Delete

it is used to remove one or many rows in your table based on certain conditions.

- delete all the records from the table.

  delete from {table_name}

- delete all the records from table where age is 22.

  delete from students
  where age = 22;

- delete all rows where email is null.

  delete from students
  where email is null;

JOINS

it is used when we have to combine two or more table on a related column between them

instructors (i)

inst-id	name	inst-age	dept-id
101	Aditya	28	204
102	divyanshy	38	203
103	Apoorva	29	202
104	narish	35	null
105	kynal	42	201

departments (d)

dept-id	name	dept-nop
201	computer	X
202	mechanical	Y
203	civil	Z
204	Physic	A
205	chemistry	B

* get all instructor name with their department name.

virtual table

g.inst-id	g. name	g.inst-age	g.dept-id	d.dept-id	d. name	d.dept-nod
101	Aditya	28	204	204	Physic	A
202	divyanshy	38	203	203	civil	Z
103	Apoorva	29	202	202	mechanical	Y
105	kunal	42	201	201	computer	r

Select   g. name , d. name
from   combine table
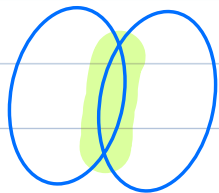where   condition g. dept_id = d. dept_id:

Select   g. name , d. name
from   Instructors (g) → nick name
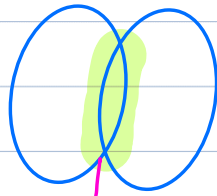Join   departments (d)
ON   g. dept_id = d. dept_id;

types of joins

s) inner join

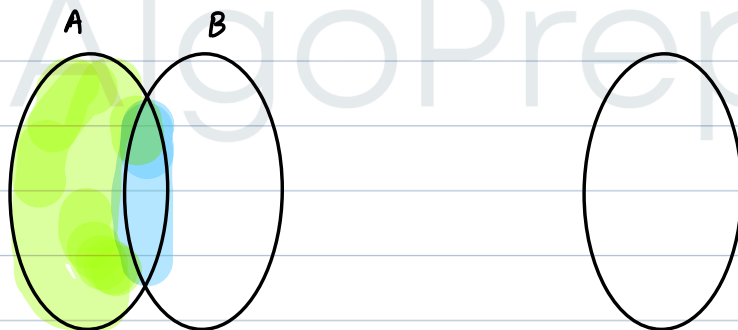- A inner join gives only those rows from both tables that have matching values in the specific columns.

matched records.

```
select    g.name  ,  d.name
from      instructors  (g)
inner     join    departments  (d)
ON        g.dept_id = d.dept_id;
```

Outer joins

if any row from A is not match with
any row from table B and vice verso.

:) **left outer join** / **left join**

all the rows from the left table and
the matched rows from the right table.

instructors (i)

inst-id	name	inst-age	dept-id
101	Aditya	28	204
202	divyanshy	38	203
103	Apoorva	29	202
104	narish	35	null
105	kynal	42	201

departments (d)

dept-id	name	dept-no
201	computer	X
202	mechanical	Y
203	civil	Z
204	physic	A
205	chemistry	B

| g.inst-id	g. name	g.inst-age	g.dept-id	d.dept-id	d. name	d.dept-no.f
101	Aditya	28	204	204	Physic	A
202	divyanshu	38	203	203	civil	Z
103	Apoorva	29	202	202	mechanical	Y
105	kynal	42	201	201	computer	X
104	harish	35	null	null	null	null

```
select    g.name  ,  d.name
from      instructors  (g)  >
  left   join    departments  d
  ON        g.dept_id = d.dept_id;
```

ii) right join

all the rows from right side and the matched rows from left side.
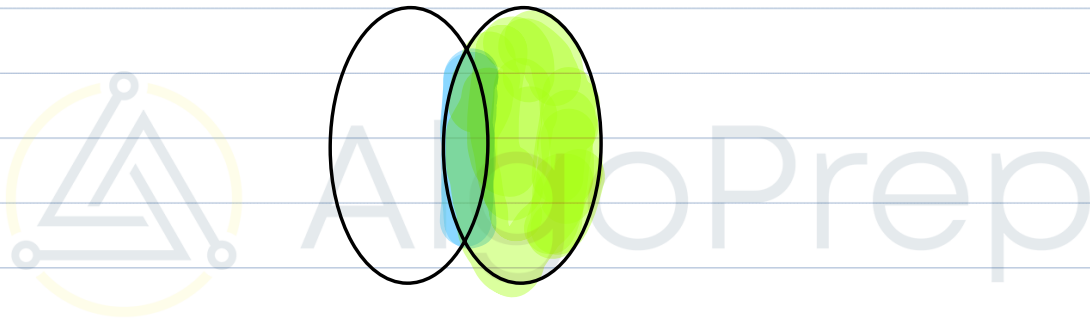
Instructors (i)

inst-id	name	inst-age	dept-id
101	Aditya	28	204
202	divyanshy	38	203
103	Apoorva	29	202
104	narish	35	null
105	kunal	42	201

departments (d)

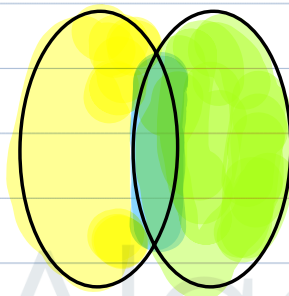| dept-id	name	dept-no9
201	computer	X
202	mechanical	Y
203	civil	Z
204	physic	A
205	chemistry	B

g.inst-id	g. name	g.inst-age	g.dept-id	d.dept-id	d. name	d.dept-nod
101	Aditya	28	204	204	Physic	A
202	divyanshy	38	203	203	civil	Z
103	Apoorva	29	202	202	mechanical	Y
105	kunal	42	201	201	computer	X
null	null	null	null	205	chemisty	B

select   g.name , d.name
from   instructors (g)
right   join   departments (d)
ON   g.dept-id = d.dept-id;

