Natural-Join Operation

□ Notation: $r \bowtie s$

Let r and s be relations on schemas R and S respectively.
Then, $r \bowtie s$ is a relation on schema $R \cup S$ obtained as follows:

- □ Consider each pair of tuples t_r from r and t_S from s.
- □ If t_r and t_S have the same value on each of the attributes in $R \cap S$, add a tuple t to the result, where
  - □ t has the same value as t_r on r
  - □ t has the same value as t_S on s

□ Example:

$R = (A, B, C, D)$
$S = (E, B, D)$

- □ Result schema = (A, B, C, D, E)
- □ $r \bowtie s$ is defined as:

$$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E}\ (\sigma_{r.B\ =\ s.B\ \wedge\ r.D\ =\ s.D}\ (r \times s))$$

Natural Join Operation – Example

□ **Relations r, s:**

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

OUTER JOIN:

The outer join operation is an extension of the join operation to deal with missing information.

There are three forms of outer join

- left outer join
- right outer join
- full outer join

employee:

Empname	Street	City
Coyote	Toon	Hollywood
Rabbit	Tunnel	carrot
Smith	Revolver	Death valley
William	Seavlew	Seattle

Ft_works:

Empname	Branch name	Salary
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
William	Redmond	1500

Employee ⋈ ft_works

Empname	Street	City	Branch name	Salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	carrot	Mesa	1300
William	Seaview	Seattle	Redmond	1500

Left outer join:

It takes all tuples in the left relation that did not match with any tuple in the right relation, pads the tuples with null values for all other attributes. The right relation and adds them to the result of the natural join. In tuple (smith, Revolcer, Death valley, null, null) is such a tuple. All information from the left relation is present in the result of the left outer join.

Empname	Street	City	Branch name	Salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	carrot	Mesa	1300
William	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death valley	Null	null

Result of Employee ⟕ ft_works

Right outer join:

It is symmetric with the left outer join. It pads tuples from the right relation that did not match any from the left relation with nulls and adds them to the result of the natural join. tuple(Gates,null,null,Redmond,5300) is such a tuple. Thus, all information from the right relation is present in the result of the right outer join.

Empname	Street	City	Branch name	Salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	carrot	Mesa	1300
William	Seaview	Seattle	Redmond	1500
gates	Null	null	Redmond	5300

Full outer join:

It does both of those operations, padding tuples from the left relation that did not match any from the right relation, as well as tuples from the right relation that did not match any from the left relation, and adding them to the result of the join. Figure 3.35 shows the result of a full outer join.

Since outer join operations may generate results containing null values, we need to specify how the different relation-algebra operations deal with null values. It is interesting to note that the outer join operations can be expressed by the basic relational algebra operations. For instance the left outer join operation

Employee ⟗ **ft_works**

Empname	Street	City	Branch name	Salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	carrot	Mesa	1300
William	Seaview	Seattle	Redmond	1500
gates	Null	null	Redmond	5300

Natural Join

~~Common~~ attribute. [illegible] 4

$4 \times 3 = 12$

$m \times n$ no. of rows.

Emp. m no. of rows

Emp.No	Ename	Address
1	Ram	Delhi
2	Varun	chd.
3	Ravi	chd
4	Amit	Delhi

Dept. n no. of rows 3

Dept No.	Name	E no
D1	HR	1
D2	IT	2
D3	Elect	4

Join → Cross product + Condition means

E.No	E name	Dept No	Eno	
1	RAM	D1	1	✓
1	Ram	D2	2	✗
1	Ram	D3	4	✗
2	Varun	D1	1	✗
2	Varun	D2	2	✓
2	Varun	D3	4	
-	-	.		

Emp | Deptt

⇓

Emp No	Ename	Deptt No	Eno
1	Ram	D1	1
2	Varun	D2	2
4	Amit	D3	4

Ex: Find the Emp Names who is working in a Deptt.

Q1 Select Emp.name from Employee, Deptt where

Emp.Empno = Deptt.Emp.No

↳ we are keeping equal for Natural Join.

Actually we are writing Select Ename from Emp Natural Join Deptt.

Self Join: find student id who is enrolled in at least two courses

at least = 2

may be more than 2 i.e 3, 4, ... ∞

Sid | Cid

Sid & Cid is a composite Primary Key

S-id	C-id	Since
S1	C1	2016
S2	C2	2017
S1	C2	2017

Study T1 ← Intermediate table

S1	C1	2016
S2	C2	2017
S1	C2	2017

T2

Select T1.Sid from Study as T1, Study as T2

where T1.Sid = T2.Sid and T1.Cid <> T2.Cid

Out - S1

join = Cross product + some condition.

Study table will be multiplied by Study table

T1		T2		
~~S1~~	~~C1~~	~~S1~~	~~C1~~	
~~S1~~	~~C1~~	~~S2~~	~~C2~~	
S1	C1	S1	C2	✓
~~S2~~	~~C2~~	~~S1~~	~~C1~~	
~~S2~~	~~C2~~	~~S2~~	~~C2~~	
~~S2~~	~~C2~~	~~S1~~	~~C2~~	
S1	C2	S1	C1	✓
~~S1~~	~~C2~~	~~S2~~	~~C2~~	
~~S1~~	~~C2~~	~~S1~~	~~C2~~	

Equi Join =

Join = Crossproduct + Condition

Find the Emp name who worked in a department having location same as their address?

Emp.

Eno	E-name	Address
1	Ram	Delhi
2	Varun	chd
3	Ravi	chd
4	Amrit	Delhi

Dept.

Dep.no	location	E-no
D1	Delhi	1
D2	Pune	2
D3	Patna	4

Emp	Dept.
1 RAM Delhi	D1 Delhi 1 ✓
~~1 RAM Delhi~~	~~D2 Pune 2~~
~~1 RAM Delhi~~	~~D3 Patna 4~~
~~2 Varun chd~~	~~D1 Delhi 1~~
~~2 Varun chd~~	~~D2 Pune 2~~
~~2 Varun chd~~	~~D3 Patna 4~~
~~3 Ravi chd~~	~~D1 Delhi 1~~
3 Ravi chd	D2 Pune 2
~~3 Ravi chd~~	~~D3 Patna 4~~
~~4 Amrit Delhi~~	~~D1 Delhi 1~~
~~4 Amrit Delhi~~	~~D2 Pune 2~~
~~4 Amrit Delhi~~	~~D3 Patna 4~~

Select E_name from Employee, Dept. where
Emp. Eno = Dept. Eno and
Emp. Address = Dept. Location

f and f = f
7

output = RAM

Left outer Join: It gives the matching rows and ~~it gives~~ the rows which are in left table but not in right table.

Emp-No	E-name	Dept. no
E1	Varun	D1
E2	Amrit	D2
E3	Ravi	D1
E4	Nitin	—

Deptno	D-name	Loc.
D1	IT	Delhi
D2	HR	Hyd.
D3	Finance	Pune.