NO=3;

1 row deleted.
```

12. Rename the table sup2.

RENAME SUP2 TO SUP3;

```
SQL> RENAME SUP2 TO SUP34;

Table renamed.
```

13. Destroy table sup1 with all the data.

DROP TABLE SUP1;

```
SQL> DROP TABLE SUP1;

Table dropped.
```

14. Update the value dept_no to 10 where second character of emp. nameis 'm'.

UPDATE EMPLOYEE SET DEPT_NO=10 WHERE EMP_NAME LIKE '_m%';

```
SQL> UPDATE EMPLOYEE SET DEPT_NO=10 WHERE EMP_NAME LIKE '_m%';

3 rows updated.
```

15. Update the value of employee name whose employee number is 103.

UPDATE EMPLOYEE SET EMP_NAME='DARSHAN' WHERE EMP_NO=103;

```
SQL> UPDATE EMPLOYEE SET EMP_NAME='DARSHAN' WHERE EMP_NO=103;

0 rows updated.
```

PRACTICAL -4

AIM: To To study Single-row functions.

1. **Write a query to display the current date. Label the column Date.**

   SELECT SYSDATE AS "DATE" FROM DUAL;

Results | Explain | Describe | Saved SQL | History

DATE
30-AUG-18

1 rows returned in 0.05 seconds CSV Export

2. **For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary**

   SELECT EMP_NO,EMP_NAME,EMP_SAL,EMP_SAL+(EMP_SAL*15/100) "New Salary" FROM EMPLOYEE;

```
SQL> SELECT EMP_NO,EMP_NAME,EMP_SAL,EMP_SAL+(EMP_SAL*15/100) "New Salary" FROM EMPLOYEE;

    EMP_NO EMP_NAME               EMP_SAL New Salary
---------- -------------------- ---------- ----------
         8 Aman                        500        575
         9 janak                      5200       5980
        10 priyank                    5200       5980
         1 smith                       500        575
         2 snehal                      600        690
         3 adama                       700        805
         4 aman                        800        920
         5 anita                       900       1035
         6 sneha                      1000       1150
         7 anamika                    1100       1265
         7 xyz                        1100       1265

    EMP_NO EMP_NAME               EMP_SAL New Salary
---------- -------------------- ---------- ----------
         7 xyz                        1200       1380

12 rows selected.
```

3. **Modify your query no 4.(2) to add a column that subtracts the oldsalary from the new salary. Label the column Increase.**

   SELECT EMP_NO,EMP_NAME,EMP_SAL,EMP_SAL+(EMP_SAL*15/100) "New Salary",(EMP_SAL+(EMP_SAL*15/100))- EMP_SAL "INCREASE" FROM EMPLOYEE;

```
SQL> SELECT EMP_NO,EMP_NAME,EMP_SAL,EMP_SAL+(EMP_SAL*15/100) "New Salary",(EMP_SAL+(EMP_SAL*15/100))
- EMP_SAL "INCREASE" FROM EMPLOYEE;

    EMP_NO EMP_NAME                EMP_SAL New Salary   INCREASE
---------- -------------------- ---------- ---------- ----------
         8 Aman                        500        575         75
         9 janak                      5200       5980        780
        10 priyank                    5200       5980        780
         1 smith                       500        575         75
         2 snehal                      600        690         90
         3 adama                       700        805        105
         4 aman                        800        920        120
         5 anita                       900       1035        135
         6 sneha                      1000       1150        150
         7 anamika                    1100       1265        165
         7 xyz                        1100       1265        165

    EMP_NO EMP_NAME                EMP_SAL New Salary   INCREASE
---------- -------------------- ---------- ---------- ----------
         7 xyz                        1200       1380        180

12 rows selected.
```

4. **Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.**

   SELECT INITCAP(EMP_NAME) "Name", LENGTH(EMP_NAME) "Length of Name" FROM EMPLOYEE WHERE EMP_NAME LIKE 'J%' OR EMP_NAME LIKE 'A%' OR EMP_NAME LIKE 'M%' ORDER BY EMP_NAME;

```
SQL> SELECT INITCAP(EMP_NAME) "Name", LENGTH(EMP_NAME) "Length of Name" FROM EMPLOYEE WHERE EMP_NAME
 LIKE 'J%' OR EMP_NAME LIKE 'A%' OR EMP_NAME LIKE 'M%' ORDER BY EMP_NAME;

