LAB MANUAL
FOR
DBMS LAB

1.
SQL BASICS

The structure queries language is a language that enable to create and operate on relational database ,which are sets of related information stored in tables .
SQL has clearly established itself as the standard relational database language.

PROCESSING CAPABILITY OF SQL:

The various capability of sql are:

1. DATA DEFINITION LANGUAGE(DDL):
   The sql DDL provides commands for defining relations schemas ,deleting relations ,creating index and modifying relations schemas.

2. INTERACTIVE DATA MANIPULATION LANGUAGE(DML):
   The sql DML includes the queries language based on both the relational algebra and the tuples relational calculas. It includes also command to insert ,delete and modifying in the database.

3. EMBEDDED DATA MANIPULATION LANGUAGE:
   The embedded form of sql is designed for use within general purpose programming language such as pl/1,cobol ,fortran,pascal and c.

4. VIEW DEFINITION :
   The sql DDL also includes commands for defining views

5. AUTHORIZATION :
   The sql DDL includes command for specifying access rights to relation and views.

6. INTEGRITY:
   The sql provides forms of integrity checking. Future products and standard of sql are likely to include enhanced features for integrity checking .

7. TRANSACTION CONTROL:
   Sql includes command for specifying the beginning and ending of transaction along with commands to have a control over transaction processing.

THE BASIC STRUCTURE OF AN SQL EXPRESSIONS CONSISTS OF THREE CLAUSES:

- SELECT
- FROM
- WHERE

A typical sql query has the form
Select a,b,c,d……
From a1,b1,c1……
Where p;

1.**SELECT CLAUSE:**

```
select branch-name from loan;
```

it will select all branch-name from the loan table

2.**WHERE CLAUSE:**

```
select loan-number from loan
where amount between 8000 and 9000
```

it will select all loan-number from loan where amount is between the 8000 and 9000

3.**FROM CLAUSE:**

```
select customer-name,borrower.loan-
number,amount

from borrower,loan
where borrower.loan-no=loan.loan-no
```

2.

TO CREATE DATABASE

```
SQL> create table Gaurav
  2  ( rollno  int,
  3  name  char(20),
  4   branch  char(20),
  5  sem  int );

Table created.
```

3.
CREATION OF TABLE WITH CONSTRAINTS:

```
SQL>  create table Gaurav
  2  ( empid  int constraint v1 primary key ,
 3  epmnm char(20) constraint v2 unique,
 4  desig char(20) default 'clerk',
5  dept char(20) constraint v3
6  check(dept in('edp','fin')),
7  salary int constraint v4 not null);

Table created.
```

Description of the above table:

```
SQL> desc Gaurav;
Name                          Null?    Type
----------------------------- -------- ----
EMPID                         NOT NULL NUMBER(38)
EPMNM                                  CHAR(20)
DESIG                                  CHAR(20)
DEPT                                   CHAR(20)
SALARY                        NOT NULL NUMBER(38)
```

8.
ALTER TABLE :

(a) Adding column : (ADD clause)

```
SQL> alter table Gaurav
2  add (marks int);
Table altered.
```

(b) Adding multiple columns clause) :

```
SQL> alter table Gaurav
 2  add (fav_sub char(20),stdid int);

Table altered.
```

(c) Changing column width :(MODIFY clause)

```
SQL>  alter table Gaurav
 2  modify branch char(10);

Table altered.
```

(d) Dropping column :(DROP clause)

```
SQL> alter table Gaurav
 2    drop column stdid;

 Table altered
```

(e) Adding NOT NULL : (MODIFY clause)

```
SQL> alter table Gaurav
 2  modify (rollno int not null);

Table altered.
```

(f) Dropping NOT NULL : (DROP clause)

```
SQL> alter table Gaurav
2  modify (rollno int not null);

Table altered
```

(g) Adding check constraint : (ADD clause)

```
SQL> alter table Gaurav
 2  add constraint v11 check(branch in('it','csc'));
Table altered.
```

(h) Dropping check constraint : (DROP clause)

```
SQL> alter table Gaurav
  2  drop constraint v11;

Table altered.
```

(i) Adding Primary key :

```
SQL> alter table Gaurav
  2  add constraint v11 primary key(name);
Table altered
.
```

(j) Removing Primary Key :

```
SQL> alter table Gaurav
  2  drop constraint v11;

Table altered.
```

(k) Dropping a primary key that have a dependent table:

```
SQL>  alter table Gaurav
 2  drop primary key cascade;

Table altered.
```

