

Q) What is join operation in DBMS.

In DBMS, Join used to combine data from 2 or more tables into a single table based on a related column between the tables. Join is denoted by  $\bowtie$

$\text{Join} = \text{Cartesian Product} + \text{Condition}$ .

$$A \bowtie B = \sum_{C} (A \times B)$$

$A \times B \rightarrow \text{Cartesian Prod.}$

Eg Emp

Emp ID	Emp Name	Add.
1	Alice	Delhi
2	Bob	Bihar
3	Candy	Bombay
4	Eric	Delhi

$\sum_C \rightarrow \text{Selector}$

$A, B \rightarrow \text{Two table with a related column.}$

Here Related Column =

Emp. Emp ID and Dept. Emp ID

Dept	D no	Name	Emp ID
	D1	HR	1
	D2	PR	2
	D3	IT	3
	D4	TE	4

Now ~~join~~ (Dept  $\times$  Emp)  $\rightarrow$  join.

~~and also =  $\sigma$  (Dept  $\times$  Emp)~~

Emp  $\times$  Dept ( $\text{Emp. Eno} = \text{Dept. Eno}$ )

Eno	EName	CAdd	Dno	DName	Eno
1	Alice	Delhi	D1	HR	1
2	Bob	Bk	D2	PR	2
3	Candy	Bombay	D3	IT	3
4	Ene	Delhi	D3	IT	4

$\leftarrow \text{Eno} = \text{Dept. Eno} \approx 1$

= 008

= 006

(A)

So join should have some common attribute

in both table, here we have Eno

The domain of common attribute must be same.

Here Emp Table and Dept table are

joined based on Eno and Dept. Eno.

There are some queries

Eg find the department name of Alice

$\rightarrow$  Emp table does not have got Alice info

$\rightarrow$  But doesn't have Department Name in Emp table

→ Department name is present in Dept table. (2)

→ We have common Eno in Emp and Dept

So to find Dname of Alie → Join Emp and Dept

(Emp X Dept)

(Dno).

(Ename = Alie)

[Emp. Eno = Dept. Eno]

Refer (A) of page 2

ENO	Name	DNO	DNO	DNAME	ENO
1	Alie	Delhi	D.	HR	1

HR

So Dept Name of Alie = HR

2) Find the name of all employees working in IT

By observing Dept → Eno 3, Eno 4.

" Emp → 3 is Cindy & 4 is Eric

Ans (Cindy and Eric)

Here also, the result is not available in a single table. So need to join tables

(S)   
 = find All employee of IT department.

(A)

(Ename)

(Emp X Dept)

(Dname=IT)

(Emp.Eno = Dept.Dno)

(emp.Ename = emp.Ename)

Ref: (A) of page

Gno	Ename	EDo	Dno	Dname	Eno
3	Cindy	Bombay	D3	IT	3
4	Eric	Delhi	D3	IT	4

Ename
Cindy
Eric

⇒ Am = Cindy & Eric

(5)

③ Find e-name of Employee who work in HR

ED

(Emp XI Dept)

(Ename)

(Dname = HR)

(Emp. Eno = Dept. Dno)

④ Find name of Employee who work in HR dept and belongs to Delhi

ED

(Emp XI Dept)

(Ename)

(Dname = HR)