Name                 Length of Name
-------------------- --------------
Aman                              4
```

5. **Write a query that produces the following for each employee: earns monthly.**

   SELECT EMP_NAME ||' earns '||EMP_SAL||' monthly' FROM EMPLOYEE;

```
SQL> SELECT EMP_NAME ||' earns '||EMP_SAL||' monthly' FROM EMPLOYEE;

EMP_NAME||'EARNS'||EMP_SAL||'MONTHLY'
--------------------------------------------------------------------------------
Aman earns 500 monthly
janak earns 5200 monthly
priyank earns 5200 monthly
smith earns 500 monthly
snehal earns 600 monthly
adama earns 700 monthly
aman earns 800 monthly
anita earns 900 monthly
sneha earns 1000 monthly
anamika earns 1100 monthly
xyz earns 1100 monthly

EMP_NAME||'EARNS'||EMP_SAL||'MONTHLY'
--------------------------------------------------------------------------------
xyz earns 1200 monthly

12 rows selected.
```

6. **Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM.**

   SELECT TO_CHAR(SYSDATE, 'fmDDTH') || ' of ' || TO_CHAR(SYSDATE, 'fmMonth') || ', ' ||TO_CHAR(SYSDATE, 'YYYY') || ', ' || TO_CHAR(SYSDATE, 'HH24:MI:SS AM') "DATE" FROM DUAL;

```
SQL> SELECT TO_CHAR(SYSDATE, 'fmDDTH') || ' of ' || TO_CHAR(SYSDATE,
  2  'fmMonth') || ', ' ||TO_CHAR(SYSDATE, 'YYYY') || ', ' || TO_CHAR(SYSDATE,
  3  'HH24:MI:SS AM') "DATE" FROM DUAL;

DATE
-----------------------------------
15TH of September, 2018, 12:04:33 PM
```

7. **Write a query to calculate the annual compensation of all employees (sal+comm.).**

   SELECT EMP_SAL+EMP_COMM "COMPENSATION" FROM EMPLOYEE;

```
SQL> SELECT EMP_SAL+EMP_COMM "COMPENSATION" FROM EMPLOYEE;

COMPENSATION
------------
        1100
        6000
        6000
         900
        1100
        1300
        1500
        1700
        1900
        2100
        2100

COMPENSATION
------------
        2300

12 rows selected.
```

PRACTICAL -5

AIM: Displaying data from Multiple Tables (join).

1. **Give details of customers ANIL.**

   SELECT D1.Acc_no, D1.B_NAME, D1.AMMOUNT, D1.ADATE, C1.CITY, B1.CITY FROM DEPOSITE D1, CUSTOMER9 C1, BRANCH B1 WHERE D1.C_NAME = C1.CNAME AND D1.B_NAME=B1.BNAME AND D1.C_NAME='ANIL';

```
SQL> SELECT  D1.Acc_no, D1.B_NAME, D1.AMMOUNT, D1.ADATE,  C1.CITY, B1.CITY FROM DEPOSITE  D1, CUSTOM
ER9  C1, BRANCH B1 WHERE D1.C_NAME = C1.CNAME AND D1.B_NAME=B1.BNAME AND D1.C_NAME='ANIL';

no rows selected
```

2. **Give name of customer who are borrowers and depositors andhaving living city nagpur.**

   SELECT C1.CNAME FROM CUSTOMER9 C1,DEPOSITE D1,BORROW B1 WHERE C1.CITY='NAGPUR' AND C1.CNAME=D1.C_NAME AND D1. C_NAME = B1.CNAME;

```
SQL> SELECT C1.CNAME FROM CUSTOMER9 C1,DEPOSITE D1,BORROW B1 WHERE C1.CITY='NAGPUR' AND C1.CNAME=D1.
C_NAME AND D1. C_NAME = B1.CNAME;

no rows selected
```

3. **Give city as their city name of customers having same living branch.**

   SELECT C.CITY FROM CUSTOMER33 C, BRANCH B WHERE C.CITY=B.CITY ;

```
SQL> SELECT C.CITY FROM CUSTOMER33 C,BRANCH B WHERE C.CITY=B.CITY ;

no rows selected
```

4. **Write a query to display the last name, department number, and department name for all employees.**

   SELECT E.EMP_NAME ,E.DEPT_NO,D. ACC_NO FROM EMPLOYEEE,DEPOSITE D WHERE E.DEPT_NO=D. ACC_NO;

