Tcl--→ transaction control statements
Commit-→ make the changes permanent.
Rollback--→ undo the changes in the table
Savepoint-→ adding markers in between multiple dml statements.

To commit the changes either use commit, or if you execute any DDL statement, it will autocommit the changes

To set auto commit off
Set autocommit=0;

To set auto commit on
Set autocommit=1;

If you have a table mytab, it contains 10 rows.
Insert ---1
Insert----1
Insert ----1
---execute any DDL
Insert----2
Savepoint A
Update---2
Savepoint B
Delete ----10
Rollback to B

Course

cid	Cname	Duration	capacity	Rid	fid
1001	DAC	400	240	10	101
1002	DBDA	350	60	12	
1003	DTISS	300	60		102

Faculty

Facid	Fname	spskill
100	Raj	Java
101	Gayatri	C#
102	Rashmi	C++

room

Roomid	Rname	Rloc
10	Lotus	1st floor
11	Rose	1 st floor
12	Mogra	2nd floor

Select * from course c,faculty f Where c.fid=f.facid	Select * from course c inner join faculty f on c.fid=f.facid

Joins

When the o/p needs columns from more than one table then use joins

Types of joins

1. Cross join→If we combine 2 or more tables without any condition, then we call it as cross join,
   If we have 2 tables and in table 1 we have 10 rows, and in table 2 we have 4 rows, then in cross join we will get total 40 rows in the o/p(every row of table 1 will be combined with every row of table 2)

2. Inner join-→ cross join with join condition is called as inner join, it gives only matching rows in the output
   a. Equi join-→ if join condition uses = sign then it is called as equi join
   b. Non equi join---→ if join condition does not uses = sign then it is called as non equi join
   c. Self join--→ if the table joins itself then it is self join
3. Outer join-→ gives matching as well as non matching rows
   a. Right join-→if we want non matching rows from the table on right side,then use right join
   b. Left join-→ if we want non matching rows from the table on left side,then use left join

   c. Full outer join→ if we want non matching rows from both side tables, then use full outer join

--------to display all employees and their department names

select
empno,ename,e.deptno,d.deptno,dname
from emp e , dept d
where e.deptno=d.deptno;
	select
	empno,ename,e.deptno,d.deptno,dname
	from emp e inner join dept d
	on e.deptno=d.deptno;

--------to display all employees and their department names for all employees with sal> 2000

select
empno,ename,sal,e.deptno,d.deptno,dname
from emp e , dept d
where e.deptno=d.deptno and sal>2000;
	select
	empno,ename,e.deptno,d.deptno,dname
	from emp e inner join dept d
	on e.deptno=d.deptno
	where sal>2000;

--------display empno,name,sal, and grade of the employee

select empno,ename,sal,grade,losal,hisal
from emp e,salgrade s
where e.sal between s.losal and s.hisal;
	select empno,ename,sal,grade,losal,hisal
	from emp e inner join salgrade s
	on e.sal between s.losal and s.hisal;

------display all employeedetails with manager name

select e.empno,e.ename,e.mgr,m.empno mgrno,m.ename mgrname from emp e, emp m where e.mgr=m.empno;	select e.empno,e.ename,e.mgr,m.empno mgrno ,m.ename mgrname from emp e inner join emp m on e.mgr=m.empno;

-----to display pairs of employee names, who has same jobs,

Note: duplicate pairs should not be there)

Select e.empno,e.ename,e.job,m.empno,m.ename, m.job From emp e,emp m Where e.job=m.job and e.empno<m.empno	Select e.empno,e.ename,e.job,m.empno,m.ename, m.job From emp e inner join emp m on e.job=m.job and e.empno<m.empno

-----to display pairs of employee names, in such a manner that first employees sal < second employees sal

select e.empno,e.ename,e.sal,m.empno,m.ename, m.sal
from emp e,emp m
where e.sal<m.sal;
	select e.empno,e.ename,e.sal,m.empno,m.ename, m.sal
	from emp e inner join emp m
	on e.sal<m.sal;

