SQL

TABLE CREATION :

```
create table tablename(column1 datatype,column datatype);
```

copying from one table to another..

```
create table dept2 as select * from DEPT;
```

INSERTING VALUES :

```
insert into tablename values( column1,column2);
insert into database values(4,'anithamumbai'); //insert in the order the table created
```

to get from console

```
insert into tablename values(&number,'&string');
insert into database values(&no,'&fname','&fcity'); // to get values from user and to perform
multiple entries ny simple running ' n ' number of times
```

example 2

```
create table client (cid number(10),cname varchar2(10) not null);
insert into client values(10,null);
```

table creating in console way

```
insert into DISTRIBUTOR values(&code,'&co_name',&rep_no,&credit_limit);
```

ALTER :

```
alter table FRIENDS add fpincode number(20);
```

```
alter table  friends drop column fpincode; //drop column
alter table friends rename to DataBase;
alter table Database rename column fno to no;
```

UPDATE :

```
update DATABASE set fcity='madurai' where no=1;
update FRIENDS set values FPINCODE = 600092 where fname ='renga';  //to insert into empty values
update database set fname='raj' ,fcity='orissa' where no=7;
```

DELETE :

```
delete from DATABASE where no=3; //delete one row
```

```
select * from dept where location='MUMBAI';
```

in this case we should not plainly give mumbai its case senstitive...while getting data from the table make sure the case senstitive nature....

NESTED QUERIES :

```
NUM NAME                 BRANC TITLE                    TARGET      SALES
---------- -------------------- ----- -------------------- ---------- ----------
        40 anu                  che   rep                    50000     100000
        50 anusha               del   rep                    56000     450000
        60 anushka              kol   rep                    86000     800000
```

```
      CODE CO_NAME                  REP_NO CREDIT_LIMIT
---------- -------------------- ---------- ------------
        20 hcl                          40       500000
        21 infosys                      41       600000
        22 value leaf                   42       450000
```

select co_name from DISTRIBUTOR where REP_NO=(select num from SALESREP where name='anu');
prits hcl

select * from SALESREP where num=(select rep_no from DISTRIBUTOR where co_name='hcl');

```
       NUM NAME                 BRANC TITLE                    TARGET      SALES
---------- -------------------- ----- -------------------- ---------- ----------
        40 anu                  che   rep                    50000     100000
```

To perform intersection using sub queries

```
select * from SALESREP where num=(select num from SALESREP intersect select rep_no from DISTRIBUTOR);
```

```
NUM NAME                 BRANC TITLE                TARGET      SALES
---------- -------------------- ----- -------------------- ---------- ----------
        40 anu                  che   rep                   50000     100000
```

UNIONS & intersections : unions combine all the datas and print (without duplication):

```
ROLL_NO NAME                SCIENCE      MATHS
---------- ------------------- ---------- ----------
       101 renga                       94         98
       102 suren                       90        100
       103 nizar                       50         50
```

```
    ROLLNO NAME                PLACE
---------- ------------------- --------------------
       101 renga               saligramam
       102 suren               vandalur
       103 nizar               kaasimedu
```

INTERSECTIONS :

```
select name from results intersection SELECT name from S_DETAILS;
```

```
NAME
--------------------
nizar
renga
suren
```

MINUS :

movie

```
NAME                 COLLECTION
-------------------- ----------
mersal               2500000000
theri                1500000000
```

movie2

```
NAME                 COLLECTION
-------------------- ----------
mersal               2500000000
theri                1500000000
viswasam                      0
```

```
select * from movie2 minus select * from movie;
```

```
NAME                 COLLECTION
-------------------- ----------
viswasam                      0
```

PRIMARY KEY :

```
CREATE table employee(e_id number(10),e_name varchar2(20),e_designation varchar2(20),primary key (e_id));
```

or

```
CREATE table employee(e_id number(10) primary key ,e_name varchar2(20),e_designation varchar2(20));
```

```
Table EMPLOYEE created.


Name          Null?    Type

------------- -------- ------------

E_ID          NOT NULL NUMBER(10)

E_NAME                 VARCHAR2(20)

E_DESIGNATION          VARCHAR2(20)
```

FOREIGHN KEY REFERENCES :

```
create table emp_personal(e_id number(10) REFERENCES employee(e_id),phone number(10) , address varchar2(100));
```

employee table

```
      E_ID E_NAME               E_DESIGNATION

---------- -------------------- --------------------

       100 renga                ssd

       101 suren                trainer

       102 nizar                tutor
```

emp_personal

```
E_ID      PHONE ADDRESS
---------- ---------- ----------------------------------------------------------------------------------------------------
       100 9500179713 saidapet
       101 7989879792 uk
       102 9840751113 us
```

```
update EMP_PERSONAL set EID=200 where PHONE=9840751113
```

```
Error report -
ORA-02291: integrity constraint (RENGAORACLE.SYS_C007201) violated - parent key not found
```

```
delete from employee where e_id=200
```

```
Error report -
ORA-02292: integrity constraint (RENGAORACLE.SYS_C007201) violated - child record found
```