```
SQL> SELECT E.EMP_NAME ,E.DEPT_NO,D. ACC_NO  FROM EMPLOYEE E,DEPOSITE D WHERE E.DEPT_NO=D. ACC_NO;

no rows selected
```

5. **Create a unique listing of all jobs that are in department 30. Include the location of the department in the output.**

   SELECT J.JOB_ID,J.JOB_TITLE,E.DEPT_NO,D.DEPT_CITY FROM JOB J,EMPLOYEE E,DEPARTMENT D WHERE J.JOB_ID=E.JOB_ID AND E.DEPT_NO=E.DEPT_NO AND E.DEPT_NO=30;

```
SQL>  SELECT J.JOB_ID,J.JOB_TITLE,E.DEPT_NO,D.DEPT_CITY FROM JOB J,EMPLOYEE E,DEPARTMENT D WHERE J.J
OB_ID=E.EMP_NO AND E.DEPT_NO=E.DEPT_NO AND E.DEPT_NO=30;

no rows selected
```

6. **Write a query to display the employee name, department number, and for all employees who work in RAJKOT**

   SELECT EMP_NAME , EMP_NAME FROM EMPLOYEE WHERE CITY='RAJKOT';

EMP_NO	EMP_NAME
101	KEYUR
105	HARDIK

2 rows returned in 0.00 seconds

7. **Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.**

   SELECT E.EMP_NAME"EMPLOYEE",E.EMP_NO"EMP #",EM.MNG_ NAME FROM " MANAGER", EM.MNG_NO "MGR#", FROM EMPLOYEE E EMPLOYEE_MANAGER EM WHERE E.EMP_NO=EM.EMP_NO;

EMPLOYEE	EMP #	MNG #	MANAGER
NIKUL	105	5	HIREN
NIKUL	105	5	HIREN
KARTIK	104	4	VIVEK
VIKAS	103	3	HIRAL
ABHISHEK	102	2	VIMAL
AMAN	101	1	ASHOK

6 rows returned in 0.00 seconds CSV Export

8. **Create a query to display the name and hire date of any employeehired after employee SCOTT.**

   SELECT E.EMP_NAME, EM.EMP_HIREDATE FROM EMPLOYEE E,EMPLOYEE_MANAGER EM WHERE E.EMP_NO =EM.EMP_NO AND EM.

EMP_ HIREDATE > (SELECT EMP_HIREDATE FROM EMPLOYEE _MANAG
ER WHERE EMP_NAME=’SCOTT’ ;

EMP MAME	EMP HTREDATE
NIML	2S-DEC-08
NIML	2S-DEC-08

2 rows returned in 0.02 seconds

C<'v E:pcd

PRACTICAL -6

AIM: To apply the concept of Aggregating Data using Group functions.

1. **List total deposit of customer having account date after 1-jan-96.**

   SELECT SUM(AMOUNT) FROM DEPOSIT WHERE ADATE >' 1-jan-96';

```
SQL>
SQL> SELECT SUM(AMOUNT) FROM DEPOSIT WHERE ADATE >' 1-jan-96';

SUM(AMOUNT)
-----------
```

2. **List total deposit of customers living in city Nagpur.**

   SELECT SUM(D.AMMOUNT) FROM DEPOSITE D ,CUSTOMER12 C WHERE C.CITY='NAGPUR' AND C.CNAME=D.C_NAME;

```
SQL> SELECT SUM(D.AMMOUNT) FROM DEPOSITE  D ,CUSTOMER12 C WHERE C.CITY='NAGPUR' AND C.CNAME=D.C_NAME
;

SUM(D.AMMOUNT)
--------------
```

3. **List maximum deposit of customers living in bombay.**

   SELECT MAX(D.AMMOUNT) FROM DEPOSITE D,CUSTOMER12 C WHERE C.CITY='BOMBAY' AND C.CNAME=D.C_NAME;

```
SQL> SELECT MAX(D.AMMOUNT) FROM DEPOSITE  D,CUSTOMER12 C WHERE C.CITY='BOMBAY' AND C.CNAME=D.C_NAME;


MAX(D.AMMOUNT)
--------------
```

4. **Display the highest, lowest, sum, and average salary of allemployees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.**

   SELECT MAX (EMP_SAL) "MAXIMUM" ,MIN(EMP_SAL) "MINIMUM",SUM (EMP_SAL) "SUM", AVG (EMP_SAL) "AVERAGE" FROM EMPLOYEE;

