For showing the contents in Table:

1) select department\_id, department\_name

from departments

where department\_id = 210 OR department\_id = 240 //------>>> shows only 210th and 240th

/

2) select department\_id, department\_name

from departments

where department\_id IN (210,240) //------>>> shows only 210th and 240th

/

3) select department\_id, department\_name

from departments

where department\_id Between 210 And 240 //------->>> shows all between 210 & 240 (incliding 210, 240)

/

4) select department\_id, department\_name, location\_id //--------->>> No Row selected

from departments

where department\_name = 'noc'

/

5) select \* //--------->>> shows all names starts with 'C' letter

from departments //--------->>> character name must be in ''

where department\_name LIKE 'C%' //--------->>> '%' sign indicates 0 or more characters or '\_' indicates 1 chara.

/

6) select department\_id, department\_name //------->>> shows the aames having second chara 'a'.

from departments

where department\_name LIKE '\_a%'

/

7) select department\_id, department\_name //------>>> shows all names ending with chara 'g'

from departments

where department\_name LIKE '%g'

/

8) select department\_id, department\_name //------>>> shows all names having chara 'e'

from departments

where department\_name LIKE '%e%'

/

9) select department\_id, department\_name //--------->>> if you want to find by chara the % sign me be there after that chara

from departments

where department\_name LIKE '\_\_r%'

/

10)select department\_name, upper(department\_name), lower(department\_name)

from departments //-------->>> shows Original\_name\_list, UPPERCASE\_NAME\_LIST & lowercase\_name\_list

/

11)select department\_name, upper(department\_name) as uppercase, lower(department\_name) as lowercase

from departments //------->>> name must be continuous ( NOT ==> upper case & lower case)

/

Create Table emp\_test

(

Empno Number(4),

Ename Varchar2(10) Not Null,

Job Varchar2(9) Not Null,

Mgr Number (4),

Hiredate Date,

Sal Number(7,2) Not Null,

Comm Number(7, 2),

Deptno Number(2) Not Null

);

Create Table dept\_test

(

Deptno Number(2) Primary Key,

Deptname Varchar2(15) Not Null,

Location Varchar2(15)Not Null

)

Alter Table emp\_test

Add constraint emp\_dept\_test\_fk Foreign Key(Deptno)

References dept\_test(Deptno)

Alter Table emp\_test

Add constraint emp\_test\_ck check(sal >=3500)

Alter Table dept\_test

Add constraint dept\_test\_uk unique(deptname)

Insert Into emp\_test (Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)

Values(1000, 'Shubham', 'Trainee', Null, sysdate, 5000, Null, 10)

Insert Into dept\_test(Deptno, Deptname, Location)

Values(10, 'Training', 'Pune')

Insert Into dept\_test(Deptno, Deptname, Location)

Values(20, 'HR', 'Mumbai')

Insert Into dept\_test(Deptno, Deptname, Location)

Values(30, 'Development', 'Mumbai')

drop table dept\_test //---------->> once I drop the table it cant be get back

Create Table emp\_test

As //-------->>> created the same table having same data

Select \* from employees;

select count(\*) from emp\_test; //------->>> gives the records present in table

delete from emp\_test;

rollback; //---------->>> discarded the changes.

commit; //---------->>> saves Changes permanently

Insert Into Emp\_Test

Select \* from Employees

where deptid = 50;

Insert Into emp\_test(Employee\_ID, Firstname, Lastname, Email, Hiredate, Job\_ID)

Values(100, 'Shubham', 'Holey', 'shubham.holey', sysdate, 'Trainee')

/

Insert Into emp\_test(Employee\_ID, Firstname, Lastname, Address, City, Postal\_code, country, homephone)

Values(20, 'Sachin', 'Pimpale', 'Rahatni', 'Pune', 411017, 'INDIA', 8275191413)

update emp\_test

set EMPLOYEE\_ID =30

where Employee\_ID =10;

GRANT select on emp\_test to lab2ctrg13 //------>>> we should give the permission to another user for accessing the particular table.

(lab2ctrg13) //---------->>> it is the user Id of the particular user who wants to access our data table.

grant update on emp\_test to lab2ctrg13 //------->>> grant for updating the data

/

REVOKE select on emp\_test from lab2ctrg13 //-------->>> get back the updating permission from the user2.

grant select emp\_test to user1 with grant option; //----->>> if user1 wants to grant the permission on the same emp\_test table

to user3 he can't able to grant the permission like this,

he can grant the permission to user3 only when the

original user will grant the permission to user1 with command "......with grant option"

then only user1 can able to grant the access to user3.