The foreighn key coloumn must contain the same data as present in the primary key (irespective of the number of rows in the primary key)..

once the references made we cannot remove the primary key data but we can remove the data from the forieghn key column being a child key...

in simple words primary key should not be altered...

to drop the primary key

```
alter table employee drop CONSTRAINT e_id;
```

SAVE POINT :

To save the whole script in a location (file) to retrieve later in case of any crashes occured...by using rollback options eventhough we used commit options..

suppose if we update a table after setting into savepoint then we can easily change the table into its original one by using rollback options......

```
savepoint savepointname;

update table movie set name='sarkar' where collection = 17000000;

table updated

rollback to savepointname;
```

after this the table changes into its original content ;

JOINS :

INNER JOINS- EQUALITY SHOULD BE THERE ALWAYS

```
create table orders (orderID number(3),customerID number(2),orderNO number(3));

insert into orders values(&orderID,&customerID,&orderNO);

select * from orders;
```

```
   ORDERID CUSTOMERID    ORDERNO
---------- ---------- ----------
      101        1        40
      102        2        45
      103       45        65
               104       50       75
      105       86       100
```

```
create table customers (customerID number(2),customerNAME varchar2(20),place varchar2(20),
balance number(5));
```

```
insert into customers values(&customerid,'&customername','&place',&balance);

select * from customers;


CUSTOMERID CUSTOMERNAME        PLACE                BALANCE

---------- ------------------- -------------------- ----------

         1 raj                trichy               25000

         2 ravi               salem                40000

         3 arun               chennai              60000

         4 jai                madurai              45000

         5 asmi               hyderabad            75000


select orders.orderid,customers.customername from orders inner join customers on
orders.customerid = customers.CUSTOMERID;


   ORDERID CUSTOMERNAME

---------- --------------------

       101 raj

       102 ravi
```

for three table

```
create table shippers(customerID number(2),shipperNAME varchar2(20));

insert into shippers values(&customerid,'&shippername');

select * from shippers;


CUSTOMERID SHIPPERNAME

---------- --------------------

         1 dhl
```

```
        20 st

        30 royalamail

        40 professional

        50 dtdc
```

```
select orders.orderid,customers.customername,shippers.shippername from ((orders inner join customers on orders.customerid = customers.CUSTOMERID)

inner join shippers on orders.customerid=shippers.customerid);
```

```
   ORDERID CUSTOMERNAME         SHIPPERNAME

---------- -------------------- --------------------

       101 raj                  dhl
```

```
select * from orders inner join customers on orders.customerid = customers.CUSTOMERID;
```

```
   ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME         PLACE
BALANCE

---------- ---------- ---------- ---------- -------------------- -------------------- ----------

       101          1         40          1 raj                  trichy                    25000

       102          2         45          2 ravi                 salem                     40000
```

```
select * from ((orders inner join customers on orders.customerid = customers.CUSTOMERID)

inner join shippers on orders.customerid=shippers.customerid);
```

```
ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE                BALANCE
CUSTOMERID SHIPPERNAME
---------- ---------- ---------- ---------- -------------------- -------------------- ---------- ---------- --------------------
       101          1         40          1 raj                  trichy                    25000          1 dhl
```

for unmatch records

select * from orders inner join customers on orders.ORDERID = customers.CUSTOMERID;

no rows found :

LEFT JOIN OR LEFT OUTER JOIN :-

returns all records from the left table as well as the matched records from the right table

if no match is found means it will return the left table records alone...

select * from orders left join customers on orders.customerid = customers.CUSTOMERID ORDER BY customers.customerid asc;

```
ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE                BALANCE
---------- ---------- ---------- ---------- -------------------- -------------------- ----------
       101          1         40          1 raj                  trichy                    25000

       102          2         45          2 ravi                  salem                    40000

       103         45         65

       105         86        100

       104         50         75
```

without matching records

```
        select * from orders left join customers on orders.orderid = customers.CUSTOMERID ;

        ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE
BALANCE

---------- ---------- ---------- ---------- -------------------- -------------------- ----------

   103        45        65

   102         2        45

   101         1        40

   105        86       100

   104        50        75
```

RIGHT OUTER JOINS :- same as left one...

```
select * from orders right join customers on orders.customerid = customers.CUSTOMERID ORDER BY
customers.customerid asc;
```

```
ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE              BALANCE

---------- ---------- ---------- ---------- -------------------- -------------------- ----------

   101         1        40        1 raj              trichy              25000

   102         2        45        2 ravi              salem               40000

                             3 arun            chennai              60000

                             4 jai             madurai             45000

                             5 asmi            hyderabad            75000
```