```
SQL> SELECT MAX (EMP_SAL) "MAXIMUM" ,MIN(EMP_SAL) "MINIMUM",SUM (EMP_SAL) "SUM", AVG (EMP_SAL) "AVER
AGE" FROM EMPLOYEE;

   MAXIMUM    MINIMUM        SUM    AVERAGE
---------- ---------- ---------- ----------
      5200        500      18800 1566.66667
```

5. **Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.**

   SELECT MAX(EMP_SAL)-MIN(EMP_SAL) "DIFFERENCE" FROM EMPLOYEE;

```
SQL> SELECT MAX(EMP_SAL)-MIN(EMP_SAL) "DIFFERENCE" FROM EMPLOYEE;

DIFFERENCE
----------
      4700
```

6. **Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998**

   SELECT COUNT (EMP_NO) FROM EMPLOYEE;

```
SQL> SELECT COUNT (EMP_NO) FROM EMPLOYEE;

COUNT(EMP_NO)
-------------
           12
```

7. **Find the average salaries for each department without displayingthe respective department numbers.**

   SELECT AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO;

```
SQL> SELECT AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO;

AVG(EMP_SAL)
------------
         700
        1100
         500
         550
        5200
         900
         800
        1000
        1100
        1200

10 rows selected.
```

8. **Write a query to display the total salary being paid to each job title, within each department.**

SELECT DEPT_NO,SUM(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO;

```
SQL> SELECT DEPT_NO,SUM(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO;

   DEPT_NO SUM(EMP_SAL)
---------- ------------
         6          700
        11         1100
         5          500
         4         1100
       110        10400
         8          900
         7          800
         9         1000
        10         1100
        12         1200

10 rows selected.
```

9. **Find the average salaries > 2000 for each department without displaying the respective department numbers.**

   SELECT AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO HAVING AVG(EMP_SAL)
   > 2000;

```
SQL> SELECT AVG(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO HAVING AVG(EMP_SAL) > 2000;

AVG(EMP_SAL)
------------
        5200
```

10. **Display the job and total salary for each job with a total salaryamount exceeding 3000, in which excludes president and sorts the list by the total salary.**

    SELECT DEPT_NO,SUM(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO HAVING SUM(EMP_SAL) > 3000 ORDER BY SUM(EMP_SAL);

```
SQL> SELECT DEPT_NO,SUM(EMP_SAL) FROM EMPLOYEE GROUP BY DEPT_NO HAVING SUM(EMP_SAL) > 3000 ORDER BY
SUM(EMP_SAL);

   DEPT_NO SUM(EMP_SAL)
---------- ------------
       110        10400
```

11. **List the branches having sum of deposit more than 5000 and locatedin city bombay.**

    SELECT D.B_NAME FROM DEPOSITE D , BRANCH B WHERE D.B_NAME =B.BNAME AND B.CITY='BOMBAY' GROUP BY D.B_NAME HAVING SUM(D.AMMOUNT) > 5000;

```
SQL> SELECT D.B_NAME FROM DEPOSITE D , BRANCH B WHERE D.B_NAME
  2  =B.BNAME AND B.CITY='BOMBAY' GROUP BY D.B_NAME HAVING SUM(D.AMMOUNT) > 5000;