(l) Adding Foreign Key :

```
SQL> alter table Gaurav
  2  add constraint v11 foreign key(rollno)
  3  references employee;

Table altered.
```

(m)Dropping Foreign Key :

```
SQL> alter table Gaurav
  2  drop constraint v11;

Table altered.
```

The description of above altered table:

```
SQL> desc Gaurav;
 Name                      Null?    Type
 ------------------------- -------- ----
 ROLLNO                    NOT NULL NUMBER(38)
 NAME                               CHAR(20)
 BRANCH                             CHAR(10)
 SEM                                NUMBER(38)
 MARKS                              NUMBER(38)
 FAV_SUB                            CHAR(20)
 STDID                              NUMBER(38)
```


4.

ADD A RECORD TO DATABASE:

(1) Simple insertion:

```
SQL> insert into Gaurav
  2  values(8031,'Gaurav','CSE',4,760,'OOPS',1);

1 row created.
```

(2) Accepting values from users:

```
SQL> insert into Gaurav
  2  values(&rollno,&name,&branch,&sem,&marks,&fav_sub,&stdid);
Enter value for rollno: 6302
Enter value for name: 'Sourabh'
Enter value for branch: 'IT'
Enter value for sem: 4
Enter value for marks: 833
Enter value for fav_sub: 'PL'
Enter value for stdid: 1
old   2: values(&rollno,&name,&branch,&sem,&marks,&fav_sub,&stdid)
new   2: values(6302,'Sourabh','IT',4,833,'PL',1)

1 row created.

SQL> /
Enter value for rollno: 6058
Enter value for name: 'Sudhir'
Enter value for branch: 'CSE'
Enter value for sem: 4
Enter value for marks: 730
Enter value for fav_sub: 'DBMS'
Enter value for stdid: 2
old   2: values(&rollno,&name,&branch,&sem,&marks,&fav_sub,&stdid)
new   2: values(6058,'Sudhir','CSE',4,730,'DBMS',2)

1 row created.
```

(3) Inserting values into specific columns:

```
SQL> insert into Gaurav
  2  (rollno,name,branch,sem)
  3  values(7006,'Amit','cse',2);
1 row created.
```

The above inserted table is shown below:

```
SQL> select *from Gaurav;
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Vikas	CSE	4	760	OOPS	1
6302	Sourabh	IT	4	833	PL	1
6058	Sudhir	CSE	4	730	DBMS	2
7006	Amit	cse	2			

7.
UPDATING TABLES(MODIFY):

(a) Updating without where clause:

SQL> update Gaurav

```
  2  set name='Goru';

4 rows updated.

SQL> select *from Gaurav;

   ROLLNO NAME                 BRANCH          SEM     MARKS FAV_SUB
STDID
--------- -------------------- ---------- --------- --------- -------------------- ---------
    6070         Goru            CSE           4               760   OOPS              1
    6302         Goru            IT            4               833   PL                1
    6058         Goru            CSE           4               730   DBMS              2
    7006         Goru           cse            2
```

(b) Updating with where clause:

```
SQL> update Gaurav
  2  set name='dada' where rollno=6302;

1 row updated.

SQL> select *from Vikas_kapoor;

   ROLLNO NAME                 BRANCH          SEM     MARKS FAV_SUB  STDID
--------- -------------------- ---------- --------- --------- -------------------- ---------
    6070         Goru            CSE           4               760   OOPS              1
    6302         dada             IT           4               833   PL                1
    6058         Gaurav          CSE           4               730   DBMS              2
    7006         Gaurav          CSE           2
```

14.
Generating sub query:

```
SQL> update Gaurav
  2  set name=(select name from Gaurav where rollno=6302)
  3  where rollno=6058;
```

1 row updated.

SQL> select *from Gaurav;

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6302	Sourabh	IT	4	833	PL	1
6058	Sourabh	CSE	4	730	DBMS	2
7006	Amit	cse	2			

6.
DELETING RECORDS:

(a) Delete (SINGLE record)

```
SQL> delete from Gaurav
  2  where rollno=7006;

1 row deleted.