FULL OUTER JOIN :-

More like select *

select * from orders full outer join customers on orders.customerid = customers.CUSTOMERID;

```
   ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE
BALANCE

---------- ---------- ---------- ---------- -------------------- -------------------- ----------

       101          1         40          1 raj              trichy                 25000

       102          2         45          2 ravi              salem                  40000

                                          3 arun             chennai               60000

                                          4 jai             madurai               45000

                                          5 asmi             hyderabad             75000

       104         50         75

       103         45         65

       105         86        100
```

8 rows selected.

for unmatching records

select * from orders full outer join customers on orders.ORDERID = customers.CUSTOMERID;

```
ORDERID CUSTOMERID    ORDERNO CUSTOMERID CUSTOMERNAME        PLACE                BALANCE

---------- ---------- ---------- ---------- -------------------- -------------------- ----------

                                          1 raj             trichy               25000

                                          2 ravi             salem                40000

                                          3 arun             chennai               60000

                                          4 jai             madurai               45000

                                          5 asmi             hyderabad             75000

       103         45         65
```

```
      102        2       45

      101        1       40

      105       86      100

      104       50       75


10 rows selected.
```

SELF JOIN :-

its a regular join , where the table is joined with itself

```
CREATE TABLE EMP (ENO NUMBER(2),ENAME VARCHAR2(20),DESIGNATION VARCHAR2(20),SALARY
NUMBER(5),MANAGER NUMBER(2),DEP_NUM NUMBER(2));

insert into emp values(&ENO,'&ename','&designation',&salary,&manager,&dep_num);



select * from emp order by eno asc;

ENO ENAME                DESIGNATION              SALARY    MANAGER    DEP_NUM

---------- -------------------- -------------------- ---------- ---------- ----------

        1 suresh             salesman                7000         2        10

        2 ramesh              manager                17000         3        10

        3 rajesh             president             40000                  30

        4 naresh              clerk                 5000         5        20

        5 vignesh             manager                20000         3        20

        6 venkatesh            clerk                 8000         5


select e.ename employeename ,m.ename managername from emp e,emp m where
e.manager=m.eno;


EMPLOYEENAME        MANAGERNAME
```

```
-------------------- --------------------
suresh              ramesh
venkatesh           vignesh
naresh              vignesh
vignesh             rajesh
ramesh              rajesh
```

NATURAL JOIN :-

```
CREATE TABLE DEPT (DEPTNO NUMBER(4),DEPTNAME VARCHAR2(20),LOCATION VARCHAR2(20));
INSERT INTO DEPT VALUES(&DEPYNO,'&DEPTNAME','&LOCATION');
SELECT * FROM DEPT;
```

```
    DEPTNO DEPTNAME             LOCATION
---------- -------------------- --------------------
        10 SALES                MUMBAI
        20 HR                   DELHI
        30 ACCOUNTS             CHENNAI
        40 PRODUCTION           BENGALURU
```

```
SELECT * FROM EMP,DEPT WHERE EMP.DEP_NUM=DEPT.DEPTNO;
```

```
       ENO ENAME                DESIGNATION              SALARY    MANAGER    DEP_NUM     DEPTNO
DEPTNAME             LOCATION
---------- -------------------- -------------------- ---------- ---------- ---------- ---------- -------------------- --------------------
--
         1 suresh               salesman                   7000          2         10         10 SALES                MUMBAI
         2 ramesh               manager                   17000          3         10         10 SALES                MUMBAI
         4 naresh           clerk                5000        5       20       20 HR               DELHI

         5 vignesh          manager              20000       3       20       20 HR               DELHI

         3 rajesh           president            40000               30       30 ACCOUNTS         CHENNAI
```

getting two rows using OR operator

```
select * from DISTRIBUTOR where CO_NAME='hcl' or CO_NAME='infosys';
```

```
CODE CO_NAME              REP_NO CREDIT_LIMIT

---------- -------------------- ---------- ------------

      20 hcl                    40      500000

      21 infosys                50      600000
```

AND operator

```
select * from DISTRIBUTOR where CO_NAME='hcl' and REP_NO=40;
```

```
CODE CO_NAME              REP_NO CREDIT_LIMIT

---------- -------------------- ---------- ------------

      20 hcl                    40      500000
```

AGGREGATION OPERATIONS :