no rows selected

----
```

PRACTICAL -7

AIM: To solve queries using the concept of sub query.

1. **Write a query to display the last name and hire date of any employeein the same department as SCOTT. Exclude SCOTT.**

   SELECT EMP_NAME FROM EMPLOYEE WHERE DEPT_NO = SELECT DEPT_NO FROM EMPLOYEE WHERE EMP_NAME LIKE 'SCOTT' AND EMP_NAME <> 'SCOTT';

EMP_NAME	EMP_HIREDATE
AMAN	01-JAN-91
ABHISHEK	10-AUG-95

2 rows returned in 0.00 seconds CSV Export

2. **Give name of customers who are depositors having same branch cityof mr. sunil.**

   SELECT D1.CNAME FROM DEPOSITE D1, BRANCH B2, D1.BNAME AND B2.CITY IN (SELECT B1.CITY FROM DEPOSITE D2 BRANCH B1 WHERE D2.CNAME='ANIL' AND D2.BNAME=B1.BNAME);

CNAME	BNAME	CITY
VRCE	AJNI	NAGPUR
AJNI	DHARAMPETH	NAGPUR

3. **Give deposit details and loan details of customer in same citywhere pramod is living.**

   SELECT D1.ACTNO,D1.BNAME,D1.AMOUNT,D1.ADATE ,BR1.LOANNO, BR1. BNAME BR1.AMOUNT FROM DEPOSITE D1 BORROW BR1 CUSTOMER C1 WHERE C1.CNAME = D1.CNAME AND D1.CNAME=BR1.CNAME AND C1.CITY IN (SELECT C2.CITY FROM CUSTOMER C2 WHERE C2.CNAME ='PRAMOD');

4. **Create a query to display the employee numbers and last names ofall employees who earn more than the average salary. Sort the results in ascending order of salary.**

   SELECT EMP_NO,EMP_NAME FROM EMPLOYEE WHERE EMP_SAL > (SELECT AVG(EMP_SAL) FROM EMPLOYEE ORDER BY EMP_SAL);

EMP_NO	EMP_NAME
103	KAJAL
105	HARDIK

5. **Give names of depositors having same living city as mr. anil andhaving deposit amount greater than 2000.**

   SELECT D1.CNAME FROM DEPOSIT D1,CUSTOMER C1 WHERE D1.AMOUNT > 2000 D1.CNAME=C1.CNAME AND C1.CITY IN (SELECT C2.CITY FROM CUS TOMER C2 WHERE C2.CNAME='ANIL');

BNAME	AMOUNT	CNAME	CITY
SUNIL	5000	AJNI	NAGPUR

1 rows returned in 0.00 seconds

6. **Display the department number, name, and job for every employee inthe Accounting department.**

   SELECT D.DEPT_NO,D.DEPT_NAME,E.JOB_ID FROM DEPARTMENT D,EMPLOYEE E WHERE D.DEPT_NO=E.DEPT_NO AND D.DEPT_NAME ='ACCOUNTING';

DEPT_NO	DEPT_NAME	NAME
101	ACCOUNTING	DARSHAN
102	ACCOUNTING	NILESH
103	ACCOUNTING	JAY
108	ACCOUNTING	AMIT

4 rows returned in 0.00 seconds CSV Export

7. **List the name of branch having highest number of depositors.**

   SELECT D1.BNAME FROM DEPOSIT D1 GROUP BY D1.BNAME HAVING CO UNT(D1.CNAME) > = ALL (SELECT COUNT (D2.CNAME) FROM DEPOSIT D2 GROUP BY D2.BNAME);

CNAME	BNAME	CITY
VRCE	AJNI	NAGPUR
AJNI	DHARAMPETH	NAGPUR

8. **Give the name of cities where in which the maximum numbers of branches are located.**

SELECT B1.CITY FROM BRANCH B1 GROUP BY B1.CITY HAVING COUNT (B1.BNAME) > ALL (SELECT COUNT(B2.BNAME) FROM BRANCH B2 WHERE B1.CITY = B2.CITY GROUP BY B2.CITY);

BNAME	CITY
AJNI	NAGPUR
DHARAMPETH	NAGPUR

2 rows returned in 0.00 seconds

9. **Give name of customers living in same city where maximumdepositors are located.**

SELECT C1.NAME FROM CUSTOMER C1 WHERE C1.CITY IN (SELECT C2. CITY FROM DEPOSIT D1,CUSTOMER C2 WHERE C2.CNAME = D1.C NA ME) GROUP BY C2.CITY HAVING COUNT (D1.CNAME) > ALL (SELECT COUNT (D2.CNAME) FROM DEPOSIT D2 CUSTOMER C3 WHERE D2.CNAME = C3.CNAME GROUP BY C3.CITY));

CNAME	BNAME	CITY
VRCE	AJNI	NAGPUR
AJNI	DHARAMPETH	NAGPUR

PRACTICAL -8

AIM: To Manipulate Data using various SQL commands.

1. **Give 10% interest to all depositors.**

   UPDATE DEPOSIT SET AMOUNT = AMOUNT + (AMOUNT*10/100);

```
SQL> UPDATE DEPOSIT SET AMOUNT = AMOUNT + (AMOUNT*10/100);

0 rows updated.
```

2. **Give 10% interest to all depositors having branch vrce**

   UPDATE DEPOSITE SET AMMOUNT = AMMOUNT + (AMMOUNT*10/100) WHEREB_NAME =’vrce’;

```
SQL> UPDATE DEPOSITE SET AMMOUNT = AMMOUNT + (AMMOUNT*10/100) WHERE
  2  B_NAME ='vrce';