,

---to find all employees name, department name, grade in the o/p

select e.empno,e.ename,e.sal,e.deptno,d.deptno,d.dname,s.grade,s.losal,s.hisal
from emp e,dept d,salgrade s
where e.deptno=d.deptno and e.sal between s.losal and s.hisal;
	select e.empno,e.ename,e.sal,e.deptno,d.deptno,d.dname,s.grade,s.losal,s.hisal
	from emp e inner join dept d on e.deptno=d.deptno inner join salgrade s on e.sal between s.losal and s.hisal;

select empno,ename,sal,e.deptno,d.deptno, dname,grade,losal,hisal from emp e,dept d,salgrade s where e.deptno=d.deptno and e.sal between s.losal and s.hisal;	select empno,ename,sal,grade,losal,hisal from emp e inner join dept d on e.deptno=d.deptno inner join salgrade s on e.sal between s.losal and s.hisal;

Course(cid,cname,duration,capacity,rid,fid)

Faculty(facid, fname,sp.skill)

Room(roomid,rname,rloc)

1. To display cid, cname,faculty name

Select cid,cname,f.facid,f.fname From course c,faculty f Where c.fid=f.facid	Select cid,cname,f.facid,f.fname From course c inner join faculty f on c.fid=f.facid

2. To display cid, cname,room name and roomlocation

Select cid,cname r.rname,r.rloc From course c, room r Where c.rid=r.roomid	Select cid,cname r.rname,r.rloc From course c inner join room r on c.rid=r.roomid

3. To display cid,cname,fname,room name,room loc

Select cid,cname,fname,rname,rloc from course c, faculty f,room r Where c.fid=f.facid and c.rid=r.roomid;	Select cid,cname,fname,rname,rloc from course c inner join faculty f on c.fid=f.facid inner join room r on c.rid=r.roomid;

-----to display empno,ename, deptno and dname for all employees, also display departments in which no employees are there

select empno,ename,e.deptno,d.deptno ,dname from emp e right join dept d on e.deptno=d.deptno;	select empno,ename,e.deptno,d.deptno ,dname from dept d left join emp e on e.deptno=d.deptno;

-----to display empno,ename, deptno and dname for all employees, also display employees for which no department is assigned.

select empno,ename,e.deptno,d.deptno ,dname from emp e left join dept d on e.deptno=d.deptno;	select empno,ename,e.deptno,d.deptno ,dname from dept d right join emp e on e.deptno=d.deptno;

-----to display empno,ename, deptno and dname for all employees, also display dname in which no employees are there, also display employees for which no department is assigned.

```
select empno,ename,e.deptno,d.deptno,dname
from emp e left join dept d
on e.deptno=d.deptno
union
select empno,ename,e.deptno,d.deptno,dname
from emp e right join dept d
on e.deptno=d.deptno
```

-----to display all departments in which no employees are there

Select * From dept d Where not exists (select * from emp e Where e.deptno=d.deptno	select empno,ename,e.deptno,d.deptno,dname from emp e right join dept d on e.deptno=d.deptno where ename is null;

-----to display all employees who are not assigned to any department

Select * From emp Where deptno is null	Select empno,ename,e.deptno,d.deptno,d.dname From emp e left join dept d on e.deptno=d.deptno Where dname is null;

----to display all employees who are not assigned to any department also display departments for which no employees are there

Select empno,ename,e.deptno,d.deptno,dname From emp e right join dept d On e.deptno=d.deptno where e.deptno is null union Select empno,ename,deptno, null,null from emp Where deptno is null	Select empno,ename,e.deptno,d.deptno,dname From emp e right join dept d On e.deptno=d.deptno where e.ename is null union Select empno,ename,e.deptno,d.deptno,dname From emp e left join dept d On e.deptno=d.deptno where d.deptno is null