SQL> select *from Vikas_kapoor;

   ROLLNO NAME       BRANCH  SEM     MARKS FAV_SUB    STDID
--------- ------------------- ---------- --------- --------- -------------------- ---------
```

6070	Goru	CSE	4	760 OOPS	1
6302	Sourabh	IT	4	833 PL	1
6058	Sourabh	CSE	4	730 DBMS	2

(b) Delete (MULTIPLE record)

```
SQL> delete from Gaurav
 2  where branch='CSE';

2 rows deleted.
 SQL> select *from Gaurav;
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6302	Sourabh	IT	4	833	PL	1

(c) Delete (ALL records)

→ USING TRUNCATE

```
SQL> truncate table Gaurav;

Table truncated.

SQL> select *from Gaurav;

no rows selected
```

(d) Delete (ALL records)

→ USING DELETE

```
SQL> delete from Gaurav;

1 rows deleted.

SQL> select *from Gaurav;

no rows selected
```

(e) Deleting using sub query

```
SQL> delete from Gaurav
 2  where rollno=(select rollno from Gaurav where name='Sudhir');
```

1 row deleted.

SQL> select *from Gaurav;

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6302	Sourabh	IT	4	833	PL	1

DROPPING TABLE:

(a) Dropping a table that has a primary key:
SQL> drop table s;

Table dropped.

(b) Dropping a table that has a foreign key:
SQL> drop table sp;

Table dropped.

RETRIEVING DATA :

(a) Retrieving all records:
SQL> select *from Gaurav;

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6302	Sourabh	IT	4	833	PL	1
7006	Amit	CSE	2	729	FCP	1
6058	Sudhir	CSE	4	729	DBMS	2
6047	Richa	CSE	4	782	CAO	2

(b) Retrieving specific columns:

```
SQL> select rollno,branch,name from Gaurav;

 ROLLNO BRANCH     NAME
--------- ---------- --------------------
   6070    CSE       Goru
   6302    IT        Sourabh
   7006    CSE       Amit
   6058    CSE       Sudhir
   6047    CSE       Richa
```

(c) Printing with user defined headings:

```
SQL> select rollno as rno, branch as stream, name as stdname from Vikas_kapoor;

    RNO STREAM     STDNAME
--------- ---------- --------------------
   6070        CSE       Goru
   6302        IT        Sourabh
   7006        CSE       Amit
   6058        CSE       Sudhir
   6047        CSE       Richa
```


(b) Using Logical operators(AND, OR, NOT):

```
SQL> select * from Gaurav
  2  where(branch='CSE' AND marks=729);

  ROLLNO NAME       BRANCH  SEM   MARKS FAV_SUB     STDID
--------- ------------------- ---------- --------- --------- -------------------- ---------
   7006 Amit            CSE          2     729    FCP                  1
   6058 Sudhir          CSE          4     729    DBMS                 2
```

SQL> select * from Gaurav
2 where(branch='cse' OR marks>750);

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6302	Sourabh	IT	4	833	PL	1
6047	Richa	CSE	4	782	CAO	2

(c) Using BETWEEN AND:

SQL> select * from Gaurav
2 where marks between 730 and 800;

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6047	Richa	CSE	4	782	CAO	2

(d) Using IN Function:

SQL> select * from Gaurav
2 where marks in(729,760,782);

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
7006	Amit	CSE	2	729	FCP	1
6058	Sudhir	CSE	4	729	DBMS	2
6047	Richa	CSE	4	782	CAO	2

(e) Using LIKE Operator:

1). percent:

SQL> select * from Gaurav
2 where name like '%h%';

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6302	Sourabh	IT	4	833	PL	1
6058	Sudhir	CSE	4	729	DBMS	2

6047 Richa CSE 4 782 CAO 2

2). underscore:

```
SQL> select * from Gaurav
  2  where marks like '8__';
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6302	Sourabh	IT	4	833	PL	1

3). IS NULL:

```
SQL> select * from Gaurav
2  where stdid is null;
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
7006	Amit	CSE	2	729	FCP	

9.
ORDERING RECORDS:

a). ascending:

```
SQL> select * from Gaurav
  2  order by name asc;
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
7006	Amit	CSE	2	729	FCP	1
6047	Richa	CSE	4	782	CAO	2
6302	Sourabh	IT	4	833	PL	1

6058 Sudhir	CSE	4	729	DBMS	2
6070 Goru	CSE	4	760	OOPS	1

b). descending:

```
SQL> select * from Gaurav
  2  order by name desc;
```

ROLLNO	NAME	BRANCH	SEM	MARKS	FAV_SUB	STDID
6070	Goru	CSE	4	760	OOPS	1
6058	Sudhir	CSE	4	729	DBMS	2
6302	Sourabh	IT	4	833	PL	1
6047	Richa	CSE	4	782	CAO	2
7006	Amit	CSE	2	729	FCP	1

a). concat:

```
SQL> select name || ',' ||  branch from Gaurav;

NAME||','||BRANCH
------------------------------
Goru           ,CSE
Sourabh          ,IT
Amit           ,CSE
Sudhir          ,CSE
```

Richa ,CSE

b). initcap:
SQL> select initcap(name) from Gaurav;

INITCAP(NAME)

Goru
Sourabh
Amit
Sudhir
Richa

c). lower:
SQL> select lower(name) from Gaurav;

LOWER(NAME)
----
vikas
sourabh
amit
sudhir
richa

d). upper:
SQL> select upper(name) from Gaurav;

UPPER(NAME)

VIKAS
SOURABH
AMIT
SUDHIR
RICHA

11.

ORACLE FUNCTION:

a). add_months:
SQL> select sysdate, add_months('9-april-2005',4) from dual;

SYSDATE ADD_MONTH
--------- ---------
08-APR-05 09-AUG-05

b). last_day:
SQL> select sysdate,last_day('9-april-2005') from dual;

SYSDATE LAST_DAY(
--------- ---------
08-APR-05 30-APR-05

c). months_between:
SQL> select sysdate, months_between(sysdate,'02-nov-1985')from dual;

```
SYSDATE   MONTHS_BETWEEN(SYSDATE,'02-NOV-1985')
--------- ------------------------------------
08-APR-05                          233.22555
```

d). next_day:
SQL> select sysdate,next_day('9-april-2005','monday')from dual;

```
SYSDATE   NEXT_DAY(
--------- ---------
08-APR-05 11-APR-05
```

Ceil:
SQL> select ceil(months_between(sysdate,'02-nov-1985'))from dual;

```
CEIL(MONTHS_BETWEEN(SYSDATE,'02-NOV-1985'))
-------------------------------------------
                                        234
```

Floor:
SQL> select floor(months_between(sysdate,'02-nov-1985'))from dual;

```
FLOOR(MONTHS_BETWEEN(SYSDATE,'02-NOV-1985'))
--------------------------------------------
                                         233
```

Mod:
SQL> select mod(10,7) from dual;

MOD(10,7)

3
Power:
SQL> select power(2,3) from dual;

POWER(2,3)

8

Sqrt:
SQL> select sqrt(10) from dual;

SQRT(10)

3.1622777

Abs:
SQL> select abs(10)from dual;

ABS(10)

10

AGGREGATE FUNCTIONS:

♦ AVG()
SQL> select avg(marks) from Gaurav;

AVG(MARKS)

766.6
♦ MAX()
SQL> select max(marks) from Gaurav;

MAX(MARKS)

833

♦ **MIN()**
SQL> select min(marks) from Gaurav;

MIN(MARKS)

729

♦ **SUM()**

SQL> select sum(marks) from Gaurav;

SUM(MARKS)

3833

♦ **COUNT()**
SQL> select count(name) from Gaurav;

COUNT(NAME)

5

SQL> select count(*) from Gaurav;

COUNT(*)

5

10.
GROUPING FUNCION

GROUP BY clause:

```
SQL> select sum(marks) from  Gaurav
  2  group by stdid;

SUM(MARKS)
----------
      2322
      1511
```

HAVING clause:

```
SQL> select sum(marks) from Gaurav
  2  group by stdid
  3  having sum(marks)>2000;

SUM(MARKS)
----------
      2322
```

ALL clause:

```
SQL> select all name from Gaurav;
```

NAME

Goru
Sourabh
Amit
Sudhir
Richa

12.
SET OPERATIONS:

(a) Union:
SQL> select name from std union select name from Gaurav;

NAME

Amit
Richa
Sourabh
Sudhir
goru
amit
preeti
sourabh
goru

9 rows selected.
(b) Union All:

SQL> select rollno,name,branch,sem from Gaurav union all select *from std;

ROLLNO	NAME	BRANCH	SEM
6070	Goru	CSE	4
6302	Sourabh	IT	4

7006 Amit	**CSE**	**2**
6058 Sudhir	**CSE**	**4**
6047 Richa	**CSE**	**4**
6302 sourabh	**it**	**4**
6317 preeti	**it**	**4**
6070 goru	**cse**	**4**
7006 amit	**cse**	**2**

9 rows selected.