1 row updated.
```

3. **Give 10% interest to all depositors living in nagpur and having branch city bombay.**

   UPDATE DEPOSITE SET AMMOUNT = AMMOUNT + (AMMOUNT*10/100) WHERE C_NAME IN (SELECT CNAME FROM CUSTOMER12 WHERE CITY=’NAGPUR’) AND B_NAME IN(SELECT BNAME FROM BRANCH WHERE CITY=’BOMBAY’);

```
SQL> UPDATE DEPOSITE SET AMMOUNT = AMMOUNT + (AMMOUNT*10/100) WHERE C_NAME IN (SELECT CNAI
TOMER12 WHERE CITY='NAGPUR') AND B_NAME IN(SELECT BNAME FROM BRANCH WHERE CITY='BOMBAY');

0 rows updated.
```

4. **Write a query which changes the department number of all employees with empno 7788’s job to employee 7844’current department number.**
   Transfer 10 Rs from account of anil to sunil if both are having same branch.

   UPDATE DEPOSITE SET AMMOUNT =AMMOUNT -10 WHERE C_NAME =’ANIL’ AND B_NAME IN (SELECT D1.B_NAME FROM DEPOSIT D1 WHERE D1.C_NAME = ’SUNIL’);

```
SQL> UPDATE DEPOSITE SET AMMOUNT =AMMOUNT -10 WHERE C_NAME ='ANIL' AND B_NAME IN (SELECT
ROM DEPOSIT D1 WHERE D1.C_NAME = 'SUNIL' );

0 rows updated.
```

UPDATE DEPOSITE SET AMMOUNT=AMMOUNT+10 WHERE C_NAME='SUNIL' AND B_NAME IN (SELECT D2.B_NAME FROM DEPOSIT D2 WHERE D2.C_NAME='ANIL');

```
SQL>  UPDATE DEPOSITE SET AMMOUNT=AMMOUNT+10 WHERE C_NAME='SUNIL' AND B_NAME IN (SELECT
ON DEPOSIT D2 WHERE D2.C_NAME='ANIL');

0 rows updated.
```

5. **Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.**

   UPDATE DEPOSITE SET AMMOUNT = AMMOUNT + 100 WHERE C_NAME IN (SELECT D1.C_NAME FROM DEPOSITE D1 GROUP BY D1.B_NAME HAVING AVG(D1.AMMOUNT) > = ALL (SELECT MAX(D2.AMMOUNT) FROM DEPOSITE D2 WHERE D1.B_NAME = D2.B_NAME GROUP BY D2.B_NAME));

```
SQL>  UPDATE DEPOSITE SET AMMOUNT=AMMOUNT+10 WHERE C_NAME='SUNIL' AND B_NAME IN (SELECT
ON DEPOSIT D2 WHERE D2.C_NAME='ANIL');

0 rows updated.
```

6. **Delete depositors of branches having number of customers between 1 to 3.**

   DELETE FROM DEPOSITE WHERE C_NAME IN (SELECT D1.C_NAME FROM DEPOSITE D1 GROUP BY D1.B_NAME HAVING COUNT(D1.C_NAME) BETWEEN 1 AND 3);

```
Table dropped.

0.89 seconds
```

7. **Delete deposit of vijay.**

   DELETE FROM DEPOSITE WHERE C_NAME='VIJAY';

```
SQL> DELETE FROM DEPOSITE WHERE C_NAME='VIJAY';

0 rows deleted.
```

8. **Delete borrower of branches having average loan less than 1000.**

   DELETE FROM BORROW WHERE CNAME IN (SELECT B.CNAME FROM

BORROW B GROUP BY B.BNAME HAVING AVG(B.AMOUNT)< 1000);

```
SQL> delete from borrow where cname in (select cname from borrow B group by B.bname havi
unt)<1000);