Create table Trainees

As

select \* from lab2ctrg13.employees;

select count(\*) from trainees;

select department\_id, count(\*)

from trainees //---------->>> this command is used to count no. of employees by department wise and

group by department\_id; //------->>> this is used to make a group of that particular department and shows the total no. of employees by dept\_id wise.

select max(salary) from trainees //----->>> shows maximum salary

/

select max(salary), min(salary), avg(salary), sum(salary) from trainees

/

select count(\*), max(salary), min(salary), avg(salary), sum(salary) from trainees

group by department\_id

select department\_id, count(\*), max(salary), min(salary), avg(salary), sum(salary) from trainees

group by department\_id

select department\_id

from trainees

where department\_id is not null

group by department\_id

/

select department\_id

from trainees // shows the output in order manner

where department\_id is not null

group by department\_id

order by department\_id

select department\_id, count(\*), max(salary), min(salary), avg(salary), sum(salary)

from trainees

where department\_id is not null

group by department\_id

having count(department\_id) > 5

order by department\_id

update staff\_master

set staff\_sal = 20000

where Staff\_name = 'Mohan'

update staff\_master

set mgr\_code = null

where staff\_name ='Ram'

update book\_master

set book\_name = 'JDBC & MySQL'

where book\_name ='Let Us C'

Select Employee\_Id, First\_Name || ' ' || Last\_Name As Name

from Trainees;

Select Employee\_Id, First\_Name || ' ' || Last\_Name As Name, Salary, Department\_Name

from Trainees, Departments;

JOINING THE TWO TABLES:

Select Employee\_Id, First\_Name || ' ' || Last\_Name As Name, Salary, Department\_Name

from Trainees, Departments

where Trainees.department\_id = Departments.department\_id; //------>>> joining condition

Select Employee\_Id, First\_Name || ' ' || Last\_Name As Name, Salary, Department\_Name

from Trainees t, Departments d //----->>> t & d are variables

where t.department\_id = d.department\_id; //------>>> joining condition

SELF JOIN:- //-------->>> it is used to join the two rows from the same table. physically the table is one but virtually it acts as two table while joining

select worker.Employee\_id, worker.Last\_Name, worker.Manager\_id, manager.Last\_Name

from Trainees worker, Trainees manager

where worker.Manager\_id = manager.Employee\_id;

Create table books

( id Number(6) primary key,

title Varchar2(100),

author Varchar2(50),

price number (6, 2));

insert into books

values(101, 'Java Closures and Lambda', 'Robert Fischer', 460);

insert into books

values(102, 'Pro Java and Programming', 'Brett Spell', 600);

insert into books

values(103, 'Java for Android, second Edition', 'Budi Kurniawan', 680);

insert into books

values(105, 'Java EE 7: The Big Picture', 'Danny Coward', 340);

insert into books

values(104, 'Java: The Complete Reference, Ninth Edition', 'Herbert Schildt', 700);

insert into books

values(106, 'Beginning Java with WebSphere', 'Robert W. Janson', 550);

commit;

CREATE TABLE Greet

(messageId NUMBER(5) PRIMARY KEY,

message VARCHAR2(25));

Create Table merchant\_inventory

(

M\_Product\_Id Number(20) PRIMARY KEY,

M\_Id Number(20),

M\_Product\_Name Varchar2(20),

M\_Product\_Price Number(10),

M\_Product\_Quantity Number(20),

M\_Product\_Discount Number(20),

M\_Product\_Availability Number(20)

);

Create Table Products

(

Id Number(20) PRIMARY KEY,

Name Varchar2(20),

Rating Number(5),

Catagory Varchar2(20),

Discription Varchar2(20),

Quantity Number(20),

Price Number(10),

Brand Varchar2(20)

);

Create Table Customer

(

AccNo Number(6) PRIMARY KEY,

Pin Number(4),

Name Varchar2(15),

PhoneNo Number(10),

Address Varchar2(20)

);

Create Table Transaction

(

Transaction\_Id Number(10) PRIMARY KEY,

Type Varchar2(15),

AcNo Number(15),

Amount Number(15)

);

ALTER TABLE Customer

ADD Balance Number(15);

ALTER TABLE Customer

DROP COLUMN Balance;