AND

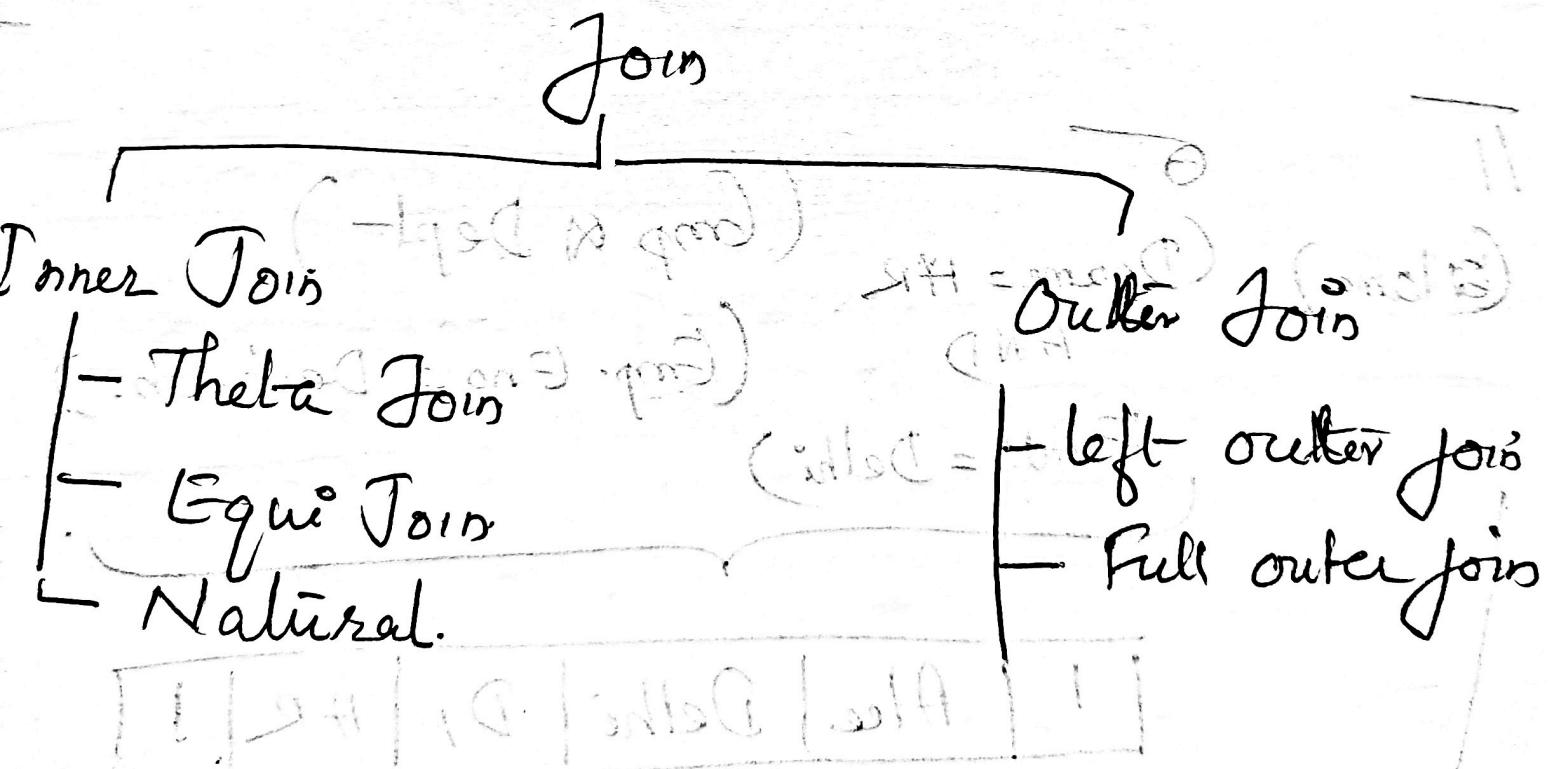
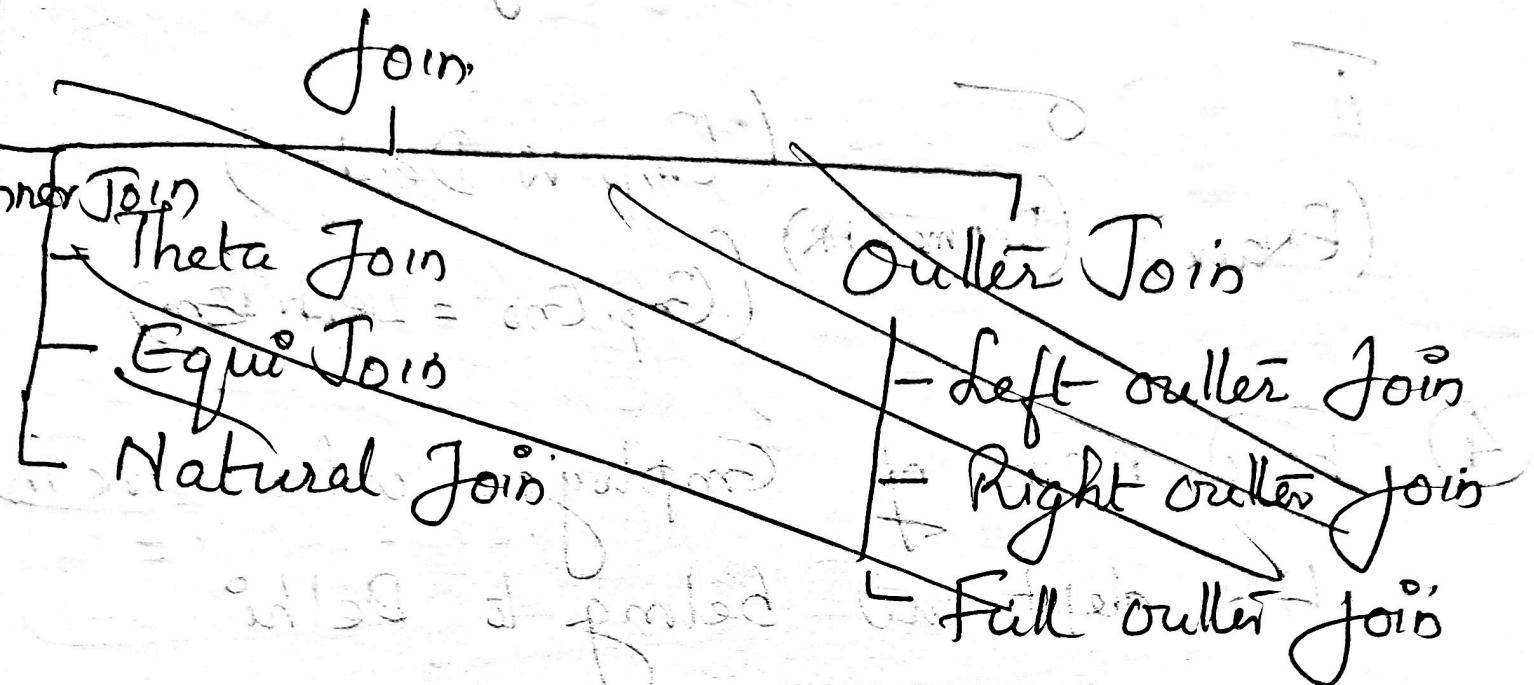
(Emp. Eno = Dept. Dno)