(iii) full join

all rows from both tables either
if is matched or not.

g.inst-id	g. name	g.inst-age	g.dept-id	d.dept-id	d. name	d.dept-hod
101	Aditya	28	204	204	Physic	A
102	divyanshu	38	203	203	civil	Z
103	Apoorva	29	202	202	mechanical	Y
105	kunal	42	201	201	computer	V
104	harish	35	null	null	null	null
null	null	null	null	203	chemistry	B

```sql
select    g.name , d.name
from      instructors  (g)  >
    full    join    departments  (d)
    ON        g.dept_id = d.dept_id;
```

```sql
select    g.name , d.name
from      instructors  (g)  >
    left    join    departments  (d)
    ON        g.dept_id = d.dept_id
    UNION
select    g.name , d.name
from      instructors  (g)  >
    right    join    departments  (d)
    ON        g.dept_id = d.dept_id;
```

break till 9:35

5) **Self join**

it is a regular join, but the table joined itself.

employee table

e_id	name	email	managerid
101	Aditya	ak@	104
102	divyanshu	xyz@	104
103	Apoorva	xcz1@	null
104	narish	xy21@	103
105	kunal	xyz@	103

get all emp_name with their manger name

select   e.name , m.name
from   employee e
join   employee m
On   e.manager_id = m.e_id;

using

it is the replacement of on keyword.

```
select    g.name  ,  d.name
from      instructors  (g)  >
   inner  join    departments  (d)
   using   dept_id
```

we can only use this keyword where
the column name should be same.

* ## Natural joins

A natural join is type of join operation
that automatically match column with the
same name in the both tables being joined.

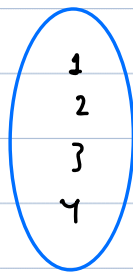```
select *
from   instuctors
natural join   departments;
```

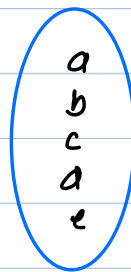* it gives the cartesian product of two tables
* all possible combinations of rows from both tables.

1
2
3
4

t1

a
b
c
d
e

t2

1a 1b 1c 1d 1e
2a 2b . --
3a 3b . _
4a 4b . - - .

n

m

n*m

select *
from instructor
cross join departments.

Q) get all the row of students with instructor.name

**student**

| s-id | s.name | c-id |
|------|--------|------|
| 101 | aditya | 201 |
| 102 | Rohit | 204 |
| 103 | Apoova | 201 |
| 104 | sohail | 202 |
| 105 | Rakesh | 204 |

**course**

| c-id | c-name | i-id |
|------|--------|------|
| 201 | english | 302 |
| 202 | chemistry | 301 |
| 203 | physics | 304 |
| 204 | Biology | 303 |

**instructor**

| i-id | i-name | email |
|------|--------|-------|
| 301 | X | abc 1@ |
| 302 | Y | abc2 @ |
| 303 | Z | abc 3@ |
| 304 | A | abc 4@ |

```sql
select   s.* ,  i. instructor_name
from       student  s
join    course  c
on      s. c_id =  c. c_id
join     instructor i
on      c. i_id =  i. i_id;
```

```sql
26    -- get all rows inst_id , inst _name , email , department name  and hod
27
28  • select i.instructor_id , i.instructor_name as inst_name , i.email ,
29      d.department_name , d.hod_name
30    from instructors i
31    join departments d
32    on i.department_id = d.department_id;
33
34
35  • select i.instructor_id , i.instructor_name as inst_name , i.email ,
36      d.department_name , d.hod_name
37    from instructors i
38    inner join departments d
39    on i.department_id = d.department_id;
40
41    -- get all the records from table instructor ,
42    --   if it is not matched any row in departments
43    --   then put null values
44
```

```sql
select i.instructor_id , i.instructor_name inst_name , i.email ,
  d.department_id , d.department_name , d.hod_name
from instructors i
left join departments d
on i.department_id = d.department_id;

-- get all the records from table department ,
--  if it is not matched any row in instructor
--  then put null values

select i.instructor_id , i.instructor_name inst_name , i.email ,
  d.department_id , d.department_name , d.hod_name
from instructors i
right join departments d
on i.department_id = d.department_id;

-- get all the records from both table whether i is matched or not
select i.instructor_id , i.instructor_name inst_name , i.email ,
  d.department_id , d.department_name , d.hod_name
from instructors i
left join departments d
on i.department_id = d.department_id
union
select i.instructor_id , i.instructor_name inst_name , i.email ,
  d.department_id , d.department_name , d.hod_name
from instructors i
right join departments d
on i.department_id = d.department_id;
-- using keyword--
select i.instructor_id , i.instructor_name as inst_name , i.email ,
  d.department_name , d.hod_name
from instructors i
inner join departments d
using (department_id);

select i.instructor_id , i.instructor_name as inst_name , i.email ,
  d.department_name , d.hod_name
from instructors i
cross join departments d;

select i.instructor_id , i.instructor_name as inst_name , i.email ,
  d.department_name , d.hod_name
from instructors i
natural join departments d;
```