

Joins in table

1. When you want to display values from more than one table then use joins
2. if we are joining n tables then minimum n-1 join conditions are needed

Types of joins

cross join	when every row in the table emp is joined with every row from other table, then it is called as crossjoin
inner join <ol style="list-style-type: none">1. equi join2. non equi join3. self join	<p>If we add join condition in the join query, then it is called as inner join</p> <ol style="list-style-type: none">1. if the condition is based on = sign then it is called as equi join2. if the condition is based on operator other than = then, it is called as non equijoin3. If in inner join we combine a table with itself, then it is called as self join
outer join <ol style="list-style-type: none">1. left outer join2. right outer join3. full outer join	<p>when you want to retrieve matching as well as non matching rows from multiple tables then use outer join</p> <ol style="list-style-type: none">1. if we want nonmatching rows from the table which is on the left side in from clause, then use left outer join2. if we want nonmatching rows from the table which is on the right side in from clause, then use right outer join3. if we want nonmatching rows from both side tables then use full outer join, to use full outer join we need to write union query in mysql.

1. to display empno, deptno,dname for all employees

<pre>select e.empno,e.deptno,d.dname from emp e, dept d where e.deptno=d.deptno;</pre>	<pre>select e.empno,e.deptno,d.dname from emp e inner join dept d on e.deptno=d.deptno;</pre>
--	---

2. display all employees and their manager names

<pre> select e.empno,e.ename,e.mgr,m.e mpno mgrno. m.ename mgrname from emp e, emp m where e.mgr=m.empno; </pre>	<pre> select e.empno,e.ename,e.mgr,m.empno,m.ename from emp e inner join emp m on e.mgr=m.empno; </pre>
--	---

3. to display empno, deptno,dname for all employees with sal>2000

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

4. to display empno, deptno,dname for all employees with deptno is either 10 or 20

<pre> select empno,e.deptno,dname from emp e,dept d where e.deptno=d.deptno and e.deptno in (10,20) </pre>	<pre> select empno,e.deptno,dname from emp e inner join dept d on e.deptno=d.deptno where e.deptno in (10,20) </pre>
--	--

5. to display empno,ename,sal, and grade
- ```

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

```

6. to display courses name along with room name

```
select cid,cname,c.rid,r.rid,rname
from course c, room r
where c.rid=r.rid;
```

7. to display courses name along with faculty name

```
select cid,cname,c.fid,fname
from course c,faculty f
where c.fid=f.fid;
```

8. list courses, with room name and faculty name

```
select cid,cname,fname,rname
from course c,room r,faculty f
where c.rid=r.rid and c.fid=f.fid
```

|                                                                                                                    |                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <pre>select cid,cname,c.rid,r.rid,rname  from course c, room r,faculty f  where c.rid=r.rid and c.fid=f.fid;</pre> | <pre>select cid,cname,c.rid,r.rid,rname,fname from course c inner join room r on c.rid=r.rid inner join faculty f on c.fid=f.fid</pre> |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|

9. to list all employee name and department name , along with departments which do not have any employee

|                                                                                                           |                                                                                                            |
|-----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| <pre>select empno,ename,e.deptno,d.dept no,dname  from dept d left join emp e on e.deptno=d.deptno;</pre> | <pre>select empno,ename,e.deptno,d.dept no,dname  from emp e right join dept d on e.deptno=d.deptno;</pre> |
|-----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|

10. find all the employees name and department name, also display employees for whom no dept is assigned, and also display departments in which no employee is there?