0 rows deleted.
```

PRACTICAL - 9

AIM: To apply the concept of security and privileges.

This chapter describes PointBase security and privileges. Schemas are an integral part of security in PointBase. When creating a PointBase user, they do not have any access privileges to schemas or other data objects within the database. The PointBase RDBMS only permits the schema owner to grant privileges to the schema and data objects within the schema. The schema owner can grant privileges to the following data objects in the schema:

- Tables
- Columns
- SQL Procedures and Functions

Table 1 describes the privileges that the schema owner can grant users for tables and columns:

Table 1 : User Privileges for Tables and Columns	
Privilege Statements	**Privilege** **Description**
DELETE	Allows a user to delete rows from tables within the schema
INSERT	Allows a user to insert rows of data into tables within the schema
REFERENCES	Allows a user to set up references to primary keys within the schema
SELECT	Allows a user to select rows from tables within the schema
TRIGGER	Allows a user to create triggers on tables within the schema
UPDATE	Allows a user to update rows in tables within the schema
EXECUTE	Allows users to execute functions or stored procedures within the schema

Granting and Revoking Privileges

When a PointBase database is first created the only user is the default user PUBLIC with a password of PUBLIC. The PUBLIC user owns the default PUBLIC schema. For security reasons, PointBase does not recommend using this schema to store sensitive data. Like any PointBase user, PUBLIC must be granted the appropriate privileges to access data objects in schema owned by other users.

The PUBLIC user can be used initially to create new users and new schema. The PUBLIC user will own any new schema that it creates unless otherwise specified during schema creation. New users are then able to create their own new schema and users, and grant appropriate privileges on schema that they own. All new users must be granted privileges to use the PUBLIC schema if this is required.

To grant the ability for a user to pass a privilege on to other users once granted, you must specify the optional WITH GRANT OPTION qualifier when granting the privilege.

GRANT Statement Syntax

GRANT *privilege-list*
ON object
TO user-list [WITH GRANT OPTION]

Use the GRANT statement to grant privileges on a data object. The following describes the GRANT statement syntax.

Privilege-list Syntax

privilege [, privilege [, privilege]...] | ALL PRIVILEGES

Privilege Syntax

SELECT [(column-name [, column-name]...)]

| DELETE

| INSERT [(column-name [, column-name]...)]

| UPDATE [(column-name [, column-name]...)]

| REFERENCES [(column-name [, column-name]...)]

| TRIGGER [(column-name [, column-name]...)]

| EXECUTE

Usage Notes

- If you do not include one or more of these privileges in the GRANT statement, an error will be raised.

- If the optional *column-names* are not specified for the SELECT, INSERT, UPDATE, REFERENCES and TRIGGER privileges, the GRANT is applied to every column in the table to which the grant is applied.
- If you execute a GRANT statement that contains privileges that you don't have or for which you do not have the right to grant, then PointBase raises an error.

Object Syntax

[TABLE] *table-name*
|SPECIFIC *routine_type specific_routine-name*
|*routine_type routine_name* (*parameter_types_list*)
[TRIGGER] *trigger-name*

Usage Notes

- If you grant a privilege on an SQL Function or Procedure, then the user can only EXECUTE that SQL Function or Procedure. The user cannot access tables that the SQL Function or Procedure uses.

User-list Syntax

user [, user]... [WITH GRANT OPTION] | PUBLIC

Usage Notes

- If you do not specify WITH GRANT OPTION, the user cannot pass the same privilege on to others. However, if you do specify WITH GRANT OPTION, you have given the user permission to pass on the privilege to other users.
- Granting a privilege to the user PUBLIC only grants the privilege to the default PointBase PUBLIC user and is not the same as granting a global privilege to all users.
- If you grant a privilege with the optional WITH GRANT OPTION and then grant the same privilege without this option (without first revoking the original privilege) the user retains the WITH GRANT OPTION.

Examples

- The following statement grants the SELECT privilege on the CUSTOMER_TBL table to the user MARKETING_MGR.