-----to display all employees who have not assigned to any project

```
select empno,ename,e.deptno
from emp e
on e.ename is null;
```

window functions

row_number()	assign unique value to every row within window
rank()	assign number to distinct values, if the values same then same rank will be assigned to both rows. but when more than one row gets same rankl then it will skip numbers and then the next rank will be assigned
dense_rank()	assign number to distinct values, if the values same then same dense rank will be assigned to both rows. but when more than one row gets same dense rank, then it will not skip in between numbers and then the next rank will be assigned
lag(val, n)	it will find nth previous value
lead(val,n)	it will find nth next value
first_value(sal)	it will find first value of the given column within window

last_value(sal)	it will find last value of the given column within window

In the following example partition by will divide the data into window and order by will arrange data within window

```
select deptno,ename,sum(sal) over (partition by deptno)

from emp
```

```
select empno,ename,sal,deptno,row_number() over (),rank() over (order by sal desc)

from emp;
```

1. to find highly paid employee

   ```
   select *  from (

   select empno,ename,sal,deptno,row_number() over () rownum,rank() over (order by sal desc) rn,dense_rank() over (order by sal desc) drn from emp) e

   where e.drn=1;
   ```

2. to find highly paid employee in each department

   ```
   select *  from (
   select empno,ename,sal,deptno,row_number() over () rownum,rank() over (order by sal desc) rn,dense_rank() over (partition by deptno order by sal desc) drn from emp) e
   where e.drn=1;
   ```

3. to find first sal in each window

   ```
   select empno,ename,sal,deptno,first_value(sal) over (partition by deptno order by sal) fv
    from emp;
   ```

4. add next salary into current sal

   ```
   select empno,ename,sal,deptno,sal+lead(sal,1) over () lagval
      -> from emp;
   ```

   indexes

   2 types of indexes

   1. clustered index
      a. there will be only one clustered index
      b. it does not require extra space because it is stored along with data in the table
   2. non clustered index
      a. there can be many non clustered index
      b. these are stored outside table, and hence need extra space

indexes speed up the search action , but reduces the speed of DML statements, so donot create unnecessary indexes.

Types of indexes

1. primary key index-----indexes on primary key gets create automatically
2. unique index
   a. indexes on column with unique constraint gets created automatically
   b. the fields on which unique index is created, then duplicate values are not allowed in that column.

create unique index idx_passport

on emp(password desc)

3. regular index

create index idx_sal

on emp(sal desc)

4. full text indexes
    a. These are usually used on text type column,
    b. it stores phrases or words and their position

create full text index idx_profile

on emp(profile)

5. geospatial index – It is used when the field store geographical location.

to drop the index

drop index idx_passport on emp

show indexes from emp;

To find query uses which index

Explain select * from emp where sal>2000;

Views in mysql

There are 2 types views in database

1. view---→ simple views or complex views
2. materialized view---- when you are working on static data, then it is good to create materialized view.
   Once you fire the query 1 st time data will be retrieved, it will get save in RAM for the current session.
   It speeds up the job of retrieval of the data.

for every view, separate table will not get created, only base query gets stored for view.

if we fire select statement on view, then the base query will get executed

while creating view, if we use with check option, if view based on single table, and if view contains all not null columns, then one can perform DML operations on the view. and only valid data for which where condition is satisfied, can be added or removed or updated

To stop all DML operations on the view, use with read only option

why to use views

1. to give limited access to the table

2. hide complex queries under the view
3. we may hide table names, which increases security.

to create view

```
create <materialized> view   < name of the  view>

as

<base query>
```

to create a view for manager of dept 10 to give access to all the records of dept 10

```
create view mgr10

as

(select * from emp

where deptno=10

with check otion)

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

create view mgr10

as

(select * from emp

where deptno=10

with read only option)
```

```
create view testview1

   -> as

   -> select empno,ename,e.deptno edeptno,d.deptno ddeptno,d.dname

   -> from emp e, dept d

   -> where e.deptno=d.deptno;
```

```
create view testview

   -> as

   -> select deptno,job,sum(sal+ifnull(comm,0)) sumsal,count(*) cnt

   -> from emp
-> group by deptno;
```