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

11. to display course name and faculty name, for all courses, and also display faculty names which are not assigned to any course

```
select cid,cname,fname
from course c right join faculty f on c.fid=f.fid;
```

12. to display course name and room name, for all courses, and also display room names which are not assigned to any course

```
select cid,cname,rname
from course c right join room r on c.rid=r.rid;
```

13. to display course name and faculty name, for all courses, and also display faculty names which are not assigned to any course, and also display courses for which no faculty is assigned.

```
select cid,cname,fname
from course c right join faculty f on c.fid=f.fid
union
select cid,cname,fname
from course c left join faculty f on c.fid=f.fid;
```

14. to display all employees, who are not assigned to any department, and all departments in which no employees are assigned

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

15. to display course name for which no faculty is assigned and display all faculty names which are not assigned to any course,

```
select cid,cname,fname
```

```
from course c left join faculty f on c.fid=f.fid
```

where f.fname is null

union

select cid,cname,fname

from course c right join faculty f on c.fid=f.fid

where c.cname is null;

16. find all employees for which no dept is assigned and all department for which no employee is assigned

select empno,ename,e.deptno,d.deptno,d.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,d.dname

from emp e left join dept d on e.deptno=d.deptno

where d.dname is null;

17. find all faculties which are not assigned to any course and find all rooms which are not assigned to any course

select c.cid,c.cname,f.fid,f.fname,null rid,null rname

from course c right join faculty f on c.fid=f.fid

where c.cname is null

union

select c.cid,c.cname,null,null,r.rid,r.rname

from course c right join room f on c.rid=r.rid

where c.cname is null

Table: Movies

| <b>Id</b> | <b>Title</b>   | <b>Director</b> | <b>Year</b> | <b>Length_minutes</b> | <b>Release date</b> |
|-----------|----------------|-----------------|-------------|-----------------------|---------------------|
| 1         | Toy Story      | John Lasseter   | 1995        | 81                    |                     |
| 2         | A Bug's Life   | John Lasseter   | 1998        | 95                    |                     |
| 3         | Toy Story 2    | John Lasseter   | 1999        | 93                    |                     |
| 4         | Monsters, Inc. | Pete Docter     | 2001        | 92                    |                     |
| 5         | Finding Nemo   | Andrew Stanton  | 2003        | 107                   |                     |

|   |                 |           |      |     |
|---|-----------------|-----------|------|-----|
| 6 | The Incredibles | Brad Bird | 2004 | 116 |
|---|-----------------|-----------|------|-----|

movieid is primary key

year >1970

length\_minutes>15 min and < 240 mins

Release\_date >1990-12-31 default '1991-01-01'

create table movie(movieid int primary key,

title varchar(20),

director varchar(20),

year int check(year>1990),

length\_min int check(length\_min between 15 and 240)

release\_date date check(release\_date>'1990-12-31') default '1991-01-01')

insert into movie values(1,'Toy story','John Lasseter',1995,81,'1995-03-02');

insert into movie values(2,'A Bugs Life','John Lasseter',1998,95,'1998-03-02');

Table: Boxoffice

| Movie_id | Rating | Domestic_sales | International_sales |
|----------|--------|----------------|---------------------|
| 5        | 8.2    | 380843261      | 555900000           |
| 14       | 7.4    | 268492764      | 475066843           |
| 8        | 8      | 206445654      | 417277164           |
| 12       | 6.4    | 191452396      | 368400000           |
| 3        | 7.9    | 245852179      | 239163000           |
| 6        | 8      | 261441092      | 370001000           |
| 9        | 8.5    | 223808164      | 297503696           |

movie(mid, title, director, year, length\_in\_min, releasedate)

boxoffice(movieid, rating, domestic\_sales, international\_sales)

1. display movieid, title, rating and domestic\_sales of all the movies.

select m.mid,m.title,b.rating,b.domestic\_sales

from movie m,boxoffice b

where m.mid=b.movieid

2. display movieid, title, rating and domestic\_sales of all the movies in which rating > 4 and domestic\_sale is > international\_sale.

```
select m.mid,m.title,b.rating, b.domestic_sales
```

```
from movie m,boxoffice b
```

```
where m.mid=b.movieid and b.rating>4 and b.domestic_sale> b.international_sale;
```

3. display all movies for which no rating is assigned

```
select * from movie m
```

```
where not exists (select * from boxoffice b where b.movieid=m.mid and rating is not null)
```

4. display all movie names with rating < average rating and director name starts with J

```
select * from movie
```