(Eadd = Delhi)

1	Alee	Delhi	D.	HR	1
---	------	-------	----	----	---

Alee.

## Types of Join



QUESTION

## 1) Theta Join

(7)

→ General case of join,  $(A \bowtie B)$

→  $\theta =$  Some condition, otherwise called Predicate.

= [ $<$ ,  $>$ ,  $\geq$ ,  $\leq$ ,  $=$ ,  $\neq$ ]

→ Schema of Result = Schema of  $(A \bowtie B)$

So Theta join can have any condition

Def -  $\theta$  join is a type of join operation

between two tables that combines rows

from both tables based on a condition

that is expressed using Theta  $\theta$  symbol

It is more flexible as  $\theta$  join allows  
more options to joining condition such as,

[ $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $=$ ,  $\neq$ ], compared

to other join

Ques 6 & 7. What is theta join?

Equi Join - Join operation between two tables that combines rows from both tables based on if the values corresponding values of common attribute match to be equal.

- 1) There must be a common attribute in both tables
  - 2) The common attribute must be derived from same domain
  - 3) If the attribute name is same in both tables only one attribute is required in the result table.
- Ex:- need to represent both attributes in result table.

Eg Student  
Table.

S-ID	S-Name	Std	Class	Subject
1	Alee.	11	11	Maths
2	Bob	12	11	Physics

Common Attributes - Student.Std and Subject.Class.

S-ID	Name	std	klass	Subject
1	Alex.	11	11	Maths
1	Alex	11	11	Phy়া
2	Bob	12	12	English
2	Bob	12	12	Chemistry

## Notes

(Student  $\bowtie$  Subject)

(std. S.W) = Subject. class

OR

(Student  $\bowtie$  Subject)

(S.W) = class

## Natural join

If common attribute got

- 1) Same name in both table
- 2) Derived from same domain

→ It's implicit join

→ No need to specify condition

↳  $(A \bowtie B)$  is sufficient.

→ Scheme of result  $\neq$  Scheme of  $A \times B$

↳ Need not specify common attribute

↳ two lines in result table. Only one common attribute is required.

So No of col in result = No of col in  $A \times B - 1$

$\Rightarrow$  Natural join = Cross product + Select (conds)  
+ Project All except  
one common  
attribute

$\Rightarrow$  Notation  $(A * B)$

~~can have more than one common attribute~~

Cmp.

Eid	Ename	Address
1	Alee	Delhi
2	Bob	Bkr
3	Sam	Chennai
4	Eric	Coches.

Dept. and Inclusion

Dno	DName	Eid
D1	HR	1
D2	PR	2
D3	IT	4

(Emp  $\times$  Dept)

(Eid, Ename, Add)  $\times$  (Eid, Dno)

Dno, Dname)

N-Join = ~~Outer Join~~ (Emp  $\star$  Dept)

Eid	Ename	Address	Dno	Dname
1	Alee	Delhi	D1	HR.
2	Bob	Bkr	D2	PR
4	Eric	Coches.	D3	IT

$\Rightarrow$

$\rightarrow$  Note - Eid is the common attribute,

$\rightarrow$  Derived from same domain

$\rightarrow$  Does not repeat in (Emp  $\star$  Dept)

- Note Join condition is not specified  
in notation of Natural join

- Schema result  $\neq$  Schema of (Emp  $\times$  Dept)