```
GRANT SELECT
ON customer_tbl
TO marketing_mgr;
```

- The following GRANT statement allows the user FINANCIAL_MGR to delete, insert and update rows from the DISCOUNT_CODE_TBL table; it also allows this user to grant the same privileges to others**.**

GRANT DELETE,INSERT,UPDATE

ON discount_code_tbl

TO financial_mgr

WITH GRANT OPTION;

- The following GRANT statement allows the user HR_MGR to have ALL PRIVILEGES on the table SALES_REP_DATA_TBL. However, the user HR_MGR will only be granted privileges that the user granting the privileges has the right to grant. For example, if the user granting the privileges does not have the right to grant DELETE privileges, the HR_MGR will not have the delete privilege.

GRANT ALL PRIVILEGES

ON sales_rep_data_tbl

TO hr_mgr

REVOKE Statement Syntax

REVOKE [GRANT OPTION FOR] *privilege_list*

ON *object*

FROM *user_name* [RESTRICT | CASCADE]

The REVOKE statement takes privileges away from users. The arguments are similar to the GRANT statement. The major difference is the additional RESTRICT or CASCADE keyword and the GRANT OPTION FOR clause. The following describes the optional clauses GRANT OPTION FOR and RESTRICT or CASCADE.

NOTE: If none of the privileges that you are trying to revoke actually exist, an error is raised.

RESTRICT | CASCADE

If you use RESTRICT keyword, the privilege will be revoked only from the specified user. If the specified user granted had the WITH GRANT OPTION and granted the same privilege to other users, they will retain the privilege.

If you use CASCADE, it will revoke the privilege and any dependent privileges as a result of your grant. A dependent privilege is one that could exist, if you granted the privilege that you're trying to revoke, which is what you are trying to achieve as a result of your REVOKE statement.

If the optional RESTRICT or CASCADE keywords are not used, PointBase uses RESTRICT by default.

GRANT OPTION FOR

If he optional GRANT OPTION FOR clause is used, the WITH GRANT OPTION right is revoked. The actual privilege itself is not revoked. the GRANT OPTION is revoked. CASCADE and RESTRICT may be used in the same way as a normal REVOKE statement.

PRACTICAL - 10

AIM: To study Transaction control commands.

TCL Commands in SQL- Transaction Control Language Examples: Transaction Control Language can be defined as the portion of a database language used for maintaining consistency of the database and managing transactions in database. A set of SQL statements that are co-related logically and executed on the data stored in the table is known as transaction. In this tutorial, you will learn different TCL Commands in SQL with examples and differences between them.

1. Commit Command
2. Rollback Command
3. Savepoint Command

TCL Commands in SQL- Transaction Control Language Examples

The modifications made by the DML commands are managed by using TCL commands. Additionally, it makes the statements to grouped together into logical transactions.

TCL Commands

There are three commands that come under the TCL:

1. Commit

The main use of Commit command is to make the transaction permanent. If there is a need for any transaction to be done in the database that transaction permanent through commit command. Here is the general syntax for the Commit command:

COMMIT;

For Example

UPDATE STUDENT SET STUDENT_NAME = ‘Maria’ WHERE STUDENT_NAME = ‘Meena’;

COMMIT;

By using the above set of instructions, you can update the wrong student name by the correct one and save it permanently in the database. The update transaction gets completed when commit is used. If commit is not used, then there will be lock on ‘Meena’ record till the rollback or commit is issued.

Now have a look at the below diagram where ‘Meena’ is updated and there is a lock on her record. The updated value is permanently saved in the database after the use of commit and lock is released**.**

Student ID	Student Name
78	Jane
79	Meena

After Update

Student ID	Student Name
78	Jane
79	Maria

After Commit

Student ID	Student Name
78	Jane
79	Maria

2. Rollback

Using this command, the database can be restored to the last committed state. Additionally, it is also used with savepoint command for jumping to a savepoint in a transaction.

The general syntax for the Rollback command is mentioned below:

Rollback to savepoint-name;

For example

UPDATE STUDENT SET STUDENT_NAME = ‘Manish’ WHERE STUDENT_NAME = ‘Meena’;

ROLLBACK;

This command is used when the user realizes that he/she has updated the wrong information after the student name and wants to undo this update. The users can issues

ROLLBACK command and then undo the update. Have a look at the below tables to know better about the implementation of this command.

Student ID	Student Name
78	Jane
79	Meena

After Update

Student ID	Student Name
78	Jane
79	Manish

After Rollback

Student ID	Student Name
78	Jane
79	Meena

3. Savepoint

The main use of the Savepoint command is to save a transaction temporarily. This way users can rollback to the point whenever it is needed.

The general syntax for the savepoint command is mentioned below:

savepoint savepoint-name;

For Example

Following is the table of a school class

ID	Name
98	Anita
99	Maria
100	Katilyn

Use some SQL queries on the above table and then watch the results

INSERT into CLASS VALUES (101, ‘Rahul);

Commit;

UPDATE CLASS SET NAME= ‘Tyler’ where id= 101

SAVEPOINT A;

INSERT INTO CLASS VALUES (102, ‘Zack’);

Savepoint B;

INSERT INTO CLASS VALUES (103, ‘Bruno’)

Savepoint C;

Select * from Class;

The result will look like

Id	Name
98	Anita
99	Maria
100	Katilyn
101	Tyler
102	Zack
103	Bruno

Now rollback to savepoint B

Rollback to B;

SELECT * from Class;

Id	Name
98	Anita
99	Maria
100	Katilyn
101	Tyler
102	Zack

Now rollback to savepoint A

rollback to A;

SELECT * from class;

Id	Name
98	Anita
99	Maria
100	Katilyn
101	Tyler