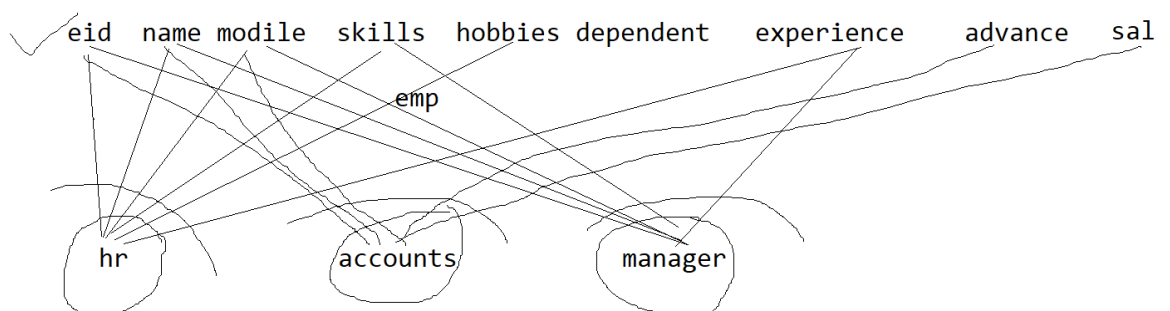
```
where director like 'J%' and mid in (select movieid
```

```
from boxoffice where rating<(select avg(rating) from boxoffice))
```

5. display all movies which are released in jan

```
select * from movies where month(releasedate)=1;
```

Views in myasql



```
create view mgr10
```

```
as
```

```
select * from emp where deptno=10;
```

to restrict the dml operation via view use with check option

```
create view mgr10
```

```
-> as
```

```
-> select * from emp
```

```
-> where deptno=10
```

```
-> with check option;
```

to stop DML operations on view, use with read only, but with read only works in oracle and not in mysql

```
create view mgr10
```

```
as
```

```
select * from emp
```

```
where deptno=10
```

```
with read only
```

When you want to provide limited access to the existing data, then we create views

views are of 2 types

1. views

- a. for view no separate memory is allocated for storing data, it only stores the base query.
- b. only base query will be stored, and any statement on the view will use base query to get the data, because of that we always get UpToDate data in view
- c. if view contains all not null columns of the single base table and if it is not read only view, then we can use all DML operation(insert, delete, update) on the view
- d. if the view is based on joins, aggregate functions, group by statement or union of multiple queries, then by default the views are readonly

```
select * from emp e
```

```
where sal<(select avg(sal) from emp m where m.mgr=e.mgr)
```

```
create a view fac_room
```

```
as
```

```
select cid,cname,fid,fname,null,null
```

```
from course c right join faculty f on f.fid=c.fid
```

```
where c.cname is null
```

```
union
```

```
select cid,cname,null,null,rid,rname
```

```
from course c right join room r on r.rid=c.rid
```

```
where c.cname is null
```

uses of views:

1. Hide complexity of the queries ( joins, aggregate functions nested queries)
2. To give restricted access to few columns or rows from tables
3. Hide table names, to increase the security of data.

2. Materialized view



- a. Views for which the first time the base query will get executed and then the output will be stored in a temporary table in RAM, within the session, the data will be retrieved from the RAM
- b. you may not get upto date data in materialized view.
- c. When your data is history data/ non changeable data, then use materialized view

create materialized view myview

as

select \* from emp;

to drop view

drop view myview

to get the 3<sup>rd</sup> highest salary

select \*

-> from emp s

-> where 2=(select count(\*)

-> from (select distinct sal from emp) e

-> where e.sal>s.sal)

-> order by sal;

to get 3 topmost salaried employee

select \*

-> from emp s

-> where 3>select count(\*)

-> from (select distinct sal from emp) e

-> where e.sal>s.sal)

-> order by sal;

to get 3 bottommost salaried employee

select \*

-> from emp s

-> where 3>select count(\*)

-> from (select distinct sal from emp) e

-> where e.sal<s.sal)

-> order by sal;