Equivalent =

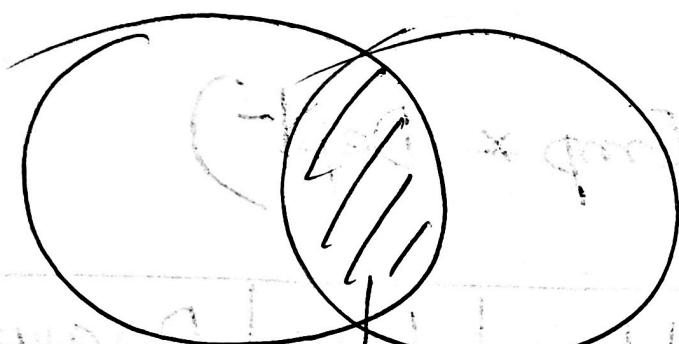
~~Join~~

## Natural Join

Def  $\Rightarrow$  Equijoin, list all attributes

$\Rightarrow$   $g_1 \quad g_2$   
 $t_1 \quad t_2$

$g_3 \quad g_4$   
list  
values



~~Inner Join~~

Natural Join

$g_1$

$g_2$

$t_1$

$g_3$

$g_4$

$t_2$

$g_5$

$g_6$

$t_3$

$g_7$

$g_8$

$t_4$

$g_9$

$g_{10}$

$t_5$

• Natural join removes all Q3.

means must match

(step 2) in conjunction

(obligatory to indicate both)

• Example:  $t_1 = t_2$  and  $t_3 = t_4$

## Outer Join

1) Left outer join.

2) Right outer join

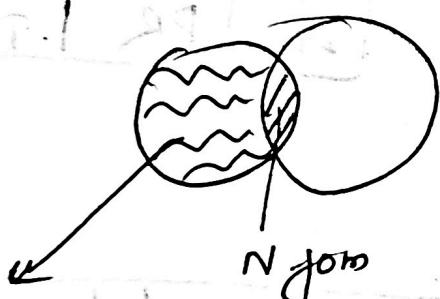
3) Full outer join

1) Left outer



Left -  
Right side.

Natural Join + which is left, table data



Extra in  
left-Table

Group E

Eid	Ename	Dept
E1	Alex	D1
E2	Bob	D2
E3	Sam	D1
E4	Eric	-

Dept.

Dept	Loc	Loc
D1	IT	Delhi
D2	HR	Chennai
D3	Fin	Bangalore

Eid Ename Dept Dname Loc

N join

E1	Alex	D1	IT	Delhi
E2	Bob	D2	HR	Chennai
E3	Sam	D1	IT	Delhi
E4	Eric	-	-	-

Extra of  
from left

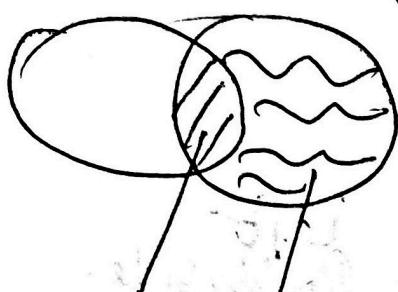
## Right outer join

N-Join + Something left in Left side Table

Right

Left side Table

Not in Left and  
present in Right



N-Join

Extra in  
Right side.

ENO	Gname	DNO
E1	Alee	D1
E2	Bob	D2
E3	Sam	D3
E4	Enc.	-

Dno	Dna	Loc
D1	IT	Def
D2	HR	Chean
D3	FIN	Bkf
D4	PR	Tvm

Emp. (Dept.)

ENO	Gname	DNO	Loc Dna
E1	Alee	D1	IT Def
E2	Bob	D2	HR Chean
E3	Sam	D3	FIN Bkf
-	-	D4	PR Tvm

Not in Left

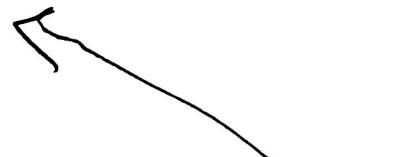
and present in Right.

Full Outer Join

Left outer join  $\cup$  Right outer join  
(union)

$$(\text{Emp} \setminus\!\!\!\setminus \text{Dept}) \cup (\text{Emp} \times \text{Dept}) \\ \Rightarrow (\text{Emp} \setminus\!\!\!\setminus \text{Dept})$$

EID	ENAME	DID	DNAME	LOC
E1	Alex	D1	IT	Delhi
E2	Bob	D2	HR	Chennai
E3	Sam	D1	IT	Delhi
E4	Mike	-	-	-
-	-	D4	PR	TVM



This is Not  $\rightarrow$  this is not cartesian product.

Cno	Ename	Dept	Dname	Loc
E1	Alex	D1	IT	Delhi
E2	Bob	D2	HR	Chennai
E3	Sam	D1	IT	Delhi
E4	Mike	-	-	-

UNION

EID	ENAME	DEPT	DNAME	LOC
E1	Alex	D1	IT	Delhi
E2	Bob	D2	HR	Chennai
E3	Sam	D3	PR	Bang
-	-	-	PR	TVM