AVG ::

```
       ENO ENAME                DESIGNATION              SALARY    MANAGER    DEP_NUM
---------- -------------------- -------------------- ---------- ---------- ----------
         1 suresh               salesman                   7000          2         10
         2 ramesh               manager                   17000          3         10
         4 naresh               clerk                      5000          5         20
         5 vignesh              manager                   20000          3         20
         3 rajesh               president                 40000                    30
         6 venkatesh            clerk                      8000          5

6 rows selected.
```

select avg (salary)from emp;

```
AVG(SALARY)
-----------
 16166.6667
```

SUM :

select sum (salary)from emp;

```
SUM(SALARY)
-----------
      97000
```

MINIMUM :-

```
select min (salary)from emp;


MIN(SALARY)
-----------
      5000


               select min (salary) from emp where DESIGNATION='clerk';
               MIN(SALARY)
-----------
      5000
```

MAXIMUM :-

```
select max (salary)from emp;
select ename from emp where salary = ( select max (salary) from emp);


MAX(SALARY)
-----------
     40000
```

COUNT (NUMBER OF ROWS)

```
select count (salary)from emp;


COUNT(SALARY)
-------------
```

```
         6
```

order by : select ename from emp where salary > 5000 order by SALARY;

```
ENAME
--------------------
suresh
venkatesh
ramesh
vignesh
rajesh
```

aggregate using count :

select count(*) from emp where DESIGNATION='clerk'; //table emp

count : 2

select designation,count(*) from emp group by DESIGNATION;

```
DESIGNATION          COUNT(*)
-------------------- ----------
salesman                  1
clerk                2
president                 1
manager                   2
```

while using aggregate function wwe are not allowed to specify any other column along with it cos aggregate function returns only one record

```
select dep_num,count(*) from emp group by dep_num order by dep_num asc;
```

```
   DEP_NUM   COUNT(*)

---------- ----------

        10          2

        20          2

        30          1

                    1
```

```
select designation,sum(dep_num) from emp group by designation;
```

```
DESIGNATION          SUM(DEP_NUM)

-------------------- ------------

salesman                       10

clerk                      20

president                      30

manager                        30
```

```
SELECT designation,count(*) from emp where salary > 10000 group by DESIGNATION;
```

```
 DESIGNATION           COUNT(*)

-------------------- ----------

president                     1

manager                       2
```

HAVING CLAUSE :-

Conditions on aggregate function cannot be specified in a where clause cos where clause gets executed before group by clause

```
select dep_num , count (*) from emp group by dep_num having count(*) > =2;
```

```
DEP_NUM   COUNT(*)
---------- ----------
        20          2
        10          2
```

```
select designation ,count(*) from emp where dep_num=10 group by designation having COUNT(*)>0;
```

```
DESIGNATION          COUNT(*)
-------------------- ----------
salesman                      1
manager                       1
```

SUB QUERY USING EMP AND DEPT TABLE

how many employees work in mumbai location

```
select count(*) from emp where dep_num=(select deptno from dept where location='MUMBAI');
```

COUNT : 2

in which dept name does employee 'ramesh' work

```
select deptname from dept where DEPTNO = ( select dep_num from emp where ename='ramesh');
```

```
DEPTNAME
--------------------
SALES
```

average salary of employee in hr dept...

```
select avg (salary) from emp where dep_num=(select DEP_NUM from dept where deptname='HR');
```

```
AVG(SALARY)
-----------
     17800
```

IN USAGE :

use 'in' instead of '=' if sub query returns more than one record or if we are unsure whether the result may return one or more records...

```
select count(*)from emp where dep_num in (select deptno from dept2 where location='MUMBAI');
```

```
COUNT(*)
----------
       4
```

SUB QUERY WITH GROUP BY :-

per job how many employees are present in delhi ?

```
select count (*) from emp where dep_num in ( select deptno from dept where location='DELHI')
group by designation ;
```

```
  COUNT(*)
----------
         1
         1
```

```
select designation , count (*) from emp where dep_num in ( select deptno from dept where
location='DELHI') group by designation ;
```

```
DESIGNATION          COUNT(*)
-------------------- ----------
clerk                   1
manager                  1
```

per job how many employees of sales dept earn more than 10000?

```
select designation,count (*) from emp where salary >10000 and dep_num in ( select deptno from
dept where deptname='SALES')group by designation ;
```

```
DESIGNATION          COUNT(*)
-------------------- ----------
manager                  1
```

ORDER BY -using sub query :- (order by always followed by group by if both comes together means)

```
select designation,count (*) from emp where salary >5000 and dep_num in ( select deptno from
dept where deptname='SALES')group by designation order by DESIGNATION desc ;
```

```
DESIGNATION          COUNT(*)
```

```
-------------------- ----------
salesman                  1
manager                   1
```

select designation,count (*) from emp where salary >5000 and dep_num in (select deptno from dept where deptname='SALES')group by designation order by 1 desc ;

```
DESIGNATION          COUNT(*)
-------------------- ----------
salesman                  1
manager                   1
```